Inaugural-Dissertation

zur Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der
Ruprecht-Karls-Universität
Heidelberg

vorgelegt von

Dipl.-Math. Lilian Kramer
aus Saarbrücken

Tag der mündlichen Prüfung: 6. Dezember 2013

Modeling Price Formation in a Multi-Commodity Market - A Graph-Theoretical Decomposition Approach to Complexity Reduction

Advisors:
Prof. Dr. Dr. h.c. mult. Willi Jäger
Prof. Dr. Sebastian Sager

Abstract

This thesis presents an optimization model to simulate the global price formation of multiple commodities over multiple time periods. The model considers the connection of commodities through their production processes. The supply side maximizes its total profit taking account of the price-demand relationships of all products. The variables of this model are production quantities, transport quantities, storage quantities, and commodity prices. We apply the model to a part of the petrochemical market.
A large multi-commodity model requires many parameters. Moreover, the interpretation of the simulation results can become difficult. Therefore, this thesis focuses on the model and complexity reduction with respect to optimization models.
We propose a graph-theoretical approach to reveal the structure of large block-separable problems and to compare different decompositions into subproblems. The connections between primal and dual variables of a constrained optimization problem are represented on a hypergraph, which can be analyzed and beneficially partitioned using appropriate graph-theoretical methods. We show how different partitions of the hypergraph constitute different decompositions of the optimization problem. Furthermore, we address the approximation of subproblems.
The decomposition approach is adapted to the commodity market model. We formulate the subproblems for chosen sets of products and processes and present an algorithm for the automated identification of model components that are suited for an aggregation. The aggregation of components of the market model in terms of approximating subproblems is discussed from different points of view. Furthermore, we conduct sensitivity analyses within the overall problem and within subproblems. The numerical results of the application to a petrochemical market model reveal different possibilities of model reduction.

Zusammenfassung

Diese Arbeit stellt ein Optimierungsmodell zur Simulation der globalen Preisbildung mehrerer rohstoffähnlicher Waren über mehrere Zeitperioden vor. In diesem Modell wird der Zusammenhang der Waren aufgrund ihrer Herstellungsprozesse miteinbezogen. Die Angebotsseite maximiert ihren Gesamtgewinn unter Berücksichtigung der Preis-Nachfrage-Zusammenhänge aller Waren. Die Variablen dieses Modells sind Produktionsmengen, Transportmengen, Lagermengen und Warenpreise. Das Modell wird auf einen Teil des Petrochemiemarktes angewandt.
Für ein großes Modell mit mehreren Waren sind viele Parameter erforderlich. Außerdem kann die Interpretation der Simulationsergebnisse kompliziert werden. Daher liegt der Schwerpunkt dieser Arbeit auf der Modell- und Komplexitätsreduktion in Bezug auf Optimierungsmodelle.
Ein graphentheoretischer Ansatz wird vorgeschlagen, um die Struktur eines großen blockseparablen Problems offenzulegen und um unterschiedliche Dekompositionen in Teilprobleme zu vergleichen. Die Verbindungen zwischen primalen und dualen Variablen eines restringierten Optimierungsproblems werden durch einen Hypergraphen dargestellt, der mit geeigneten graphentheoretischen Methoden analysiert und vorteilhaft zerlegt werden kann. Es wird gezeigt, inwiefern unterschiedliche Zerlegungen des Hypergraphen unterschiedliche Dekompositionen des Optimierungsproblems darstellen. Zudem wird die Approximation von Teilproblemen diskutiert.
Der Dekompositionsansatz wird auf das Marktmodell für rohstoffähnliche Waren angewandt. Für ausgewählte Produkt- und Prozessmengen werden Teilprobleme formuliert. Des Weiteren wird ein Algorithmus zur automatischen Identifikation von Modellkomponenten, die sich zur Aggregation eignen, vorgeschlagen. Die Aggregation von Marktmodellkomponenten im Sinne der Approximation von Teilproblemen wird unter verschiedenen Gesichtspunkten erörtert. Außerdem werden Sensitivitätsanalysen im Gesamtproblem und in Teilproblemen durchgeführt. Aus den numerischen Ergebnissen der Anwendung auf ein Petrochemiemarktmodell lassen sich verschiedene Möglichkeiten der Modellreduzierung ableiten.

Danksagung

An dieser Stelle möchte ich die Gelegenheit nutzen, denjenigen zu danken, die auf ganz unterschiedliche Weise zum Gelingen dieser Arbeit beigetragen haben.

Mein besonderer Dank gilt Prof. Dr. Dr. h.c. mult. Willi Jäger für die Möglichkeit, diese Dissertation unter seiner Betreuung in der Arbeitsgruppe Applied Analysis der Fakultät für Mathematik und Informatik der Ruprecht-Karls-Universität Heidelberg anzufertigen, und mich somit auch im Anschluss an meine Diplomarbeit wissenschaftlich mit der Modellierung von Rohstoffmärkten zu befassen. Sein Vertrauen, seine Erfahrung und seine Zuversicht haben mich insbesondere in schwierigen Phasen motiviert und seine Denkanstöße und Ratschläge waren mir immer hilfreich.

Ebenso möchte ich mich besonders bei Prof. Dr. Sebastian Sager von der Otto-von-Guericke-Universität Magdeburg für die Betreuung dieser Dissertation und die Vermittlung seines Wissens und seiner Erfahrung im Bereich der Optimierung bedanken. Seine Unterstützung und Motivation haben zum Gelingen dieser Arbeit beigetragen. Seine Anregungen bei Projekt- und Mentorengruppentreffen während der gemeinsamen Zeit an der Universität Heidelberg haben mir geholfen, die Forschungsrichtung meiner Doktorarbeit festzulegen.

Dem Interdisziplinären Zentrum für wissenschaftliches Rechnen (IWR) und seiner Graduiertenschule HGS MathComp danke ich für die finanzielle Unterstützung und die Gelegenheit, unter besonders guten Rahmenbedingungen forschen zu dürfen.

Bei Sabrina Kellner und Holger Diedam möchte ich mich herzlich für die gute Zusammenarbeit bei der Entwicklung des Rohstoffmarktmodells in der Anfangsphase unserer Doktorarbeiten bedanken. Der stetige Austausch mit Sabrina und die gegenseitige Motivation haben die Freude an der Arbeit bis heute aufrechterhalten. Holgers Hilfsbereitschaft hat mir das Einarbeiten in C++ und AMPL sehr erleichtert.

Bei Dr. Simon Jäger möchte ich mich besonders für den Austausch von Erfahrungen im Bereich der Modellierung von Rohstoffmärkten und seine Anregungen bei der Modellwahl bedanken.

Sabrina Kellner, Christian Kramer und Michael Jung danke ich speziell für das Korrekturlesen unterschiedlicher Teile dieser Arbeit und die wertvollen Anmerkungen und Ratschläge.

Ina Scheid und Gabriela Schocke danke ich herzlich für die Hilfe bei organisatorischen Angelegenheiten, und dafür, bei Bedarf immer eine Möglichkeit für ein Treffen mit Herrn Jäger gefunden zu haben.

Der Arbeitsgruppe Applied Analysis möchte ich für die tolle Gruppenatmosphäre, die gemeinsamen Grillabende, Gruppenfrühstücke und Feiern danken. Sabrina Kellner, Dr. Stephan Ludwig, Cristian Croitoru und Dr. Anamaria Bodea gilt mein Dank besonders für den Erfahrungsaustausch bei der Modellierung von Markt- und Preisdynamiken und viele schöne und lustige Diskussionen. Außerdem möchte ich Dr. Franziska Matthäus und ihrer Forschungsgruppe Complex Biological Systems für die Hilfsbereitschaft und Initiierung verschiedenster Gruppenaktivitäten danken.

Den Arbeitsgruppen Simulation and Optimization und Mathematical Algorithmic Optimization danke ich außer für den fachlichen Austausch und die Hilfsbereitschaft speziell für die Einladungen zu den beiden Gruppenausflügen und Grillabenden, die viel Spaß gemacht haben.

Ein spezielles Dankeschön an Sabrina, Anne, Raphael, Tobi, Agnes, Stephan, Martin und Georg für die abwechslungsreichen, unterhaltsamen und anregenden Gespräche in den gemeinsamen Mittagspausen, die einen angenehmen Ausgleich zur Arbeit dargestellt haben, und vor allem in besonders arbeitsintensiven Zeiten zu meinem Wohlbefinden beigetragen haben.

Ganz besonders möchte ich meinen Eltern, Christine und Lukas, meinen Großeltern, Ischu und Erwin, sowie meinen beiden Brüdern, Christian und Lorenz, dafür danken, dass sie immer für mich da sind und ich mich uneingeschränkt auf sie verlassen kann. Während meines Studiums und meiner Promotion haben sie mich stets motiviert und in jeglicher Hinsicht unterstützt.

Meiner Familie und meinen Freunden, an dieser Stelle ganz speziell den nicht (mehr) Heidelbergern, danke ich besonders herzlich für die vielen schönen gemeinsamen Erlebnisse, Diskussionen, Ausflüge und Reisen, durch die ich in meiner Freizeit wunderbar entspannen und Energie sammeln konnte.

Contents

1 Introduction

Motivation

A commodity is, by its original definition, a physical good of uniform quality. This means, commodities of the same type are readily interchangeable. The availability and the prices of commodities have a major impact on the global economic development. Standard examples of commodities include agricultural products, energy, and metals. A generalized definition of commodities also comprises services. In this sense, for instance, airline tickets for a certain flight route at a certain date can be regarded as a commodity. This work focuses on the price formation of commodities in the original sense. In general, commodity prices are determined by supply and demand, i.e., they are equilibrium prices. The supply is defined by production and inventory. The demand is usually price-dependent and influenced by economical factors. In contrast to financial products, commodities are physically traded. Therefore, transport costs have an impact on global commodity trade. Moreover, holding a commodity causes storage costs. Considering historical data, prices of commodities show a common development, which indicates that they do not form independently.
To promote scientific computing in new application areas, the Interdisciplinary Center for Scientific Computing (IWR) supports several Pioneering Projects. One of the research topics concerns the modeling and simulation of price dynamics in commodity markets. In this context, this thesis contributes to a partial project, which aims at developing a multi-commodity market model that is able to explain and simulate price formation. We focus on markets in which products are connected by operational processing functions. The regionally distinct prices are subject to the global interplay between producers and consumers. The resulting model is supposed to provide a rational approach to operational planning and risk management by facilitating the analysis and prognosis of prices, various scenario simulations, sensitivity analysis, and, thus, the identification of the essential market and price drivers.
There are several fields of research dealing with commodity markets. Our approach to market and price modeling differs in several major aspects from the established scientific approaches. Usually, stochastic models are applied for risk management in commodity markets and the pricing of commodity derivatives [Gem05, Pil07, BK04]. These models were originally designed for stock prices and interest rates and are currently extended for modeling the development of commodity prices. They include few assets and do not take into account the system aspect of the price formation. This means, these models disregard the technologies that connect different commodities in a deterministic way. In contrast, we aim to explicitly model the interplay between supply and demand. The research

areas production optimization [Hac08, Ras11], supply chain management [SK07, VW03], and chemical engineering [KC08, AQE09, Hüb07] resemble our modeling approach, since they use optimization models for process planning. Consequently, the modeling of production technologies is substantial. Typically, the variables of the respective models are production and transport quantities, while product prices enter as parameters. Only a few publications deal with the simultaneous modeling and optimization of pricing and production or assortment planning [Cha05, LB08, Kan08, KGvB$^+$09, TBS12]. Nevertheless, supply optimization approaches including pricing are of great importance for many industries.

One difficulty of a comprehensive market model is the need for a large amount of data and information. Since detailed data is often hardly available, data procurement can become expensive and time-consuming. In order to solve this problem, we will proceed as follows. We start with an extensive model and try to reduce it to its major components. This means, our model basically includes all potentially necessary information in terms of parameters. Analyzing this comprehensive model, we aim to differentiate between influential parameters and parameters that have less effect. In order to do this, appropriate methods for model reduction and complexity reduction are required.

Since we propose to model a multi-commodity market by a constrained nonlinear optimization problem that shows a certain network structure, there are two major mathematical fields of research connected to our work. These are graph theory and optimization theory. We shortly discuss some of their research directions with regard to our main objective of reducing the complexity of the market model. The scientific field of network optimization [BMMN95, EM92] seems close to our work. However, there is no formulation of our optimization problem in a standard form of a network problem, such as the shortest path problem, the max flow problem, the minimum cost flow problem, or the routing problem. Even multi-commodity flow problems, which are discussed, e.g., in [Ber98, OMV00, Min01], differ considerably from our model. The discrepancy lies mainly in the fact that, in our model, the quantities of different products are required to flow to a node (plant) in a certain ratio that is determined by the production process of the plant. Furthermore, in our model, commodities are transformed while flowing through the network. Concepts and methods of network analysis and graph theory, cf. [Kol09, BE05, New10, EK10], focus on the analysis of network structures and on revealing of patterns of relations. Since it is not possible to regard our model in a network form without losing information, methods of these fields are not readily applicable for our purpose, as they could lead to wrong conclusions. However, it will turn out that, after certain reformulations, tools from graph theory can be used to detect weakly connected parts of the optimization model. These are often well-suited for aggregation.

In the field of numerical optimization, reduction methods arise for solving problems with (linear) equality constraints. According to our knowledge, there are no explicit reduction methods for nonlinear optimization problems with inequality constraints. However, decomposition methods are used for the numerical solution of large-scale problems that show a certain structure [Las70, Geo72, CCMGB06, PC06, CLCD07]. They separate

the optimization of several subproblems that are linked by a master problem. Therefore, we will refer to the variables of the master problem as links. In general, the choice of links and the corresponding subproblems is not predefined. [PC06, CLCD07] initialize a discussion about the possible impact of architectural decisions in the context of decomposition methods. In this work, we will propose an approach to reveal and compare different decompositions of a problem by considering graph-theoretical aspects. In this sense, our approach is supposed to be rather a tool for a global model analyses than an optimization method.

Aims of this Thesis

The major goals of this thesis can be summarized as follows:

- We aim to develop a detailed optimization model for a multi-commodity market that makes the analysis of price formation, sensitivity analyses, and also various scenario simulations possible. We test this model on real data of the petrochemical market. The products of this market are connected by chemical reaction chains, which form the basis of the production technologies within the chosen market. As far as available, we will use real historical data as model parameters. If real data is missing, we substitute it by reasonable estimates.

- The main objective of this thesis is the complexity and model reduction within the proposed multi-commodity market model. Model reduction is not merely beneficial to ease and speed up the numerical solution of the optimization problem, but rather needed for applications since obtaining detailed data about all market components is often expensive, time-consuming, or impossible. Furthermore, the answer to a certain question usually does not depend on detailed information about all market components. Therefore, we aim for developing a method that facilitates the differentiation between parameters that strongly influence the results of a simulation with respect to a certain question and parameters that have little to no effect. Furthermore, we aim for revealing the overall problem structure and for detecting and aggregating model components that are weakly connected to the remaining part of the model.

- As far as we know, there is no suitable mathematical method for analyzing the structure of constrained nonlinear optimization problems. Therefore, we aim to develop a general approach to expose the structure of an optimization problem and to approximate suitable components of a problem. For this purpose, we intend to represent the connections between all variables in a plain way.

Main Contributions and Results

A multi-commodity supply-demand trade model for simulating price formation:
In cooperation with Holger Diedam, Willi Jäger, Sabrina Kellner, and Sebastian Sager, who are also involved in the IWR's Pioneering Project about modeling and simulation of price dynamics in commodity markets, we develop a multi-commodity supply-demand trade model. Since simulated prices and sales of three major regions, by tendency, conform with historical market data over eight annual periods, our modeling approach appears to be reasonable. This thesis provides an algorithm for setting up a consistent model to simulate prices of selected products.

A graph-theoretical decomposition approach to the analysis of large constrained optimization problems with a block-separable structure:
We propose a new approach to the analysis of optimization problems with a block-separable structure by means of complexity reduction. This approach is based on the representation of the connections between primal and dual variables on a hypergraph. We show that each vertex-separator of the hypergraph corresponds to the set of links of a primal decomposition of the optimization problem. Furthermore, each net-separator of the hypergraph implies the set of links of a favorable dual decomposition. This means that the proper transfer of possible graph partitions to the original optimization problem leads to different decompositions. Our approach can be used to explore and compare decompositions of the overall problem into different sets of subproblems and to identify particularly balanced decompositions.
Since the subproblems are parametric optimization problems in the links, those with only a few links can be approximated in terms of approximating their optimal value functions. For this purpose, a subproblem must be solved for a representative set of values of its links. The corresponding optimization results of the subproblem lead simultaneously to an extended overview of the possible submodel's solutions and their sensitivities. This overview does not require any parameter outside the submodel to be fixed. In this sense, our approach allows a global sensitivity analysis within the subproblems. Subproblem aggregations can be used to approximate solutions of the overall problem for varying parameters outside the submodel. Furthermore, the decomposition considerably facilitates the interpretation of the overall optimization results.

A tailored decomposition approach to reduce the complexity of a multi-commodity model:
We present a tailored version of the general decomposition approach, which can be persued to identify nearly independent subsystems of a multi-commodity market model for one region and one time period and to appropriately aggregate these components. In this regard, we develop an algorithm for the automated identification of linking products and corresponding subsystems that are suited for an aggregation. The algorithm, firstly, converts the network of processes and products to a smaller, undirected graph,

secondly, partitions this graph based on methods of graph theory and, finally, transfers the decomposition of the simplified graph back to the optimization model.
Each detected subsystem can be optimized independently from the remaining part of the system given fixed input/output quantities of certain linking products. Solving a subproblem for varying input/output quantities allows us to approximate an aggregated input/output-profit/cost relationship for the whole subsystem. We show that the relevant domain of an input/output-profit/cost function can, in some cases, be restricted by taking into account the economic interpretation of certain dual variables as shadow prices. These dual variables are the Lagrange multipliers that correspond to the input/output constraints of a subproblem.

Various aspects of a sensitivity analysis within a decomposed market model:
The variation of the input/output quantities within the relevant domain represents the remaining system's accumulated possible impact on a subsystem. Therefore, the solution of a subproblem for varying input/output quantities yields a detailed overview of its sensitivities. Given a set of subsystem parameters and optimal subsystem solutions for varying input/output quantities, we determine a set of perturbations of external prices that do not cause any change in the primal optimal solution of a subsystem as long as all other subsystem parameters are fixed. In a similar way, one can deduce such ranges for capacity limits. The consideration of sensitivities within the subsystem with respect to the input/output quantities and with respect to subsystem parameters shows to which extent submodel solutions are influenced by submodel parameters and how far they depend on input/output quantities. Furthermore, the effect of subsystem parameters on the optimal solution of the remaining part of the system can be analyzed by means of the optimal value sensitivities within the subproblem, since subproblem parameters influence the remaining system solely through the input/output-profit/cost relationship. Summarizing, our approach enables the identification of those submodel parameters that have little influence on certain model components, which facilitates the data procurement regarding external prices and process capacities.

The application to a real multi-commodity market:
Applying our approach to a model for a part of the petrochemical market, which includes four major subsystems, we come to approximated solutions of the original problem, which are promising. Sensitivity analysis shows that certain external prices can vary in a broad range without influencing the overall result. Others have high impact on the optimal solutions. Furthermore, we obtain the linear separability of two submodels' input/output-profit/cost functions. Further analysis of the respective subproblems reveals that certain model variables can be fixed as long as submodel parameters do not vary. Regarding the graphical representation of the overall optimization problem, this fixing of variables leads to a subgraph of the original graph. This subgraph can again be partitioned and discloses a new favorable decomposition of the overall optimization problem into subproblems.

Thesis Overview

In *chapter 2*, we present a new supply-demand trade model for the simulation of price formation in a multi-commodity market. In view of the complexity of a large commodity network, we propose a deterministic optimization model. The whole market is regarded as a monopoly, in which producers set prices by maximizing their cumulative profit given the price-demand relationship for all products. We discuss the determination of appropriate sets of products and processes to simulate the prices of selected products and propose an algorithm for setting up consistent models. Finally, we apply our model to a small part of the petrochemical market and propose three different approaches to simulate the price formation over multiple time periods. They differ in the choice of the parameters that model the dependence of a commodity's present demand function on its previous prices and quantities of sales.

Chapter 3 gives a short introduction to the theory of constrained nonlinear optimization. It summarizes main sensitivity and stability results concerning parametric optimization problems. Furthermore, we introduce duality theory and several results about nonsmooth convex optimization, which are fundamental to decomposition methods for optimizing large-scale systems. Finally, we present an interior point method as standard numerical algorithm for solving constrained nonlinear problems and introduce least squares methods for parameter estimation.

Chapter 4 presents important concepts of graph theory with a special focus on providing a basis for (hyper-)graph partitioning. Moreover, we provide relevant definitions and results about graph connectivity as well as different centrality measures and measures for graph fragmentation.

In *chapter 5*, we present a new approach to the analysis and reduction of large constrained optimization problems that have a block-separable structure. This approach is closely related to the decomposition methods for the numerical solution of such problems. We show how the overall structure of a problem with coupling constraints can be exposed on a bipartite graph or a hypergraph that captures the connectivity of primal and dual variables. These graphs can be analyzed and beneficially partitioned using suitable graph theoretical methods. We discuss the disclosure of advantageous decompositions on two examples. Furthermore, we present general results concerning the relationship between the possible decompositions of a block-separable optimization problem and the possible partitions of the hypergraph that represents the connectivity of the problem's variables. Finally, we discuss the approximation of subproblems.

In *chapter 6*, we adapt the decomposition approach of chapter 5 to a specific case of the multi-commodity market model of chapter 2 and formulate resulting subproblems. We present first numerical results of aggregations within a small consistent production system, which is a part of the petrochemical market. Regarding larger models, we discuss possibilities to automatically identify subsystems that are suited for aggregation

and present a resulting decomposition of an extended real market model. Furthermore, this chapter addresses the approximation of a subproblem, which includes determining a suitable domain on which the subproblem's optimal value function is approximated, generating subsystem data, and choosing a parametric model. We discuss and compare sensitivity analysis within the overall model and within subsystems. We show how sensitivity analysis can justify model simplifications and how it supports the task of data procurement. Finally, we present the numerical results of applying our approach to an extended petrochemical production system.

In *chapter 7*, we summarize the conclusions of this thesis and give an outlook on further directions of research to be taken in this field.

2 Modeling and Simulation of Price Formation in a Multi-Commodity Supply-Demand Trade Network

In this chapter, we present an optimization model to explain and simulate price formation in a multi-commodity market, at which products are directly connected by operational processing functions and the regionally distinct prices are the result of the transregional interplay between producers and consumers.[1] With this model, we intend to provide a rational approach to operational planning and risk management that supports the analysis and prognosis of prices, various scenarios and sensitivity analysis, as well as the identification of the essential market and price drivers. This new approach to market and price modeling differs in several aspects from the existing scientific fields of research that are related to commodity markets. We introduce these fields in the following.

For *risk management*, *portfolio optimization*, and *derivative pricing* in commodity markets, usually stochastic models are applied [Lud13, Bod12, Ruj08, Gem05, Pil07, BK04]. These models were originally developed to model stock prices and interest rates, and are currently extended to commodity prices. According to their purpose, they include few assets and do not take the system aspect of price formation into account. Technologies that connect different commodities in a deterministic way are neglected. However, the latter seems crucial for a comprehensive understanding of the price formation. Therefore, in contrast to the pure modeling of price data, we will explicitly model the interplay between supply and demand.

In the field of *revenue management* and *dynamic pricing*, expected revenues are maximized by dynamically choosing prices over a time period given a predefined maximal sales volume for this period and stochastically varying demand, cf. for example, [GvR97, BC03]. Single companies of monopolistic and oligopolistic markets apply the corresponding scientific models and methods aiming for a price discrimination that captures the market's consumer surplus. This topic is of special interest for sectors with high fixed costs of capacities and low variable costs, such as the airline industry or the hotel industry.

In contrast to these research fields, optimization models are used for process planning in the *operations research* areas like *production optimization* [Hac08, Ras11], *supply chain management* [SK07, VW03], and *chemical engineering* [KC08, AQE09, Hüb07]. In this respect, modeling of technologies is obviously essential. However, the variables of these

[1] This model has been developed in the context of IWR's Pioneering Project about modeling and simulation of price dynamics in commodity markets, which is also mentioned in the introduction of this thesis. Therefore, the model that we present in sections 2.1 to 2.3 similarly appears in Sabrina Kellner's PhD thesis, "Modeling and Analysis of Demand for Commodities and a Case Study of the Petrochemical Market", [Kel13].

models are commonly only production-related quantities, while product prices are mostly included as parameters. Prices are not supposed to be variable, since, usually, the models optimize the strategic actions of smaller market participants without influence on prices. Furthermore, demand is mostly considered price-inelastic, i.e., as a constant quantity. The results are linear programs (sometimes with mixed integer decisions), whereas our model yields a nonlinear optimization problem with more types of variables (production quantities, product prices, interregional transport) and more complex constraints between them. Because of its particular importance for several industries, the topic of integrating pricing decisions and production planning has begun to attract attention in the scientific operations research community in the last few years. However, so far, only a small number of works addresses supply optimization approaches that include pricing [Cha05, LB08, Kan08, KGvB$^+$09, TBS12].[2]

The following new optimization model for price formation applies to a market that is characterized on the supply side by a multiplicity of *processes* transforming some resource *products* into outputs. The predominant variable costs of the production are those of the resources, so that we only consider those in the following. The ratio between inputs and outputs of a process is uniquely described by factors for inputs and outputs, where input factors are negative, output factors are positive and the factor of the major output product of a process is set to one. The processes are operated at different *plants* that are located in certain *regions*. The production of a plant in a certain *time period* is limited by its capacity. Running a plant will cause some fixed costs in the corresponding time period.
We assume that the market's demand side is given by *consumers* whose buying behavior regarding a product (in a certain region, during a certain time period) is representable by a functional relation between the ordered quantity and product prices.
Taking into account the supply and the demand side, we categorize the products in three groups: *products with external demand*, *intermediates without external demand* and *external products*. The first class consists of all products with "external demand"; by external demand, we refer to the demand from industrial sectors outside the observed market. We model this demand as consumer's demand. Intermediates are products that are produced as well as processed. We assume them to appear only in the market under observation. The part of them without external demand, forms our second class of products. The third class of external products consists of all remaining products. They appear only as resources or outputs of the processes. Under the assumption that their prices form outside the observed market and that buying or selling them does not influence their prices, we integrate their prices in our model as parameters.
Furthermore, we assume that transport within a region does not cause any costs, whereas interregional transportation costs are proportional to transport quantities. When con-

[2] A special case is the electricity market. Fundamental market models for this particular commodity usually include an explicit modeling of demand and of price drivers as natural gas prices, coal prices, and the weather, cf. [BGS07, EW03, Kra09].

sidering multiple time periods, we have to take storage into account. This means that we include storage capacities and storage costs as parameters in our model. Transportation and storage are only considered for products that are produced in the observed market, i.e., not for external products, since we assume the markets of external products to be arbitrage free[3]. Within one time period, we leave the chronological order of production processes out of consideration.[4]
Under these assumptions, in this work, we consider the overall optimization of the total supply side given commonly known price-demand relationships for all products. The price-demand relationship of a certain product is built by the aggregation of all consumers' price-dependent demand for it. In the sense of microeconomics, we deal with the problem of profit maximization in a monopoly [PR10, Var10]. The variables of this optimization problem are production quantities, prices of products with external demand, as well as transportation quantities. When optimizing over multiple time periods, storage quantities appear as further variables. Finally, our price simulations for the products with external demand are given by the optimal solutions of the optimization problem.
Note that our model does not include long-term contracts between suppliers and customers, which exist in reality. However, such contracts can be integrated into our modeling by additional parameters and constraints.[5] Furthermore, the multiple time period model does not consider the possibility to destroy surplus production if storage capacities are exhausted or if storage is not economic.
In section 2.1, we define all sets, parameters, variables, and the demand function, which are needed for formulating the optimization problem. Then, in section 2.2, for a clearer representation, we present the objective function and constraints in the reduced case of optimizing one single time period. Section 2.3 addresses the generalized case of optimizing over a time interval subdivided in multiple time periods. Section 2.4 discusses the construction of consistent network models. In section 2.5, we present simulation results for a selected part of the petrochemical market.

2.1 Definitions

Sets

$\mathcal{P}$ products $(\mathcal{P} = \mathcal{P}_{ex} \dot{\bigcup} \mathcal{P}_{out} \dot{\bigcup} \mathcal{P}_{mid})$

[3] A market is called *arbitrage free* if prices permit taking a risk-free advantage of price differences.
[4] By this simplification, in some special cases, production can become possible without any resources. For instance, this can occur in a consistent model, cf. definition 2.4.1, that is a closed circuit of products with external demand and intermediates, i.e., a consistent production system that does not include external products. However, in real applications, such a closed circuit should not exist, since, some time after an external initialization it is either not able to produce anything or it can endlessly produce output without any resources.
[5] An example for integrating contracts into the price-demand relationship can be found in [Kel13].

$\mathcal{P}_{ex}$	external products (prices are given from outside the network)
$\mathcal{P}_{out}$	products with external demand
$\mathcal{P}_{mid}$	intermediates without external demand
$\mathcal{R}$	regions
$\mathcal{I}$	plants
$\mathcal{S}$	processes
$\mathcal{C}$	consumers
$\mathcal{T}$	time periods

Parameters

a) Plant and process related parameters:

$a_{i,r}^{\text{plant\_reg}} \in \{0,1\}$	is equal to one if plant i is located in region r
$a_{i,s,t}^{\text{cap}} \geq 0$	capacity of process s at plant i in period t
$a_{i,s,t}^{\text{fixed\_costs}} \geq 0$	fixed costs of process s at plant i in period t
$a_{s,p,t}^{\text{f}}$	factor of product p in process s during period t, $a_{s,p,t}^{\text{f}} < 0$ if p is an input of s, $a_{s,p,t}^{\text{f}} > 0$ if p is an output of s

b) External prices:

$a_{p_{ex},r,t}^{\pi} \geq 0$	price of product p_{ex} in region r in time period t

c) Demand related parameters:

$a_{c,p_{out},r,t}^{\text{con\_last}} \geq 0$	consumer c's consumption of product p_{out} in region r in period $t-1$
$a_{p_{out},r,t}^{\pi\text{\_last}} \geq 0$	price of product p_{out} in region r in period $t-1$
$a_{c,p_{out},r,t}^{\text{con\_max}} > a_{c,p_{out},r,t}^{\text{con\_last}}$	maximum consumption of product p_{out} that we assume for consumer c in region r in period t
$a_{c,p_{out},r,t}^{\pi\text{\_max}} > a_{p_{out},r,t}^{\pi\text{\_last}}$	assumed maximum price (paid by consumer c) for product p_{out} in region r in period t
$a_{r,t}^{GDP}$	change in the gross domestic product of region r from $t-1$ to t (cf. (2.5))
$a_{r,t}^{IndPro}$	change in a certain industrial production index of region r from $t-1$ to t (cf. (2.5))

d) Transport and storage related parameters:

$a_{r_1,r_2,t}^{\text{tr}} \geq 0$	transport costs to deliver one unit from region r_1 to r_2 in period t
$a_{p,r,t}^{\text{c\_stor}} \geq 0$	costs of storing one unit of product $p \in \mathcal{P}_{out} \dot{\bigcup} \mathcal{P}_{mid}$ in region r from period t to period $t+1$
$a_{p,r,t}^{\text{stor\_cap}} \geq 0$	storage capacity for product $p \in \mathcal{P}_{out} \dot{\bigcup} \mathcal{P}_{mid}$ in region r from period t to $t+1$

Variables

$x^{\mathrm{q}}_{i,s,t} \geq 0$	production quantity of process s at plant i in period t
$x^{\pi}_{p_{out},r,t} \geq 0$	price of product p_{out} in region r in period t
$x^{\mathrm{tr}}_{p,r_1,r_2,t} \geq 0$	transport quantity of product $p \in \mathcal{P}_{out} \dot{\bigcup} \mathcal{P}_{mid}$ from r_1 to r_2 in period t
$x^{\mathrm{stor}}_{p,r,t} \geq 0$	quantity of product $p \in \mathcal{P}_{out} \dot{\bigcup} \mathcal{P}_{mid}$ stored in region r from t to $t+1$
$x^{\pi}_{p_{mid},r,t} \geq 0$	price of product p_{mid} in region r at time t

Remark 2.1.1. The prices $x^{\pi}_{p_{mid},r,t}$ for the intermediate products that are not sold to consumers cannot be uniquely determined by our model. They depend on the division of the overall profit over the supply side. However, by the assumption that each single process is profitable, we get price ranges in which these prices lie. The profitability of each process is guaranteed by the constraints (2.14). Furthermore, Lagrange multipliers corresponding to the production-transport constraints of intermediates, (2.11), indicate the equilibrium value of these products from the supply side's point of view.[6]

Demand Function

Assuming economical behavior of the consumers and keeping in mind the applicability of the demand functions $\phi : \mathbb{R}^+_0 \to \mathbb{R}^+_0, x \mapsto \phi(x)$ in the context of our optimization model, we choose a class of functions that fulfill the following properties:[7]

- ϕ is bounded,
- ϕ has a zero,
- ϕ is strictly decreasing outside of the domain where it equals zero,
- ϕ is twice continuously differentiable outside of the domain where it equals zero.

Taking these properties into account, we model the basic demand function by

$$\phi^{\mathrm{basic}}_{c,p_{out},r,t}(x^{\pi}_{p_{out},r,t}) = \max\left[a^1_{c,p_{out},r,t} \cdot \tanh\Big(\frac{a^2_{c,p_{out},r,t} - x^{\pi}_{p_{out},r,t}}{a^3_{c,p_{out},r,t}}\Big), 0\right] \tag{2.1}$$

where $a^1_{c,p_{out},r,t}, a^2_{c,p_{out},r,t}, a^3_{c,p_{out},r,t} > 0$ are given by

$$\begin{aligned} &a^1_{c,p_{out},r,t} = a^{\mathrm{con\_max}}_{c,p_{out},r,t}\,, \quad a^2_{c,p_{out},r,t} = a^{\pi\_\mathrm{max}}_{c,p_{out},r,t} \quad \text{and} \\ &a^3_{c,p_{out},r,t} = \Big(a^{\pi\_\mathrm{max}}_{c,p_{out},r,t} - a^{\pi\_\mathrm{last}}_{p_{out},r,t}\Big) \Big/ \operatorname{artanh}\Big(\frac{a^{\mathrm{con\_last}}_{c,p_{out},r,t}}{a^{\mathrm{con\_max}}_{c,p_{out},r,t}}\Big). \end{aligned} \tag{2.2}$$

[6] This is due to the economic interpretability of Lagrange multipliers as shadow prices. An additionally available unit of an intermediate in a certain region would increase the optimal objective value, i.e., the supply side's profit, approximately by the value of the respective constraint's Lagrange multiplier.

[7] Contrary to this assumed consumer behavior, there is the special case of conspicuous consumption goods, which are bought to publicly display economical power, e.g., a luxury car. The demand for these products is usually not monotonically decreasing in price, cf. [HFH+11]. However, commodities obviously do not fall in this category of products.

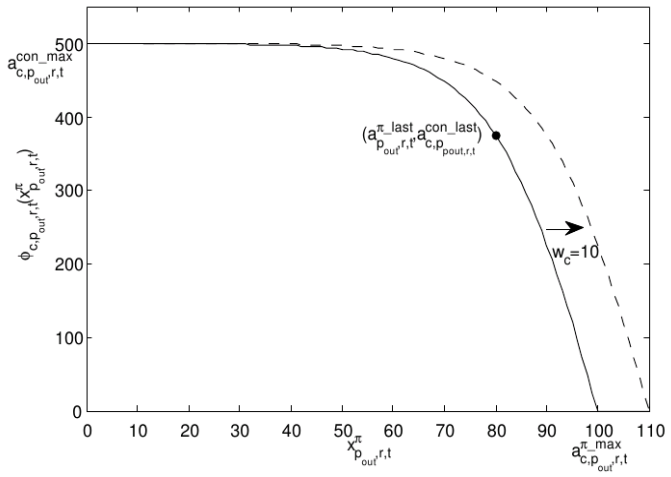

Figure 2.1: *Shape of the demand function* $\phi_{c,p_{out},r,t}$

The shape of such a demand function is sketched in figure 2.1 by the solid line. Given its basic form by (2.1) and maximal values for consumption ($a^{\text{con\_max}}_{c,p_{out},r,t}$) and price ($a^{\pi\_\text{max}}_{c,p_{out},r,t}$), the third parameter $a^3_{c,p_{out},r,t}$ of the demand function is chosen such that the point $(a^{\pi\_\text{last}}_{p_{out},r,t}, a^{\text{con\_last}}_{c,p_{out},r,t})$, i.e., previous year's price-demand relationship observed at the market, lies on the curve.

Remark 2.1.2. Note that we have $\phi_{c,p_{out},r,t}(0) < a^{\text{con\_max}}_{c,p_{out},r,t}$ because $a^{\text{con\_max}}_{c,p_{out},r,t}$ is the limiting value of the scaled hyperbolic tangent.

Motivated by the historical data, we extend the demand function by a dependency on economic factors, namely the gross domestic product (GDP) and an industrial production index of each region:

$$\phi_{c,p_{out},r,t}(x^{\pi}_{p_{out},r,t}) = \max\left[a^1_{c,p_{out},r,t}\cdot\tanh\Big(\frac{a^2_{c,p_{out},r,t} + w_{c,p_{out},r}(a^{GDP}_{r,t}, a^{IndPro}_{r,t}) - x^{\pi}_{p_{out},r,t}}{a^3_{c,p_{out},r,t}}\Big), 0\right], \tag{2.3}$$

where

$$w_{c,p_{out},r}(a^{GDP}_{r,t}, a^{IndPro}_{r,t}) = \alpha_{c,p_{out},r}\cdot a^{GDP}_{r,t} + \beta_{c,p_{out},r}\cdot a^{IndPro}_{r,t} \tag{2.4}$$

with $\alpha_{c,p_{out},r}$ and $\beta_{c,p_{out},r}$ estimated by a least squares method (cf. section 3.7) and

$$a^{GDP}_{r,t} = GDP_{r,t} - GDP_{r,t-1}, \qquad a^{IndPro}_{r,t} = IndPro_{r,t} - IndPro_{r,t-1}. \tag{2.5}$$

This means that we estimate regression parameters $\alpha_{c,p_{out},r}$ and $\beta_{c,p_{out},r}$ corresponding to the influence of changes in GDP and the production index on the consumer's willingness to pay. For instance, if $\alpha_{c,p_{out},r} > 0$ and $\beta_{c,p_{out},r} > 0$, an increase in the GDP and the production index leads to a right shift of the demand curve. Such a right shift is illustrated in figure 2.1 by the dashed line.

[Kel13] provides an elaborate study of demand modeling. It proposes an extended demand model, which includes (2.3) as special case, and discusses methods for parameter identification.

Remark 2.1.3. Besides the above mentioned basic properties, the demand functions (2.1) and (2.3) have the following desirable property, which confirms the modeling by means of the hyperbolic tangent function:
We define for each consumer $c \in \mathcal{C}$ a constant k_c as the quotient of his maximum consumption and his previous year consumption, i.e., we have $a^{\text{con\_max}}_{c,p_{out},r,t} = k_c \cdot a^{\text{con\_last}}_{c,p_{out},r,t}$. Then, the aggregated demand of multiple consumers $c_1, ..., c_n$ with the same demand behavior equals the demand of one consumer c^* with the previous year consumption $a^{\text{con\_last}}_{c^*,p_{out},r,t} = \sum_{i=1}^{n} a^{\text{con\_last}}_{c_i,p_{out},r,t}$ and the same demand behavior. We mean with "same demand behavior" that

$$a^{\pi\_\max}_{c_i,p_{out},r,t} = a^{\pi\_\max}_{c^*,p_{out},r,t}, \quad w_{c_i,p_{out},r}(a^{GDP}_{r,t}, a^{IndPro}_{r,t}) = w_{c^*,p_{out},r}(a^{GDP}_{r,t}, a^{IndPro}_{r,t}) \quad \forall\, i = 1, ..., n, \tag{2.6}$$

and that a constant $k > 1$ exists with

$$a^{\text{con\_max}}_{c_i,p_{out},r,t} = k \cdot a^{\text{con\_last}}_{c_i,p_{out},r,t} \quad \forall\, i \quad \text{and} \quad a^{\text{con\_max}}_{c^*,p_{out},r,t} = k \cdot a^{\text{con\_last}}_{c^*,p_{out},r,t}, \tag{2.7}$$

i.e., $k_c = k \;\forall\; c \in \mathcal{C}$. We prove this by the following calculation:

$$\begin{aligned}
&\sum_{i=1}^{n} \phi_{c_i,p_{out},r,t}(x^{\pi}_{p_{out},r,t}) \\
&\quad = \sum_{i=1}^{n} \max\Big[a^{\text{con_max}}_{c_i,p_{out},r,t} \cdot \tanh\Big(\frac{a^{\pi_\max}_{c_i,p_{out},r,t} + w_{c_i,p_{out},r}(a^{GDP}_{r,t}, a^{IndPro}_{r,t}) - x^{\pi}_{p_{out},r,t}}{\Big(a^{\pi_\max}_{c_i,p_{out},r,t} - a^{\pi_\text{last}}_{p_{out},r,t}\Big) \Big/ \operatorname{artanh}\Big(\frac{a^{\text{con_last}}_{c_i,p_{out},r,t}}{a^{\text{con_max}}_{c_i,p_{out},r,t}}\Big)}\Big), 0\Big] \\
&\quad = \sum_{i=1}^{n} a^{\text{con_max}}_{c_i,p_{out},r,t} \cdot \max\Big[\tanh\Big(\frac{a^{\pi_\max}_{c^*,p_{out},r,t} + w_{c^*,p_{out},r}(a^{GDP}_{r,t}, a^{IndPro}_{r,t}) - x^{\pi}_{p_{out},r,t}}{\Big(a^{\pi_\max}_{c^*,p_{out},r,t} - a^{\pi_\text{last}}_{p_{out},r,t}\Big) \Big/ \operatorname{artanh}(1/k)}\Big), 0\Big] \\
&\quad = a^{\text{con_max}}_{c^*,p_{out},r,t} \cdot \max\Big[\tanh\Big(\frac{a^{\pi_\max}_{c^*,p_{out},r,t} + w_{c^*,p_{out},r}(a^{GDP}_{r,t}, a^{IndPro}_{r,t}) - x^{\pi}_{p_{out},r,t}}{\Big(a^{\pi_\max}_{c^*,p_{out},r,t} - a^{\pi_\text{last}}_{p_{out},r,t}\Big) \Big/ \operatorname{artanh}(1/k)}\Big), 0\Big] \\
&\quad = \phi_{c^*,p_{out},r,t}(x^{\pi}_{p_{out},r,t})
\end{aligned} \tag{2.8}$$

2.2 The Optimization Problem for a Single Time Period

Let us consider a single time period t. With the above definitions, we are able to formulate the problem of optimizing the total profit of the supply side given the demand functions of the consumers. This profit is composed of the total revenue of the system minus the costs. The total revenue, in turn, consists of the revenues from selling the

products $p_{out} \in \mathcal{P}_{out}$ at prices $x^{\pi}_{p_{out},r,t}$ and from selling some of the products $p_{ex} \in \mathcal{P}_{ex}$ at fixed prices $a^{\pi}_{p_{ex},r,t}$. Sales quantities are related to production quantities $x^{\mathrm{q}}_{i,s,t}$ and transport quantities $x^{\mathrm{tr}}_{p,r_1,r_2,t}$ through the constraints (2.12). Lower bounds to prices $x^{\pi}_{p_{out},r,t}$ of products with external demand are indirectly given by total production capacities, since, in an optimum of our monopolistic problem, the aggregated price-dependent demand, $\sum_{c\in\mathcal{C}} \phi_{c,p_{out},r,t}(x^{\pi}_{p_{out},r,t})$, is satisfied for all products $p_{out} \in \mathcal{P}_{out}$ and all regions $r \in \mathcal{R}$. This is because a monopolist is able to perfectly adjust production and sales quantities: given his favorite sales quantities, he chooses the highest possible prices that induce the respective necessary demand. These facts motivate our modeling of the dependence between prices, sales, production, and transport quantities in (2.9) and (2.12).[8] The costs of the system are made up of variable costs from buying feedstock at fixed prices $a^{\pi}_{p_{ex},r,t}$, of transportation costs, and of fixed costs for running plants. In the case of optimizing only one single time period, there is no incentive to storage.
In addition to the mentioned constraints (2.12), there are further side conditions on the choice of the production, price, and transport variables. We will list them after formulating the objective function.[9]

Objective function (for time period t)

$$\begin{aligned}\max_{\substack{x^{\mathrm{q}}_{i,s,t},\ x^{\pi}_{p_{out},r,t}\\ x^{\mathrm{tr}}_{p_{out},r_1,r_2,t},\ x^{\mathrm{tr}}_{p_{mid},r_1,r_2,t}}} \quad & \sum_{\substack{p_{out}\in\mathcal{P}_{out}\\ r\in\mathcal{R}}} x^{\pi}_{p_{out},r,t}\cdot\Big(\sum_{c\in\mathcal{C}}\phi_{c,p_{out},r,t}(x^{\pi}_{p_{out},r,t})\Big)\\ & + \sum_{\substack{p_{ex}\in\mathcal{P}_{ex}\\ i\in\mathcal{I},\ s\in\mathcal{S},\ r\in\mathcal{R}}} x^{\mathrm{q}}_{i,s,t}\cdot a^{\mathrm{f}}_{s,p_{ex},t}\cdot a^{\mathrm{plant\_reg}}_{i,r}\cdot a^{\pi}_{p_{ex},r,t}\\ & - \sum_{\substack{p\in\mathcal{P}_{mid}\bigcup\mathcal{P}_{out}\\ r_1\in\mathcal{R},\ r_2\in\mathcal{R}}} x^{\mathrm{tr}}_{p,r_1,r_2,t}\cdot a^{\mathrm{tr}}_{r_1,r_2,t} - \sum_{i\in\mathcal{I},s\in\mathcal{S}} a^{\mathrm{fixed\_costs}}_{i,s,t}\cdot\mathbb{1}_{\{0<x^{\mathrm{q}}_{i,s,t}\}}\end{aligned} \tag{2.9}$$

Constraints (for time period t)

- Capacity Constraints:

  $\forall\ i\in\mathcal{I}, \forall\ s\in\mathcal{S}$

  $$x^{\mathrm{q}}_{i,s,t} \le a^{\mathrm{cap}}_{i,s,t} \tag{2.10}$$

  Production is bounded by plant capacities.

[8] [LB08]'s integrated production planning and pricing model includes a stochastic demand component. Therefore, the producer cannot perfectly adjust his production to the demand that corresponds to a chosen price. In this context, the difference between demand and sales is referred to as lost demand if the demand at the chosen price is greater than sales.

[9] Regarding definition 3.1, we should consider an objective function defined on an open domain. Since price variables $x^{\pi}_{p_{out},r,t}$ are constrained to be non-negative, we could define the demand functions $\phi_{c,p_{out},r,t}$ on $\mathbb{R}$ instead of $\mathbb{R}^{+}_{0}$ without any influence on optimal solutions.

- Production-transport constraints:

  $\forall\ p_{mid} \in \mathcal{P}_{mid}, \forall\ r_1 \in \mathcal{R}$

  $$\sum_{r_2 \in \mathcal{R}} \left(x^{\mathrm{tr}}_{p_{mid},r_1,r_2,t} - x^{\mathrm{tr}}_{p_{mid},r_2,r_1,t} \right) \leq \sum_{i \in \mathcal{I}, s \in \mathcal{S}} x^{\mathrm{q}}_{i,s,t} \cdot a^{\mathrm{f}}_{s,p_{mid},t} \cdot a^{\mathrm{plant\_reg}}_{i,r_1} \tag{2.11}$$

  Net production of an intermediate in a region must exceed this intermediate's net export of the region.

- Sales-production-transport constraints:

  $\forall\ p_{out} \in \mathcal{P}_{out}, \forall\ r_1 \in \mathcal{R}$

  $$\sum_{c \in \mathcal{C}} \phi_{c,p_{out},r_1,t}(x^{\pi}_{p_{out},r_1,t}) + \sum_{r_2 \in \mathcal{R}} \left(x^{\mathrm{tr}}_{p_{out},r_1,r_2,t} - x^{\mathrm{tr}}_{p_{out},r_2,r_1,t} \right) \leq \sum_{i \in \mathcal{I}, s \in \mathcal{S}} x^{\mathrm{q}}_{i,s,t} \cdot a^{\mathrm{f}}_{s,p_{out},t} \cdot a^{\mathrm{plant\_reg}}_{i,r_1} \tag{2.12}$$

  Net production of a product with external demand in a region must exceed the sum of sales in the region plus the net export of the product.

- No-arbitrage constraints:

  $\forall\ p \in \mathcal{P}_{mid} \bigcup \mathcal{P}_{out}, \forall\ r_1 \neq r_2 \in \mathcal{R}$

  $$x^{\pi}_{p,r_1,t} \leq x^{\pi}_{p,r_2,t} + a^{\mathrm{tr}}_{r_2,r_1,t} \tag{2.13}$$

  To avoid arbitrage, inter-regional price differences must not exceed transport costs.

- Constraints for prices of intermediates:

  $\forall\ s \in \mathcal{S}, \forall\ r \in \mathcal{R}$

  $$0 \leq \sum_{i \in \mathcal{I}} x^{\mathrm{q}}_{i,s,t} \cdot a^{\mathrm{plant\_reg}}_{i,r} \cdot \Big(\sum_{p \in \mathcal{P}_{mid} \bigcup \mathcal{P}_{out}} a^{\mathrm{f}}_{s,p,t} \cdot x^{\pi}_{p,r,t} + \sum_{p_{ex} \in \mathcal{P}_{ex}} a^{\mathrm{f}}_{s,p_{ex},t} \cdot a^{\pi}_{p_{ex},r,t} \Big) \tag{2.14}$$

  Each running process is required to operate profitably.

Remark 2.2.1. This constrained nonlinear optimization problem is in general neither differentiable nor convex. Its non-differentiability is caused by the fixed costs for running plants and by the non-differentiability of the aggregated demand due to the maximum function in each consumer's demand: If there are at least two consumers c_1, c_2 with different maximal price values $a^{\pi\_\max}_{c_i,p_{out},r,t} + w_{c_i,p_{out},r}(a^{GDP}_{r,t}, a^{IndPro}_{r,t}), i = 1, 2$, for a product p_{out}, the aggregated demand function has a non-differentiable point in the relevant domain where the aggregated demand does not equal zero. This problem can possibly be avoided by bounding the prices by

$$x^{\pi}_{p_{out},r,t} \leq \min_{c_i \in \mathcal{C}} \left[a^{\pi\_\max}_{c_i,p_{out},r,t} + w_{c_i,p_{out},r}(a^{GDP}_{r,t}, a^{IndPro}_{r,t}) \right] \qquad \forall\ p_{out} \in \mathcal{P}_{out}. \tag{2.15}$$

However, in general, these bounds can get active and the solution of the original problem would be distorted.
The non-convexity is also caused by the aggregated demand functions that appear in the sales-production-transport constraints (2.12), and that include in each summand the maximum function. In section 6.1, we will show differentiability and convexity for a special case of the optimization problem.

2.3 Optimizing over Multiple Time Periods

In this model setting, we optimize the overall profit of multiple time periods $t_1, ..., t_n$ simultaneously. In this context, the parameters $a^{\pi\_\text{last}}_{p_{out},r,t_j}$ and $a^{\text{con\_last}}_{c,p_{out},r,t_j}$ (and, most appropriately, also $a^{\pi\_\max}_{c,p_{out},r,t_j}$ and $a^{\text{con\_max}}_{c,p_{out},r,t_j}$) are determined by the previous period's variables $x^{\pi}_{p_{out},r,t_{j-1}}$:
$\forall\ t_j \geq t_2$, we define

$$\begin{aligned} a^{\pi\_\text{last}}_{p_{out},r,t_j} &:= x^{\pi}_{p_{out},r,t_{j-1}} \\ a^{\text{con\_last}}_{c,p_{out},r,t_j} &:= \phi_{c,p_{out},r,t_{j-1}}(x^{\pi}_{p_{out},r,t_{j-1}}) \end{aligned} \tag{2.16}$$

That means, suppliers' price decisions of today affect the demand function of tomorrow, and therefore also suppliers' objective function of tomorrow.
Furthermore, in the multi-period context, suppliers are allowed to carry over produced products to subsequent time periods. This causes the extension of the profit function by storage costs. In addition, some constraints change by storage quantities as described below.

Objective function

$$\begin{aligned} \max_{\substack{x^{\text{q}}_{i,s,t_1},\dots,x^{\text{q}}_{i,s,t_n},\\ x^{\pi}_{p_{out},r,t_1},\dots,x^{\pi}_{p_{out},r,t_n},\\ x^{\text{tr}}_{p_{out},r_1,r_2,t_1},\dots,x^{\text{tr}}_{p_{out},r_1,r_2,t_n},\\ x^{\text{tr}}_{p_{mid},r_1,r_2,t_1},\dots,x^{\text{tr}}_{p_{mid},r_1,r_2,t_n}}} \quad & \sum_{t=t_1}^{t_n} \Bigg(\sum_{\substack{p_{out}\in\mathcal{P}_{out}\\ r\in\mathcal{R}}} x^{\pi}_{p_{out},r,t} \cdot \Big(\sum_{c\in\mathcal{C}} \phi_{c,p_{out},r,t}(x^{\pi}_{p_{out},r,t}) \Big) \\ & + \sum_{\substack{p_{ex}\in\mathcal{P}_{ex}\\ i\in\mathcal{I},s\in\mathcal{S},r\in\mathcal{R}}} x^{\text{q}}_{i,s,t} \cdot a^{\text{f}}_{s,p_{ex},t} \cdot a^{\text{plant\_reg}}_{i,r} \cdot a^{\pi}_{p_{ex},r,t} \\ & - \sum_{\substack{p\in\mathcal{P}_{mid}\bigcup\mathcal{P}_{out}\\ r_1\in\mathcal{R},\ r_2\in\mathcal{R}}} x^{\text{tr}}_{p,r_1,r_2,t} \cdot a^{\text{tr}}_{r_1,r_2,t} - \sum_{i\in\mathcal{I},s\in\mathcal{S}} a^{\text{fixed\_costs}}_{i,s,t} \cdot \mathbb{1}_{\{0<x^{\text{q}}_{i,s,t}\}} \\ & - \sum_{\substack{p\in\mathcal{P}_{mid}\bigcup\mathcal{P}_{out}\\ r\in\mathcal{R}}} x^{\text{stor}}_{p,r,t} \cdot a^{\text{c\_stor}}_{p,r,t} \Bigg) \end{aligned} \tag{2.17}$$

Constraints

- Recursive definition of the storage variables: production-transport-storage constraints:

  $\forall\ p_{mid} \in \mathcal{P}_{mid}, \forall\ r_1 \in \mathcal{R}, \forall\ t_j \in \mathcal{T}$ (where $x^{\text{stor}}_{p_{mid},r,t_0} := 0$)

  $$x^{\text{stor}}_{p_{mid},r_1,t_j} = x^{\text{stor}}_{p_{mid},r_1,t_{j-1}} + \sum_{i\in\mathcal{I},s\in\mathcal{S}} x^{\text{q}}_{i,s,t_j} a^{\text{f}}_{s,p_{mid},t_j} a^{\text{plant\_reg}}_{i,r_1} - \sum_{r_2\in\mathcal{R}} \left(x^{\text{tr}}_{p_{mid},r_1,r_2,t_j} - x^{\text{tr}}_{p_{mid},r_2,r_1,t_j}\right) \tag{2.18}$$

  Given the storage level of the previous period, new storage quantities of intermediates are directly determined by production and transportation variables.

- Recursive definition of the storage variables: sales-production-transport-storage constraints:

  $\forall\ p_{out} \in \mathcal{P}_{out}, \forall\ r_1 \in \mathcal{R}, \forall\ t_j \in \mathcal{T}$ (where $x^{\text{stor}}_{p_{out},r,t_0} := 0$)

  $$\begin{aligned} x^{\text{stor}}_{p_{out},r_1,t_j} = x^{\text{stor}}_{p_{out},r_1,t_{j-1}} &+ \sum_{i\in\mathcal{I},s\in\mathcal{S}} x^{\text{q}}_{i,s,t_j} \cdot a^{\text{f}}_{s,p_{out},t_j} \cdot a^{\text{plant\_reg}}_{i,r_1} \\ &- \sum_{c\in\mathcal{C}} \phi_{c,p_{out},r_1,t_j}(x^{\pi}_{p_{out},r_1,t_j}) - \sum_{r_2\in\mathcal{R}} \left(x^{\text{tr}}_{p_{out},r_1,r_2,t_j} - x^{\text{tr}}_{p_{out},r_2,r_1,t_j}\right) \end{aligned} \tag{2.19}$$

  Given the storage level of the previous period, new storage quantities of products with external demand are directly determined by production, price, and transport variables.

- Storage capacity constraints:

  $\forall\ p \in \mathcal{P}_{mid} \bigcup \mathcal{P}_{out}, \forall\ r \in \mathcal{R}, \forall\ t \in \mathcal{T}$

  $$(0 \leq)\ x^{\text{stor}}_{p,r,t} \leq a^{\text{stor\_cap}}_{p,r,t} \tag{2.20}$$

  Storage is bounded by product dependent storage capacities.

- Furthermore, $\forall\ t \in \mathcal{T}$, the constraints (2.10), (2.13), and (2.14) must be fulfilled.

Note that in case of multiple time periods, (2.11) and (2.12) are replaced by $x^{\text{stor}}_{p,r,t} \geq 0\ \forall\ p \in \mathcal{P}_{mid} \bigcup \mathcal{P}_{out}, r \in \mathcal{R}, t \in \mathcal{T}$ together with the recursive definition of storage variables in (2.18) and (2.19).

2.4 Setting up Consistent Network Models for Price Simulations

In this section, we discuss the question of how to set up a reasonable network model for one time period t starting with at least one product with external demand for which we intend to simulate price formation. Since our modeling approach is based on the

optimization of production and pricing decisions, it is essential that the model includes all production and processing facilities of certain relevant products. More precisely, to ensure proper simulations of prices and overall profit, the following requirements must be fulfilled:

a) For products with external demand (i.e., those for which we would like to simulate prices), all production facilities and, if available, all processing facilities have to be taken into account to allow for correct simulation of production quantities and costs.

b) Similarly, for intermediates without external demand, all production facilities and all processing facilities have to be taken into account to allow for correct simulation of production quantities and costs of following products with external demand.

Definition 2.4.1. *We call a network optimization model for simulating price formation* ***consistent*** *if requirements a) and b) are satisfied.*

Remark 2.4.2. Usually, a consistent network optimization model includes external products as resources. The special case of a consistent model without any external product is a closed production circuit of products with external demand and intermediates. In reality, such circuits should not exist, since, some time after an external initialization, they are either not able to produce anything or they are able to endlessly produce output without using any resources.

Let be given the global set of products $\mathcal{P}^0$, which are connected by processes of the set $\mathcal{S}^0$, where some products, $\mathcal{P}^0_{out} \subset \mathcal{P}^0$, are in demand of consumers and some other products, $\mathcal{P}^0_{ex} \subseteq \mathcal{P}^0 \setminus \mathcal{P}^0_{out}$, are natural resources. The products $\mathcal{P}^0_{mid} := \mathcal{P}^0 \setminus (\mathcal{P}^0_{out} \cup \mathcal{P}^0_{ex})$ are produced and processed.[10] These global sets constitute an inherently consistent model. To build up a smaller consistent model starting with some preselected products $\mathcal{P}_{\text{init}} \subset \mathcal{P}^0$ of which at least one has external demand, i.e., $\mathcal{P}_{\text{init}} \cap \mathcal{P}^0_{out} \neq \emptyset$, we pass through the following steps:

1. Identify all processes in which the preselected products are involved.

2. Check which new products appear within these processes and whether they have external demand or not.

3. If new products have external demand and are outputs of any processes in present consideration, continue with the first step regarding these products.

4. Similarly, for any intermediate of the network (i.e., a product that is produced as well as further processed), continue with the first step.

[10] A product without consumer demand that is not processed is a waste product and, mostly, does not need to be considered. In case of disposal costs, we can treat the product like an external product.

5. Alternatively to steps 3 and 4, products with external demand or intermediates could also be regarded as external products, i.e., they could be modeled with fixed, quantity-independent prices.

6. Stop if requirements a) and b) are satisfied (possibly enforced by step 5).

Algorithm 2.4.1: Setting up a Minimal Consistent Model

Input: Global set of products $\mathcal{P}^0$ of which some have external demand, $\mathcal{P}^0_{out} \subset \mathcal{P}^0$; global set of processes $\mathcal{S}^0$; nonempty set of initial products $\mathcal{P}_{\text{init}} \subset \mathcal{P}^0$, $\mathcal{P}_{\text{init}} \cap \mathcal{P}^0_{out} \neq \emptyset$.

Output: Disjoint product sets $\mathcal{P}_{ex}$, $\mathcal{P}_{out}$, $\mathcal{P}_{mid} \subset \mathcal{P}^0$, where $\mathcal{P}_{out} \neq \emptyset$, $\mathcal{P}_{out} \subseteq \mathcal{P}^0_{out}$, $\mathcal{P}_{mid} \subseteq \mathcal{P}^0_{mid}$, and process set $\mathcal{S} \subseteq \mathcal{S}^0, \mathcal{S} \neq \emptyset$, such that $\mathcal{P}_{ex}$, $\mathcal{P}_{out}$, $\mathcal{P}_{mid}$, and $\mathcal{S}$ build a consistent model.

$\mathcal{P}_{ex} = \mathcal{P}_{out} = \mathcal{P}_{mid} = \mathcal{P}_i = \mathcal{P}_o = \mathcal{S} = \emptyset$
for $p \in \mathcal{P}_{\text{init}}$ **do**
  for $s \in \mathcal{S}^0$ **do**
    if $a^{\text{f}}_{s,p,t} \neq 0$ **then**
      $\mathcal{S} := \mathcal{S} \cup s$
      if $a^{\text{f}}_{s,p,t} < 0$ **then**
        $\mathcal{P}_i := \mathcal{P}_i \cup p$
      else
        $\mathcal{P}_o := \mathcal{P}_o \cup p$
      for $p' \in \mathcal{P}^0 \setminus \mathcal{P}_{\text{init}}$ **do**
        if $a^{\text{f}}_{s,p',t} \neq 0$ **then**
          $\mathcal{P}_{ex} := \mathcal{P}_{ex} \cup p'$
  if $p \in \mathcal{P}^0_{out} \cap \mathcal{P}_o$ **then**
    $\mathcal{P}_{out} := \mathcal{P}_{out} \cup p$
  else if $p \in \mathcal{P}_i \cap \mathcal{P}_o$ **then**
    $\mathcal{P}_{mid} := \mathcal{P}_{mid} \cup p$
  else
    $\mathcal{P}_{ex} := \mathcal{P}_{ex} \cup p$

Algorithm 2.4.1 leads to a minimal model for a predefined set $\mathcal{P}_{\text{init}}$ of initial products, which fulfills the above-mentioned criteria for consistency. Thereby the aim is to enable price simulations for all predefined products that have external demand, i.e., for products of the set $\mathcal{P}_{\text{init}} \cap \mathcal{P}^0_{out}$. Minimality is reached by applying step 5 to all appearing intermediates and products with external demand that are not elements of the predefined product set. Optionally, one could furthermore check if any of the external products is an intermediate with respect to $\mathcal{S}$ and if there is no process in $\mathcal{S}^0 \setminus \mathcal{S}$ that includes it. Such products can be moved from the set $\mathcal{P}_{ex}$ to $\mathcal{P}_{mid}$ without adding further processes.

This would have the advantage that we would not need any price parameter $a^{\pi}_{p_{ex},r,t}$ for them.

Remark 2.4.3. Note that we do not need to know the global sets $\mathcal{P}^0$ and $\mathcal{S}^0$ to set up a consistent model. We must only know all processes related to products that we aim to include in our model as products with external demand or intermediates.

Next, we show how algorithm 2.4.1 can be used in an application to expand a given consistent network model. In order to do this, taking the above requirements a) and b) into account, one should pass through the following steps:

1. Choose one or more external products of the consistent network.
2. Identify all processes in which these products are involved.
3. Check if new products appear within these new processes and, if so, check whether they have external demand or not.
4. If new products have external demand and are outputs of some processes in present consideration, continue with step 2 regarding these products.
5. For any intermediate of the new network, continue with step 2.
6. Alternatively to steps 4 and 5, products with external demand or intermediates can also be regarded as external products, i.e., they can be modeled with fixed, quantity-independent prices.
7. Stop if requirements a) and b) are satisfied (possibly enforced by step 6) and if the network has the requested size.

Algorithm 2.4.2: Extending a Consistent Production Network

Input: Global set of products $\mathcal{P}^0$ of which some have external demand, $\mathcal{P}^0_{out} \subset \mathcal{P}^0$, and global set of processes $\mathcal{S}^0$, disjoint product sets $\mathcal{P}^{\text{orig}}_{ex}$, $\mathcal{P}^{\text{orig}}_{out}$, $\mathcal{P}^{\text{orig}}_{mid} \subset \mathcal{P}^0$, where $\mathcal{P}^{\text{orig}}_{ex} \neq \emptyset$, $\mathcal{P}^{\text{orig}}_{out} \neq \emptyset$, and $\mathcal{P}^{\text{orig}}_{out} \subseteq \mathcal{P}^0_{out}$, as well as set of processes $\mathcal{S}^{\text{orig}} \subset \mathcal{S}^0, \mathcal{S}^{\text{orig}} \neq \emptyset$, such that $\mathcal{P}^{\text{orig}}_{ex}$, $\mathcal{P}^{\text{orig}}_{out}$, $\mathcal{P}^{\text{orig}}_{mid}$, and $\mathcal{S}^{\text{orig}}$ build a consistent model, nonempty set of products $\mathcal{P}_{\text{extend}} \subseteq \mathcal{P}^{\text{orig}}_{ex}$ forming the basis for the model extension.

Output: Disjoint product sets $\mathcal{P}^{\text{new}}_{ex}$, $\mathcal{P}^{\text{new}}_{out}$, $\mathcal{P}^{\text{new}}_{mid} \subset \mathcal{P}^0$ and set of processes $\mathcal{S}^{\text{new}} \subseteq \mathcal{S}^0$ that build a consistent model where $\mathcal{S}^{\text{old}} \subseteq \mathcal{S}^{\text{new}}$.

Apply algorithm 2.4.1 to the set $\mathcal{P}_{\text{init}} := \mathcal{P}_{\text{extend}} \cup \mathcal{P}^{\text{orig}}_{out} \cup \mathcal{P}^{\text{orig}}_{mid}$.
$\Rightarrow$ Disjoint sets $\mathcal{P}_{ex}, \mathcal{P}_{out}$, $\mathcal{P}_{mid}$ and $\mathcal{S}$ that build a consistent model.
$\mathcal{P}^{\text{new}}_{ex} := (\mathcal{P}_{ex} \cup \mathcal{P}^{\text{orig}}_{ex}) \setminus (\mathcal{P}_{out} \cup \mathcal{P}_{mid})$, $\mathcal{P}^{\text{new}}_{out} := \mathcal{P}_{out}$, $\mathcal{P}^{\text{new}}_{mid} := \mathcal{P}_{mid}$, and
$\mathcal{S}^{\text{new}} := \mathcal{S} \cup \mathcal{S}^{\text{orig}}$

Algorithm 2.4.2 creates a consistent network extension given some predefined products $\mathcal{P}_{\text{extend}}$ of the original network, which are preferably elements of the set $\mathcal{P}^0_{out} \cap \mathcal{P}^{\text{orig}}_{ex}$. Similar to algorithm 2.4.1, it leads to a "minimal" extension of the given network in terms of applying step 6 to all intermediates and products with external demand of the new network that are not elements of the predefined product set. To achieve an appropriate model size, algorithm 2.4.2 can be repeated several times with new start products from the recently created network, in particular while choosing $\mathcal{P}_{\text{extend},i+1} \subseteq \mathcal{P}^{\text{new}}_{ex,i} \setminus \mathcal{P}^{\text{new}}_{ex,i-1}$.

Remark 2.4.4. We usually have $\mathcal{S}^{\text{orig}} \backslash \mathcal{S} = \emptyset$. Otherwise, the processes of the set $\mathcal{S}^{\text{orig}} \backslash \mathcal{S}$ include only external products $p \in \mathcal{P}^{\text{orig}}_{ex}$. They are solely added to the new model in the last step of the algorithm to make sure that $\mathcal{S}^{\text{orig}} \subseteq \mathcal{S}^{\text{new}}$ holds. Such processes are, dependent on external product prices, either profitable or not, and independent of the other model variables. Therefore, we get $x^{\text{q}*}_s = 0$ or $x^{\text{q}*}_s = a^{\text{cap}}_s \ \forall \ s \in \mathcal{S}^{\text{orig}} \setminus \mathcal{S}$. This means that the inclusion of these processes to the model only influences the total profit but not the optimal choice of the other variables.

An example for a network extension will be given in the last part of section 6.3.

Remark 2.4.5. (Using presolve methods for an initial model analysis) To check whether the modeled market is reasonable, especially in the sense that all production capacities $a^{\text{cap}}_{i,s,t}$ could be exploited, we propose to make use of presolve methods. They modify a given optimization model by proper transformations and simplifications aiming for a reduced problem size without changing the feasible set of the problem. Before the problem is sent to a solver, for instance arithmetical tests on bounds, equality constraints, and inequality constraints are run to eliminate redundant constraints, to identify infeasible problems, and to tighten variable bounds. In special cases, some variables can even be fixed before the optimization algorithm starts, which gives a remarkable a priori insight into a model setting. [Kal02] includes a short introduction to preprocessing in the context of proper modeling of practical optimization problems. Several techniques for automatic model reformulation and presolve methods are surveyed, e.g., in [AA95]. We formulate our market optimization model in the modeling language AMPL [FGK03] to communicate it to an appropriate solver for constrained nonlinear optimization problems, as SNOPT [Sno08] or IPOPT [Ipo, Wäc02, WB06], and to examine its solutions. AMPL's presolve phase runs automatically when a `solve` command is called up. To track its effects, the option `show_stats` should be changed from its default value 0 to 1. During the presolve phase, AMPL saves two sets of lower and upper bounds on the variables. The bounds that derive from tightening implied by eliminated constraints can be examined using the suffixes `.lb1` and `.ub1`. The suffixes `.lb2` and `.ub2` give the bound values deduced from constraints that presolve could not eliminate. In our application, the latter are of special interest in view of the usability of available production capacities. Many of AMPL's presolve transformations are based on ideas first presented in [BMW75].

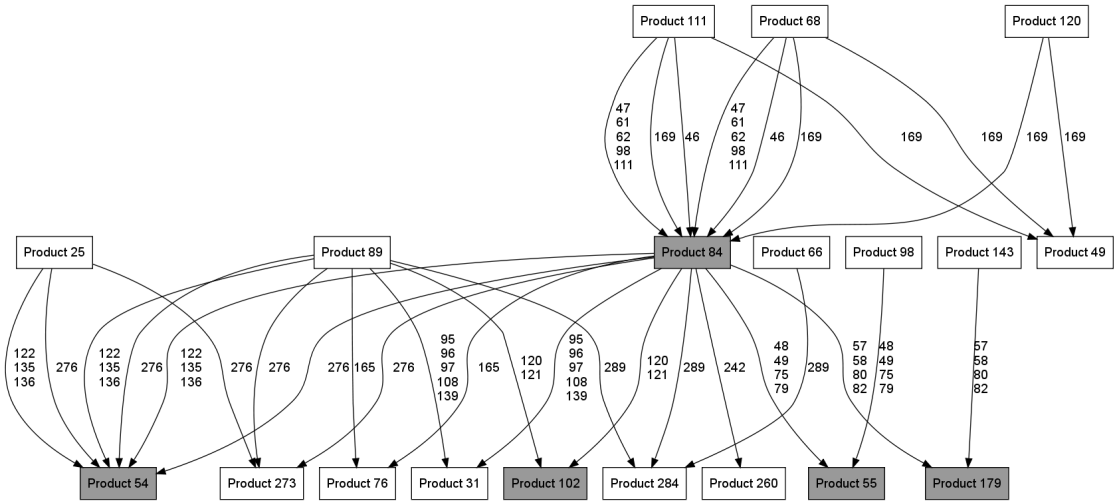

Figure 2.2: *Products and processes of a small petrochemical production system*

2.5 Example: Simulation of Petrochemical Product Prices

In this section, we present a consistent network model for price simulation. We choose a small part of the petrochemical market. The corresponding network of products connected by processes is shown in figure 2.2[11]. We set for each process an arrow from each input to each output product. The arrows are labeled by the reference numbers of the processes. Processes that agree in input and output products, but not necessarily in the related factors, label the same arrows. Products with external demand are gray colored. The remaining products are external. This means, we have $\mathcal{P}_{mid} = \emptyset$ in this example. This model is built by the algorithm 2.4.1 of the previous section, initiated by the products 54, 55, 84, 102, and 179.
Lacking detailed data about some model components, we simplify the models of sections 2.2 and 2.3 by the following points:

- We assume that all consumers have the same demand behavior. By remark 2.1.3, this means that we must consider only one consumer. Therefore, we omit the consumer index c.

- Fixed costs for running the plants are neglected so that we do not need to consider plants, but we can attribute the aggregated capacities of plants that run a certain process to this process, i.e., we define $\forall\ s \in \mathcal{S}:\ a^{\text{cap}}_{s,t} := \sum_{i \in \mathcal{I}} a^{\text{cap}}_{i,s,t}$ and respective decision variables $x^{\text{q}}_{s,t}$.

- There is no storage possibility between time periods.

[11] The network plots in this thesis are created by the open source software Graphviz that provides various common types of graph layouts, and can be used via a C library interface, cf. [Gan11, JM04].

Under the above assumptions, we simulate prices of the five gray highlighted products 54, 55, 84, 102, and 179 for three major regions over the years 2002 to 2009. Exemplary model parameters for one region in one year can be found in section 6.3, tables 6.1 to 6.4. We compare three kinds of simulations that differ mainly in the modeling of changes in the demand-price relationship over the years. The first two kinds of simulations base upon the model for optimizing a single time period, the third is a simultaneous optimization over all periods:

a) **Independent simulations:** We solve problem (2.9) for each year t_i. Thereby, for the modeling of (2.3) we set in (2.2) as $a^{\pi\_\text{last}}_{p_{out},r,t_i}$ and $a^{\text{con\_last}}_{p_{out},r,t_i}$ the historical data of year t_{i-1}.

b) **Dependent sequential simulations:** We solve problem (2.9) for each year t_i, where we set in (2.2) as $a^{\pi\_\text{last}}_{p_{out},r,t_i}$ and $a^{\text{con\_last}}_{p_{out},r,t_i}$ the simulation results of year t_{i-1}.

c) **Simultaneous simulations:** We solve problem (2.17), where $a^{\pi\_\text{last}}_{p_{out},r,t_i}$ and $a^{\text{con\_last}}_{p_{out},r,t_i}$ are defined by (2.16).

Remark 2.5.1. By the first two assumptions of this section and by adding the constraints

$$x^{\pi}_{p_{out},r,t} \leq a^{\pi\_\text{max}}_{p_{out},r,t} + w_{p_{out},r}(a^{GDP}_{r,t}, a^{IndPro}_{r,t}) \qquad \forall\ p_{out} \in \mathcal{P}_{out}, r \in \mathcal{R}, t \in \mathcal{T}, \tag{2.21}$$

the optimization problems related to independent simulations and to dependent sequential simulations turn from non-differentiable, non-convex NLPs to differentiable, convex NLPs, see also remark 2.2.1. This is because they consist of several independent problems, each in one time period. We will discuss the properties of the optimization problem for one time period in some more detail in section 6.1.

For all three kinds of simulations, we choose

$$a^{\pi\_\text{max}}_{p_{out},r,t_j} := 1.4 \cdot a^{\pi\_\text{last}}_{p_{out},r,t_j} \qquad \text{and} \qquad a^{\text{con\_max}}_{p_{out},r,t_j} := 1.5 \cdot a^{\text{con\_last}}_{p_{out},r,t_j}. \tag{2.22}$$

In case of simultaneous optimizations, we set the following further constraints to avoid unrealistic solutions:

$\forall\ p_{out} \in \mathcal{P}_{out}, \forall\ r \in \mathcal{R}, \forall\ t \in \mathcal{T}$

$$0.5 \cdot x^{\text{sales}}_{p_{out},r,t-1} \leq x^{\text{sales}}_{p_{out},r,t} \leq 1.5 \cdot x^{\text{sales}}_{p_{out},r,t-1}. \tag{2.23}$$

In the optimal solution of our example, these constraints are inactive for most products, regions, and time periods.
Figures 2.3 and 2.4 show price and sales results of independent, dependent sequential, and simultaneous market simulations. For comparison purposes, each figure includes also the corresponding historical data of the products 54 and 84, respectively. Note that

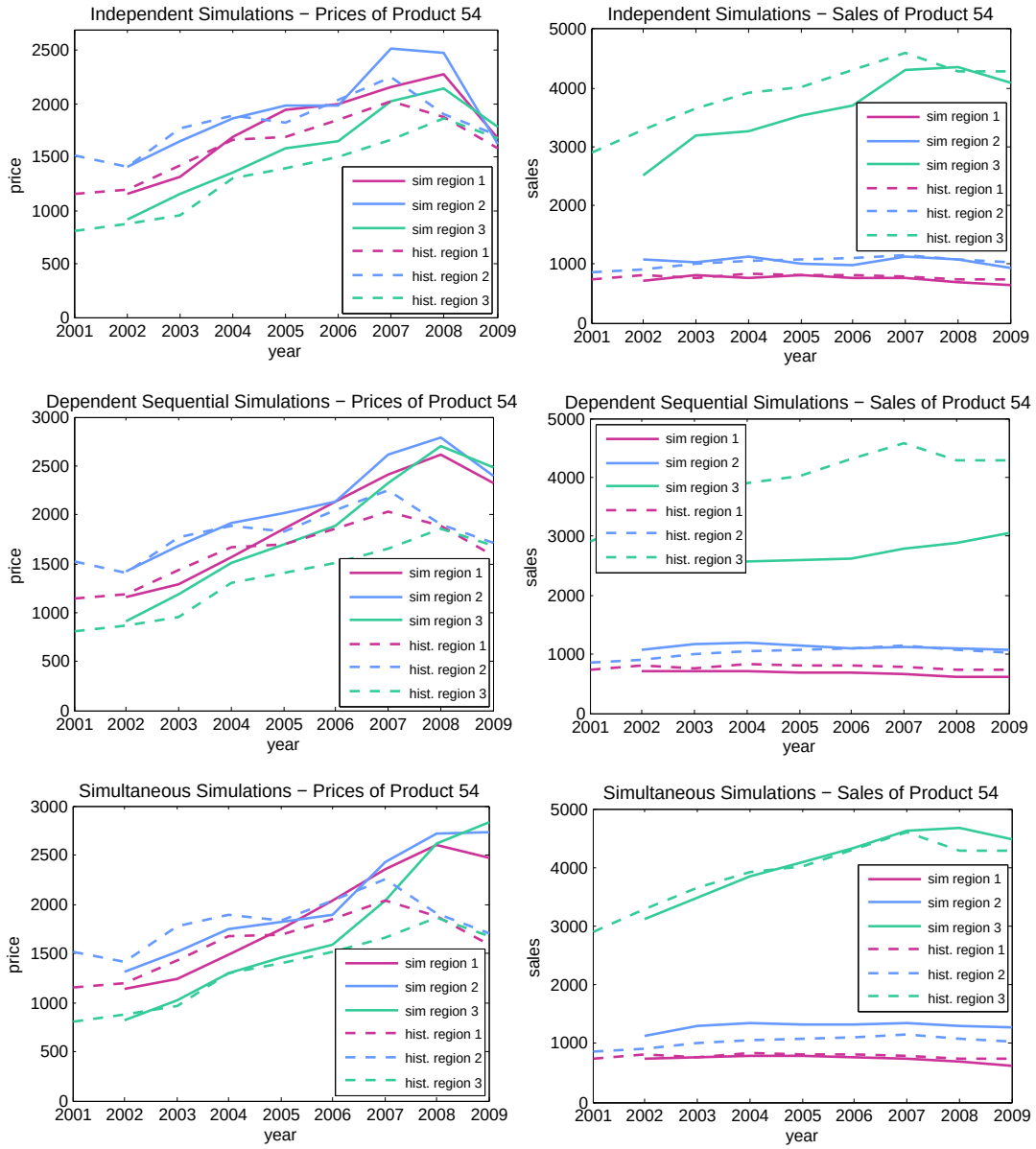

Figure 2.3: *Price and sales of product 54 resulting from three different kinds of market simulations*

the independent and dependent sequential simulations are composed of the solutions of eight market optimizations, one for each year. The simultaneous simulations are part of the solution of one much larger optimization problem.
In general, the results are quite satisfactory and confirm our modeling approach. It seems that the choice of a deterministic optimization model together with the selection of model parameters is appropriate for reproducing price formation. Prices of external

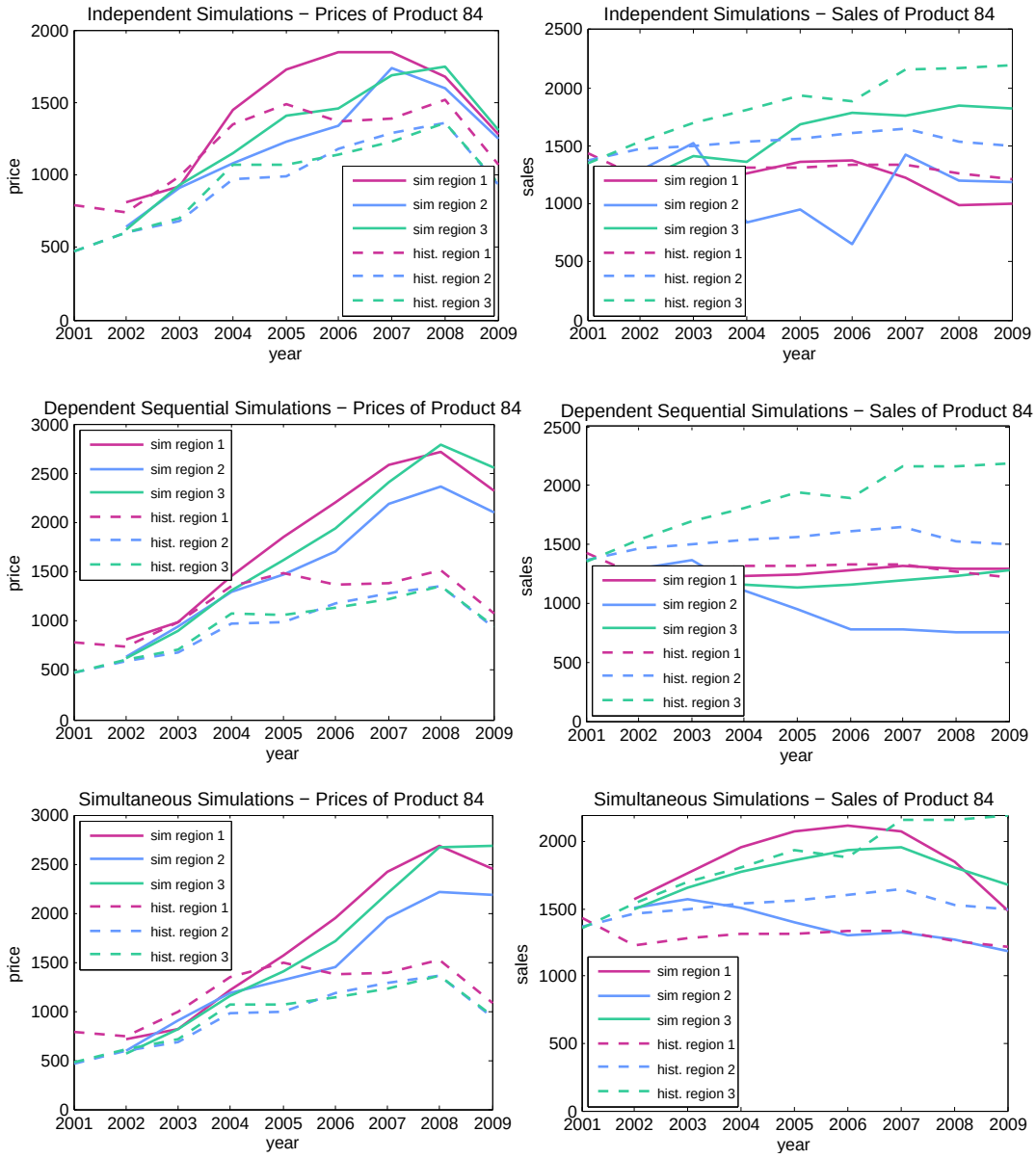

Figure 2.4: *Price and sales of product 84 resulting from three different kinds of market simulations*

products, production and processing facilities, and demand related parameters qualify as main model parameters. Especially the price simulations of the years from 2002 to 2005 represent prices of both products well. However, in the following years simulated prices are too high, and the decrease of prices in 2009 is adequately simulated only in the first case of independent simulations. Dependent sequential and simultaneous simulations show an accumulating too high price increase, particularly for product 84. Regarding

product 54, sales simulations are very good, except for the dependent sequential ones for region 3. Sales simulations of product 84 are less satisfactory: several simulated sales series display considerably more variation than historical data.
Summarizing, the simulation results give reason to further apply and analyze our model. We expect that enhancing our modeling by a better demand model and further data to estimate demand parameters will improve the simulation results. In the present application, we just set reasonable values for some demand parameters, cf. (2.22). Furthermore, the weighting parameters $\alpha_{p_{out},r}$ and $\beta_{p_{out},r}$ concerning the economic factors, cf. (2.4), are estimated by an ordinary least squares method. A generalized least squares method to identify demand parameters could improve the simulation results. An extensive discussion of demand modeling approaches and the related parameter identification methods is given by [Kel13]. A further reason for the deviations of the simulations from the historical time series could be the incomplete supply-side data.

3 Constrained Nonlinear Optimization: A Choice of Theory and Methods

This chapter outlines important theoretical results on optimization theory, in particular on parametric optimization (section 3.2), duality (section 3.3), and nonsmooth optimization (section 3.4). We mainly restrict ourselves to those results that we will need for introducing a new model reduction approach in chapter 5 and for its application in chapter 6. Since our approach is strongly related to the decomposition methods for solving constrained programs with a block-separable structure, we review primal and dual decomposition in section 3.5. Furthermore, in section 3.6, we present a successful interior point method, which serves in this thesis as standard approach to solve constrained nonlinear optimization problems as that of chapter 6.1. Section 3.7 shortly introduces the method of least squares for parameter estimation.

3.1 Optimization Theory for Differentiable Problems

This section gives an overview on major results about differentiable constrained nonlinear optimization. Detailed introductions as well as further references can be found, e.g., in [Fle87, Alt02, Lue08].

A general finite-dimensional optimization problem in $\mathbb{R}^n$ is given by the following definition:

Definition 3.1.1. *Let $D \subseteq \mathbb{R}^n$ be open, $f : D \to \mathbb{R}$ and $\mathcal{F} \subseteq D$. An **optimization problem** is the problem of minimizing the function f over $\mathcal{F}$, i.e.,*

$$\min_{\boldsymbol{x} \in \mathcal{F}} f(\boldsymbol{x}). \tag{3.1}$$

*The function f is called **objective function**, $\mathcal{F}$ is the **feasible set** and each $\boldsymbol{x} \in \mathcal{F}$ is called a **feasible point**.*
*If $\mathcal{F} = D$, the problem is **unconstrained**. Is $\mathcal{F}$ defined by certain constraints, the problem is called a **constrained** optimization problem.*

The above optimization problem is specified by the following definitions:

Definition 3.1.2. *a) $\boldsymbol{x}^* \in \mathcal{F}$ is a **local minimum** of f over $\mathcal{F}$ if there exists $r > 0$ with*

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}^*) \ \forall \ \boldsymbol{x} \in \mathcal{F} \cap B(\boldsymbol{x}^*, r). \tag{3.2}$$

b) $\boldsymbol{x}^* \in \mathcal{F}$ *is a* ***strict local minimum*** *of* f *over* $\mathcal{F}$ *if there exists* $r > 0$ *with*

$$f(\boldsymbol{x}) > f(\boldsymbol{x}^*) \ \forall \ \boldsymbol{x} \in \mathcal{F} \cap B(\boldsymbol{x}^*, r), \boldsymbol{x} \neq \boldsymbol{x}^*. \tag{3.3}$$

c) $\boldsymbol{x}^* \in \mathcal{F}$ *is a* ***global minimum*** *of* f *over* $\mathcal{F}$ *if*

$$f(\boldsymbol{x}) \geq f(\boldsymbol{x}^*) \ \forall \ \boldsymbol{x} \in \mathcal{F}. \tag{3.4}$$

d) $\boldsymbol{x}^* \in \mathcal{F}$ *is a* ***strict global minimum*** *of* f *over* $\mathcal{F}$ *if*

$$f(\boldsymbol{x}) > f(\boldsymbol{x}^*) \ \forall \ \boldsymbol{x} \in \mathcal{F}, \boldsymbol{x} \neq \boldsymbol{x}^*. \tag{3.5}$$

e) $\boldsymbol{x}^* \in \mathcal{F}$ *is an* ***isolated local minimum*** *of* f *over* $\mathcal{F}$ *if there exists* $r > 0$ *such that* $\mathcal{F} \cap B(\boldsymbol{x}^*, r)$ *includes no other local minimum* $\bar{\boldsymbol{x}} \neq \boldsymbol{x}^*$.

f) If $\boldsymbol{x}^* \in \mathcal{F}$ *is a solution of* (3.1)*, we call* $f(\boldsymbol{x}^*)$ ***optimal value****.*

Remark 3.1.3. Each isolated local minimum is a strict local minimum.

Remark 3.1.4. The problem of maximizing f over $\mathcal{F}$ is equivalent to the problem of minimizing $(-f)$ over $\mathcal{F}$, which defines *(strict/isolated) local/global maxima* of f over $\mathcal{F}$.

The number of (strict) local/global solutions of an optimization problem depends on the form of f and $\mathcal{F}$. A special class of nonlinear problems with certain convenient properties is that of convex problems.

Definition 3.1.5. *The minimization problem* (3.1) *is called* ***convex*** *if* $\mathcal{F}$ *is a convex set and if the objective function* $f : D \to \mathbb{R}$ *is convex over* $\mathcal{F}$*.*

Theorem 3.1.6. *Let* $D \subseteq \mathbb{R}^n$*,* $\mathcal{F} \subseteq D$ *nonempty and convex and* $f : D \to \mathbb{R}$ *convex over* $\mathcal{F}$*. Then, every local minimum of* f *over* $\mathcal{F}$ *is global, and the set of minima of* (3.1)

$$\mathcal{S} = \{ \ \boldsymbol{x} \in \mathcal{F} \mid f(\boldsymbol{x}) \leq f(\boldsymbol{y}) \ \forall \ \boldsymbol{y} \in \mathcal{F} \ \} \tag{3.6}$$

is convex.
If f *is even strictly convex over* $\mathcal{F}$ *and there exists a solution* $\boldsymbol{x}^*$ *of* (3.1) *then* $\boldsymbol{x}^*$ *is unique and a strict global minimum of* f *over* $\mathcal{F}$*.*

Proof Proofs can be found, e.g., in [Fle87, Alt02].

From now on, we focus on constrained problems where $\mathcal{F}$ is of the form

$$\mathcal{F} = \{ \ \boldsymbol{x} \in \mathbb{R}^n \mid g_i(\boldsymbol{x}) \leq 0 \ (i \in I = \{1, ..., m\}), \ h_j(\boldsymbol{x}) = 0 \ (j \in J = \{1, ..., p\}) \ \} \tag{3.7}$$

with $g_i : D \to \mathbb{R}$ $(i \in I)$, $h_j : D \to \mathbb{R}$ $(j \in J)$ and $p \leq n$.

Example 3.1.7. Problem (3.1) is convex if and only if f is convex and if $\mathcal{F}$ can be written as (3.7) with g_i convex for $i \in I$ and $h_j := (\boldsymbol{a}_j)^{\mathrm{T}}\boldsymbol{x} - b_j$ ($\boldsymbol{a}_j \in \mathbb{R}^n, b_j \in \mathbb{R}$) for $j \in J$. In the following, we refer to the matrix of rows $(\boldsymbol{a}_j)^{\mathrm{T}}$ as $\boldsymbol{A}$ so that we can express the equality constraints of convex problems by $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} = \boldsymbol{0}$.

To formulate necessary and sufficient conditions that a point $\boldsymbol{x}^* \in \mathcal{F}$ is a local solution of (3.1), further definitions are needed:

Definition 3.1.8. *An inequality constraint g_i is **active** in $\bar{\boldsymbol{x}} \in \mathcal{F}$ if $g_i(\bar{\boldsymbol{x}}) = 0$ and **inactive** otherwise. For $\bar{\boldsymbol{x}} \in \mathcal{F}$, the set*

$$I(\bar{\boldsymbol{x}}) := \{\ i \in I \mid g_i(\bar{\boldsymbol{x}}) = 0\ \} \tag{3.8}$$

*is called the **active set**.*

Definition 3.1.9. *A point $\boldsymbol{x}^* \in \mathcal{F}$ is **regular** if the gradient vectors $\nabla g_i(\boldsymbol{x}^*), \nabla h_j(\boldsymbol{x}^*), i \in I(\boldsymbol{x}^*), j \in J$ are linearly independent. In this case, one also says that the **Linear Independence Constraint Qualification (LICQ)** holds.*

a. First Order Optimality Conditions

A first order necessary condition for solutions of (3.1) is given by

Theorem 3.1.10 (**Karush-Kuhn-Tucker (KKT) Conditions**). *Let $\boldsymbol{x}^*$ be a local minimizer of (3.1), $\boldsymbol{x}^*$ regular. Then there exist unique **Lagrange multipliers** $\boldsymbol{\lambda} \in \mathbb{R}^m, \boldsymbol{\lambda} \geq \boldsymbol{0}, \boldsymbol{\mu} \in \mathbb{R}^p$ such that the **Lagrangian function***

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) := f(\boldsymbol{x}) + \boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{h}(\boldsymbol{x}) \tag{3.9}$$

satisfies

$$\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \nabla f(\boldsymbol{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(\boldsymbol{x}^*) + \sum_{j=1}^{p} \mu_j \nabla h_j(\boldsymbol{x}^*) = \boldsymbol{0} \tag{3.10}$$

*and **complementarity***

$$\boldsymbol{\lambda}^{\mathrm{T}}\boldsymbol{g}(\boldsymbol{x}^*) = \sum_{i=1}^{m} \lambda_i g_i(\boldsymbol{x}^*) = 0 \tag{3.11}$$

*holds. A point $(\boldsymbol{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathcal{F} \times \mathbb{R}^m \times \mathbb{R}^p$ that satisfies $\boldsymbol{\lambda} \geq \boldsymbol{0}$, (3.10), and (3.11) is called a **KKT-point**.*

Proof Proofs are given in any book that covers constrained nonlinear optimization, e.g., in [Fle87, Lue08].

Equation (3.11) is equivalent to

$$\begin{aligned} g_i(\boldsymbol{x}^*) < 0 \quad &\Rightarrow \quad \lambda_i = 0 \\ \text{resp.} \quad \lambda_i > 0 \quad &\Rightarrow \quad g_i(\boldsymbol{x}^*) = 0. \end{aligned} \tag{3.12}$$

This shows that the uniqueness of the Lagrange multipliers related to the inactive constraints is directly given by complementarity. The uniqueness of the remaining multipliers is then given by the LICQ and (3.10). Under weaker constraint qualifications than LICQ as the Abadie-CQ and the Mangasarian-Fromovitz-CQ, Langrange multipliers exist but they are not necessary unique.
For cases where also the reverse directions of (3.12) hold, one defines:

Definition 3.1.11. *A KKT-point* $(\boldsymbol{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu})$ *fulfills* ***strict complementarity*** *if* $\forall\, i \in I$

$$\begin{aligned} g_i(\boldsymbol{x}^*) = 0 \quad &\Rightarrow \quad \lambda_i > 0 \\ \textit{resp.} \quad \lambda_i = 0 \quad &\Rightarrow \quad g_i(\boldsymbol{x}^*) < 0. \end{aligned} \tag{3.13}$$

An active constraint g_i *that satisfies the first condition of* (3.13) *is called* ***strictly active*** *in* $\boldsymbol{x}^*$*. The* ***set of strictly active constraints*** *in* $\boldsymbol{x}^*$ *is defined as*

$$I^+(\boldsymbol{x}^*) := \{\ i \in I(\boldsymbol{x}^*) \mid \lambda_i > 0\ \}. \tag{3.14}$$

For convex optimization problems that meet Slater's condition, the KKT conditions are not only necessary but also sufficient for a global optimum:

Definition 3.1.12. *A minimization problem* (3.1) *with feasible set* (3.7) *fulfills* ***Slater's condition*** *if there exists a feasible point* $\boldsymbol{x} \in \mathcal{F}$ *for which none of the inequality constraints is active:*

$$\exists\, \boldsymbol{x} \in \mathbb{R}^n \ :\ g_i(\boldsymbol{x}) < 0\ (i \in I = \{1, ..., m\}),\ h_j(\boldsymbol{x}) = 0\ (j \in J = \{1, ..., p\}). \tag{3.15}$$

Theorem 3.1.13 (KKT Conditions for Convex Problems). *Let a convex minimization problem* (3.1) *fulfill Slater's condition. Then the following statements about a feasible point* $\boldsymbol{x}^* \in \mathcal{F}$ *are equivalent*

a) $\boldsymbol{x}^*$ *is a global minimum of* (3.1)*,*

b) $\exists\, \boldsymbol{\lambda} \geq \boldsymbol{0}, \boldsymbol{\mu}$ *such that* $(\boldsymbol{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu})$ *is a KKT-point of* (3.1)*.*

Proof As mentioned in example 3.1.7, a convex problem can be written by a convex objective function f, convex inequality constraint functions g_i, and affine equality constraint functions h_j.
"*a)* $\Rightarrow$ *b)*" follows from strong duality, i.e., zero duality gap, for convex problems that

fulfill Slater's condition, cf. theorem 3.3.4: Let $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ solve the Lagrange dual problem (3.40) associated with problem (3.1). Since $\boldsymbol{x}^*$ minimizes $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$, the gradient $\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ must vanish in $\boldsymbol{x}^*$:

$$\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \nabla f(\boldsymbol{x}^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g_i(\boldsymbol{x}^*) + \sum_{j=1}^{p} \mu_j^* \boldsymbol{a}_j = \boldsymbol{0}. \tag{3.16}$$

Complementarity, i.e., $\boldsymbol{\lambda}^{\mathrm{T}} g(\boldsymbol{x}^*) = 0$, results from the zero duality gap ($f(\boldsymbol{x}^*) = d(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) = f(\boldsymbol{x}^*) + \boldsymbol{\lambda}^{*\mathrm{T}} \boldsymbol{g}(\boldsymbol{x}^*) + \boldsymbol{\mu}^{*\mathrm{T}}(\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b})$).
"*b) ⇒ a)*": For a feasible point $\boldsymbol{x} \in \mathcal{F}$, we have under convexity and the KKT conditions:

$$\begin{aligned}
f(\boldsymbol{x}) &\geq f(\boldsymbol{x}^*) + \nabla f(\boldsymbol{x}^*)^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}^*) \\
&= f(\boldsymbol{x}^*) - \sum_{i=1}^{m} \lambda_i \nabla g_i(\boldsymbol{x}^*)^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}^*) - \sum_{j=1}^{p} \mu_j \boldsymbol{a}_j^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}^*) \\
&= f(\boldsymbol{x}^*) - \sum_{i \in I(\boldsymbol{x}^*)} \lambda_i \nabla g_i(\boldsymbol{x}^*)^{\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}^*) \\
&\geq f(\boldsymbol{x}^*) - \sum_{i \in I(\boldsymbol{x}^*)} \lambda_i (g_i(\boldsymbol{x}) - g_i(\boldsymbol{x}^*)) \\
&\geq f(\boldsymbol{x}^*).
\end{aligned} \tag{3.17}$$

b. Second Order Optimality Conditions

With the above definitions it is now possible to state second order necessary and sufficient conditions for a locally optimal point of problem (3.1):

Theorem 3.1.14 (**Second Order Necessary Condition**). *Let $\boldsymbol{x}^*$ be a local minimizer of (3.1), $\boldsymbol{x}^*$ regular. Then the Hessian $\nabla^2_{\boldsymbol{xx}}\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu})$ of the Lagrangian is positive semidefinite on the tangent space of the equality and active inequality constraints, i.e.,*

$$\boldsymbol{p}^{\mathrm{T}} \nabla^2_{\boldsymbol{xx}}\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu})\boldsymbol{p} \geq 0 \quad \forall\ \boldsymbol{p} \in T(\boldsymbol{x}^*), \tag{3.18}$$

where $T(\boldsymbol{x}^*) = \{\ \boldsymbol{p} \in \mathbb{R}^n \mid \nabla g_i(\boldsymbol{x}^*)^{\mathrm{T}}\boldsymbol{p} = 0, \nabla h_j(\boldsymbol{x}^*)^{\mathrm{T}}\boldsymbol{p} = 0, i \in I(\boldsymbol{x}^*), j \in J\ \}$.

Proof A proof can be found, e.g., in [Lue08].

To obtain a sufficient condition for a local minimizer, in case that there are inequality constraints that are not strictly active, $\nabla^2_{\boldsymbol{xx}}\mathcal{L}$ must be positive definite on a subspace that is larger than the tangent space $T(\boldsymbol{x}^*)$:

Theorem 3.1.15 (**Strong Second Order Sufficient Condition**). *Let $(\boldsymbol{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu})$ be a KKT-point for problem (3.1). Let the Hessian $\nabla^2_{\boldsymbol{xx}}\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu})$ of the Lagrangian be positive definite on the tangent space of the equality and strictly active inequality constraints, i.e.,*

$$\boldsymbol{p}^{\mathrm{T}} \nabla^2_{\boldsymbol{xx}}\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu})\boldsymbol{p} > 0 \quad \forall\ \boldsymbol{p} \in T^+(\boldsymbol{x}^*) \setminus \{0\}, \tag{3.19}$$

where $T^+(\boldsymbol{x}^*) = \{\ \boldsymbol{p} \in \mathbb{R}^n \mid \nabla g_i(\boldsymbol{x}^*)^{\mathrm{T}}\boldsymbol{p} = 0, \nabla h_j(\boldsymbol{x}^*)^{\mathrm{T}}\boldsymbol{p} = 0, i \in I^+(x^*), j \in J\ \}$. *Then* $\boldsymbol{x}^*$ *is a strict local minimizer for problem* (3.1).

Proof A proof can be found, e.g., in [Lue08].

3.2 Parametric Optimization Problems

This section gives a short summary of important sensitivity and stability results for constrained nonlinear optimization problems. Main interests are, e.g., in continuity and differentiability of optimal value functions as well as of local optimal solutions. An introduction to sensitivity and stability analysis for nonlinear programming is given in [Fia83]. [FI90] gives a brief overview. Detailed representations and advanced results can be found in [BS00] and [GG97], respectively.

We extend the constrained nonlinear problem (3.1) by a parameter $\boldsymbol{\theta} \in \mathbb{R}^l$:

$$\min_{\boldsymbol{x} \in \mathcal{F}(\boldsymbol{\theta})} f(\boldsymbol{x}, \boldsymbol{\theta}) \tag{3.20a}$$

with

$$\mathcal{F}(\boldsymbol{\theta}) = \{\ \boldsymbol{x} \in \mathbb{R}^n \mid g_i(\boldsymbol{x}, \boldsymbol{\theta}) \leq 0\ (i \in I),\ h_j(\boldsymbol{x}, \boldsymbol{\theta}) = 0\ (j \in J)\ \} \tag{3.20b}$$

where $f : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}$, $g_i : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}$ $(i \in I)$ and $h_j : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}$ $(j \in J)$.

Definition 3.2.1. *The* ***optimal value function*** f^* *of a parametric optimization problem* (3.20) *is defined as*

$$f^*(\boldsymbol{\theta}) := \begin{cases} \inf\limits_{\boldsymbol{x} \in \mathcal{F}(\boldsymbol{\theta})} f(\boldsymbol{x}, \boldsymbol{\theta}) & \text{if } \mathcal{F}(\boldsymbol{\theta}) \neq \emptyset, \\ +\infty & \text{if } \mathcal{F}(\boldsymbol{\theta}) = \emptyset. \end{cases} \tag{3.21}$$

The corresponding ***optimal solution map*** S *is given by*

$$S(\boldsymbol{\theta}) = \{\ \boldsymbol{x} \in \mathcal{F}(\boldsymbol{\theta}) \mid f(\boldsymbol{x}, \boldsymbol{\theta}) = f^*(\boldsymbol{\theta})\ \}, \tag{3.22}$$

i.e., it is a point-to-set mapping from $\mathbb{R}^l$ *to* $\mathcal{F}$.

To state the basic result on continuity of the optimal-value function, we define continuity for a point-to-set map according to Berge [Ber63]:

Definition 3.2.2. *Let* (X, d_X) *and* (Θ, d_Θ) *be metric spaces. A point-to-set mapping* $\Gamma : \Theta \to 2^X$ *(where* 2^X *is the power set of* X*) is called*

a) ***upper semicontinuous*** *at a point* θ_0 *if for each open set* $\Omega \subseteq X$ *containing* $\Gamma\theta_0$ *there exists a number* $\delta = \delta(\Omega) > 0$ *such that* $\Gamma\theta \subseteq \Omega\ \forall\ \theta \in U_\delta(\theta_0)$*;*

b) ***lower semicontinuous*** *at a point* θ_0 *if for each open set* $\Omega \subseteq X$ *satisfying* $\Omega \cap \Gamma\theta_0 \neq \emptyset$ *there exists a number* $\delta = \delta(\Omega) > 0$ *such that* $\Gamma\theta \cap \Omega \neq \emptyset \ \forall\ \theta \in U_\delta(\theta_0)$*;*

c) ***continuous*** *at a point* θ_0 *if it is upper semicontinuous and lower semicontinuous at* θ_0*.*

Remark 3.2.3. This definition of continuity for a point-to-set map differs from the one for a real-valued function. However, we use the same term in the following since it will be clear which definition holds.

Theorem 3.2.4. *For problem* (3.20)*, one has*

a) *if* $\mathcal{F}$ *is lower semicontinuous at* $\bar{\boldsymbol{\theta}}$*, and* f *is upper semicontinuous on* $\mathcal{F}(\bar{\boldsymbol{\theta}}) \times \{\bar{\boldsymbol{\theta}}\}$*, then* f^* *is upper semicontinuous at* $\bar{\boldsymbol{\theta}}$*;*

b) *if* $\mathcal{F}$ *is upper semicontinuous at* $\bar{\boldsymbol{\theta}}$*,* $\mathcal{F}(\bar{\boldsymbol{\theta}})$ *compact, and* f *is lower semicontinuous on* $\mathcal{F}(\bar{\boldsymbol{\theta}}) \times \{\bar{\boldsymbol{\theta}}\}$*, then* f^* *is lower semicontinuous at* $\bar{\boldsymbol{\theta}}$*.*

Proof A proof can be found, e.g., in [BGK$^+$82].

This result can be furthermore specified for the right-hand-side constraint map

$$\mathcal{F}(\boldsymbol{\theta}) = \{\ \boldsymbol{x} \in \mathbb{R}^n \mid g_i(\boldsymbol{x}) \leq \theta_i \ (i = 1, ..., l)\ \} \tag{3.23}$$

where $\boldsymbol{g} : \mathbb{R}^n \to \mathbb{R}^l$ and $\boldsymbol{\theta} \in \mathbb{R}^l$.

Theorem 3.2.5. *Let* $\mathcal{F}$ *be given by* (3.23)*, where* $\boldsymbol{g}$ *is continuous, and let* $\mathcal{F}(\bar{\boldsymbol{\theta}})$ *be compact. Then*

a) $\mathcal{F}$ *is upper semicontinuous at* $\bar{\boldsymbol{\theta}}$ *if and only if there exists a vector* $\tilde{\boldsymbol{\theta}} > \bar{\boldsymbol{\theta}}$ *such that* $\mathcal{F}(\tilde{\boldsymbol{\theta}})$ *is compact;*

b) *if the 'strict interior'* $\mathcal{F}^0(\bar{\boldsymbol{\theta}}) = \{\ \boldsymbol{x} \in \mathbb{R}^n \mid g_i(\boldsymbol{x}) < \bar{\theta}_i \ (i = 1, ..., l)\ \}$ *is nonempty, then* $\mathcal{F}$ *is lower semicontinuous at* $\bar{\boldsymbol{\theta}}$ *if and only if* $\overline{\mathcal{F}^0(\bar{\boldsymbol{\theta}})} = \mathcal{F}(\bar{\boldsymbol{\theta}})$*, where* $\overline{\mathcal{F}^0(\bar{\boldsymbol{\theta}})}$ *is the closure of* $\mathcal{F}^0(\bar{\boldsymbol{\theta}})$*.*

Proof These results are proved by [EG70].

We will now state conditions for the convexity (concavity) of f^*. For this purpose, $\Theta \subseteq \mathbb{R}^l$ is assumed to be a nonempty and convex set.

Definition 3.2.6. *A point-to-set map* $\mathcal{F} : \Theta \to 2^{\mathbb{R}^n}$ *is* ***convex (concave)*** *on* Θ *if* $\forall\ \boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \Theta$ *and* $\forall\ c \in (0, 1)$

$$c\mathcal{F}(\boldsymbol{\theta}_1) + (1 - c)\mathcal{F}(\boldsymbol{\theta}_2) \subseteq (\supseteq)\ \mathcal{F}(c\boldsymbol{\theta}_1 + (1 - c)\boldsymbol{\theta}_2). \tag{3.24}$$

If the inclusion $\subseteq$ *holds* $\ \forall\ \boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \Theta, \boldsymbol{\theta}_1 \neq \boldsymbol{\theta}_2$ *and* $\forall\ c \in (0, 1)$*,* $\mathcal{F}$ *is called* ***essentially convex*** *on* Θ*.*

Theorem 3.2.7. *For problem* (3.20)*, let f be jointly convex*[1] *on $\{(\boldsymbol{x}, \boldsymbol{\theta}) | \boldsymbol{x} \in \mathcal{F}(\boldsymbol{\theta}), \boldsymbol{\theta} \in \Theta\}$ and let $\mathcal{F}$ be essentially convex on Θ, where Θ is convex. Then f^* is convex on Θ.*

Proof A proof is given by [FK86].

Theorem 3.2.8. *Let $\mathcal{F}(\boldsymbol{\theta})$ be defined by* (3.20b) *with g_i $(i \in I)$ jointly quasiconvex on $\mathbb{R}^n \times \Theta$ and h_j $(j \in J)$ jointly affine on $\mathbb{R}^n \times \Theta$. Then $\mathcal{F}$ is convex on Θ.*

Proof By quasiconvexity of g_i $(i \in I)$ and affinity of h_j $(j \in J)$, the graph G of $\mathcal{F}$ defined by $G(\mathcal{F}) = \{(\boldsymbol{x}, \boldsymbol{\theta}) \in \mathbb{R}^n \times \Theta | \boldsymbol{x} \in \mathcal{F}(\boldsymbol{\theta})\}$ is a convex set. This implies the convexity of $\mathcal{F}$ on Θ.

Theorem 3.2.9. *For problem* (3.20)*, let f be jointly concave on $\mathbb{R}^n \times \Theta$ and let $\mathcal{F}$ be concave on Θ, where Θ is convex. Then f^* is concave on Θ.*

Proof A proof is given by [FK86].

The following theorem states conditions under which the optimal value function is locally differentiable.

Theorem 3.2.10. *For problem* (3.20) *let $f, \boldsymbol{g}$, and $\boldsymbol{h}$ be two times continuously differentiable on $\mathbb{R}^n \times \mathbb{R}^l$. Suppose that for $\boldsymbol{\theta} = \mathbf{0}$ we have a KKT-point $\boldsymbol{x}(\mathbf{0})$ that satisfies strict complementarity and, together with the Lagrange multipliers $\boldsymbol{\lambda}(\mathbf{0}) \geq \mathbf{0}, \boldsymbol{\mu}(\mathbf{0})$, corresponding to inequality and equality constraints, the second-order sufficiency conditions for a strict local minimum.*
Then

a) $\boldsymbol{x}(\mathbf{0})$ is an isolated local minimum of problem (3.20) *for $\boldsymbol{\theta} = \mathbf{0}$ with unique Lagrange multipliers $\boldsymbol{\lambda}(\mathbf{0}), \boldsymbol{\mu}(\mathbf{0})$;*

b) $\exists\ r > 0$ and a once continuously differentiable vector function $\boldsymbol{z} : B(\mathbf{0}, r) \to \mathbb{R}^{n+m+p}, \boldsymbol{\theta} \mapsto \boldsymbol{z}(\boldsymbol{\theta}) = [\boldsymbol{x}(\boldsymbol{\theta}), \boldsymbol{\lambda}(\boldsymbol{\theta}), \boldsymbol{\mu}(\boldsymbol{\theta})]^{\mathrm{T}}$ such that $\forall\ \boldsymbol{\theta} \in B(\mathbf{0}, r)$, $\boldsymbol{x}(\boldsymbol{\theta})$ is an isolated local minimum of the corresponding problem (3.20) *with associated unique Lagrange multipliers $\boldsymbol{\lambda}(\boldsymbol{\theta}), \boldsymbol{\mu}(\boldsymbol{\theta})$;*

c) the LICQ and strict complementarity hold at $[\boldsymbol{x}(\boldsymbol{\theta}), \boldsymbol{\lambda}(\boldsymbol{\theta}), \boldsymbol{\mu}(\boldsymbol{\theta})]$;

d) moreover, the optimal value function $f^*(\boldsymbol{\theta})$ is twice continuously differentiable on $B(\mathbf{0}, r)$.

Proof The proof is based on the implicit function theorem and can be found, e.g., in [Fia83].

[1] A function $f : \mathbb{R}^n \times \Theta \to \mathbb{R}$ is called **jointly convex** if $\forall\ \boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathbb{R}^n, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \Theta$ and $\forall\ c \in [0, 1]$, we have $cf(\boldsymbol{x}_1, \boldsymbol{\theta}_1) + (1 - c)f(\boldsymbol{x}_2, \boldsymbol{\theta}_2) \geq f(c\boldsymbol{x}_1 + (1 - c)\boldsymbol{x}_2, c\boldsymbol{\theta}_1 + (1 - c)\boldsymbol{\theta}_2)$.

In the context of parametric optimization, the Lagrangian function $\mathcal{L}$ is defined as

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\theta}) := f(\boldsymbol{x}, \boldsymbol{\theta}) + \boldsymbol{\lambda}^{\mathrm{T}} \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{\theta}) + \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{\theta}). \tag{3.25}$$

The last item of theorem 3.2.10 can be furthermore specified by

Theorem 3.2.11. *Defining the* ***optimal value Lagrangian*** $\mathcal{L}^*(\boldsymbol{\theta})$ *by*

$$\mathcal{L}^*(\boldsymbol{\theta}) := \mathcal{L}[\boldsymbol{x}(\boldsymbol{\theta}), \boldsymbol{\lambda}(\boldsymbol{\theta}), \boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{\theta}], \tag{3.26}$$

we have, under the assumptions of theorem 3.2.10, in a neighborhood $B(\mathbf{0}, r)$ *of* $\boldsymbol{\theta} = \mathbf{0}$

a) $f^*(\boldsymbol{\theta}) = \mathcal{L}^*(\boldsymbol{\theta})$*;*

b)
$$\begin{aligned} \nabla_{\boldsymbol{\theta}} f^*(\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \mathcal{L} = \nabla_{\boldsymbol{\theta}} f + \sum_{i=1}^{m} \lambda_i(\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} g_i + \sum_{j=1}^{p} \mu_j(\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} h_j \\ &= \nabla_{\boldsymbol{\theta}} f + \boldsymbol{\lambda}(\boldsymbol{\theta})^{\mathrm{T}} \nabla_{\boldsymbol{\theta}} \boldsymbol{g} + \boldsymbol{\mu}(\boldsymbol{\theta})^{\mathrm{T}} \nabla_{\boldsymbol{\theta}} \boldsymbol{h}; \end{aligned}$$

c)
$$\begin{aligned} \nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} f^*(\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \big[\nabla_{\boldsymbol{\theta}} \mathcal{L}[\boldsymbol{x}(\boldsymbol{\theta}), \boldsymbol{\lambda}(\boldsymbol{\theta}), \boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{\theta}]^{\mathrm{T}} \big] \\ &= \nabla^2_{\boldsymbol{x}\boldsymbol{\theta}} \mathcal{L} \nabla_{\boldsymbol{\theta}} \boldsymbol{x}(\boldsymbol{\theta}) + \sum_{i=1}^{m} \nabla_{\boldsymbol{\theta}} g_i^{\mathrm{T}} \nabla_{\boldsymbol{\theta}} \lambda_i(\boldsymbol{\theta}) + \sum_{j=1}^{p} \nabla_{\boldsymbol{\theta}} h_j^{\mathrm{T}} \nabla_{\boldsymbol{\theta}} \mu_j(\boldsymbol{\theta}) + \nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} \mathcal{L}. \end{aligned}$$

d) Furthermore,

$$\begin{pmatrix} \nabla_{\boldsymbol{\theta}} \boldsymbol{x}(\boldsymbol{\theta}) \\ \nabla_{\boldsymbol{\theta}} \boldsymbol{\lambda}(\boldsymbol{\theta}) \\ \nabla_{\boldsymbol{\theta}} \boldsymbol{\mu}(\boldsymbol{\theta}) \end{pmatrix} = -\boldsymbol{M}^{-1} \boldsymbol{N}, \tag{3.27}$$

where

$$\boldsymbol{M} \equiv \left(\begin{array}{c|ccc|ccc} \nabla^2_{\boldsymbol{x}\boldsymbol{x}} \mathcal{L} & \nabla_{\boldsymbol{x}} g_1 & \cdots & \nabla_{\boldsymbol{x}} g_m & \nabla_{\boldsymbol{x}} h_1 & \cdots & \nabla_{\boldsymbol{x}} h_p \\ \hline \lambda_1(\boldsymbol{\theta}) \nabla_{\boldsymbol{x}} g_1^{\mathrm{T}} & g_1 & & \mathbf{0} & & & \\ \vdots & & \ddots & & & \mathbf{0} & \\ \lambda_m(\boldsymbol{\theta}) \nabla_{\boldsymbol{x}} g_m^{\mathrm{T}} & \mathbf{0} & & g_m & & & \\ \hline \nabla_{\boldsymbol{x}} h_1^{\mathrm{T}} & & & & & & \\ \vdots & & \mathbf{0} & & & \mathbf{0} & \\ \nabla_{\boldsymbol{x}} h_p^{\mathrm{T}} & & & & & & \end{array} \right), \tag{3.28}$$

i.e., the Jacobian of the KKT conditions (3.10), (3.11) *and feasibility* (3.7) *with respect to* $[\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}]^{\mathrm{T}}$*, and*

$$\boldsymbol{N} \equiv \big(\ \nabla^2_{\boldsymbol{\theta}\boldsymbol{x}} \mathcal{L}, \lambda_1(\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} g_1, ..., \lambda_m(\boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}} g_m, \nabla_{\boldsymbol{\theta}} h_1, ..., \nabla_{\boldsymbol{\theta}} h_p \ \big)^{\mathrm{T}}, \tag{3.29}$$

i.e., the Jacobian of the KKT conditions and feasibility with respect to $\boldsymbol{\theta}$*.*

Thereby all quantities are evaluated at $[\boldsymbol{x}(\boldsymbol{\theta}), \boldsymbol{\lambda}(\boldsymbol{\theta}), \boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{\theta}]$*.*[2]

[2] For example, $\nabla_{\boldsymbol{\theta}} f$ means $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}(\boldsymbol{\theta}), \boldsymbol{\theta})$, where $\boldsymbol{x}(\boldsymbol{\theta})$ is a local optimal solution of the parametric problem for fixed $\boldsymbol{\theta}$.

Proof A proof can be found in [Fia83].

For a problem with unperturbed objective value function and right-hand-side perturbations of the constraints, one gets from theorem 3.2.11 the following well-known result that establishes the interpretation of the Lagrange multipliers as *shadow prices* for perturbations of the active constraints.

Corollary 3.2.12. *For the problem* $\min\limits_{\boldsymbol{x}\in\mathcal{F}(\boldsymbol{\theta})} f(\boldsymbol{x})$ *with*

$$\mathcal{F}(\boldsymbol{\theta}) = \{\ \boldsymbol{x} \in \mathbb{R}^n \mid g_i(\boldsymbol{x}) \leq \theta_i\ (i \in I = \{1,...,m\}),\ h_j(\boldsymbol{x}) = \theta_{m+j}\ (j \in J = \{1,...,p\})\ \}, \tag{3.30}$$

where $f, \boldsymbol{g}$, *and* $\boldsymbol{h}$ *are twice continuously differentiable, let the assumptions of theorem 3.2.10 be satisfied. Then, in a neighborhood* $B(\mathbf{0}, r)$ *of* $\boldsymbol{\theta} = \mathbf{0}$,

a) $\nabla_{\boldsymbol{\theta}} f^*(\boldsymbol{\theta}) = \begin{pmatrix} -\boldsymbol{\lambda}(\boldsymbol{\theta}) \\ -\boldsymbol{\mu}(\boldsymbol{\theta}) \end{pmatrix}$ *and*

b) $\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}} f^*(\boldsymbol{\theta}) = \begin{pmatrix} -\nabla_{\boldsymbol{\theta}}\boldsymbol{\lambda}(\boldsymbol{\theta}) \\ -\nabla_{\boldsymbol{\theta}}\boldsymbol{\mu}(\boldsymbol{\theta}) \end{pmatrix}$.

[CCC+06] generalizes the sensitivity results of theorem 3.2.11, and proposes a perturbation approach to sensitivity analysis. This method does not require strict complementarity, and it does neither assume active inequality constraints to remain active, nor partial derivatives to exist. To sketch this method, we consider problem (3.20). Let $\boldsymbol{x}(\boldsymbol{\theta})$ be a local solution of this parametric problem and regular, $\boldsymbol{\lambda}(\boldsymbol{\theta}), \boldsymbol{\mu}(\boldsymbol{\theta})$ be the corresponding Lagrange multipliers, and $f^*(\boldsymbol{\theta})$ be the optimal value of the objective function $f(\boldsymbol{x}, \boldsymbol{\theta})$. Consider perturbations of $\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}$, and f^* in such a way that the KKT conditions still hold. Consequently, differentiating the KKT conditions yields the following system of equalities and inequalities describing the set of feasible perturbations, where all quantities are evaluated at $[\boldsymbol{x}(\boldsymbol{\theta}), \boldsymbol{\lambda}(\boldsymbol{\theta}), \boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{\theta}]$, as in theorem 3.2.11:

$$\nabla_{\boldsymbol{x}} f^{\mathrm{T}}\ d\boldsymbol{x} + \nabla_{\boldsymbol{\theta}} f^{\mathrm{T}}\ d\boldsymbol{\theta} - df^* = 0, \tag{3.31a}$$

$$\begin{aligned} &\big(\nabla^2_{\boldsymbol{x}\boldsymbol{x}} f + \sum_{i=1}^{m} \lambda_i(\boldsymbol{\theta}) \nabla^2_{\boldsymbol{x}\boldsymbol{x}} g_i + \sum_{j=1}^{p} \mu_j(\boldsymbol{\theta}) \nabla^2_{\boldsymbol{x}\boldsymbol{x}} h_j\big) d\boldsymbol{x} \\ +&\big(\nabla^2_{\boldsymbol{x}\boldsymbol{\theta}} f + \sum_{i=1}^{m} \lambda_i(\boldsymbol{\theta}) \nabla^2_{\boldsymbol{x}\boldsymbol{\theta}} g_i + \sum_{j=1}^{p} \mu_j(\boldsymbol{\theta}) \nabla^2_{\boldsymbol{x}\boldsymbol{\theta}} h_j\big) d\boldsymbol{\theta} + \nabla_{\boldsymbol{x}}\boldsymbol{g}\ d\boldsymbol{\lambda} + \nabla_{\boldsymbol{x}}\boldsymbol{h}\ d\boldsymbol{\mu} = \mathbf{0}, \end{aligned} \tag{3.31b}$$

$$\nabla_{\boldsymbol{x}} \boldsymbol{h}^{\mathrm{T}}\ d\boldsymbol{x} + \nabla_{\boldsymbol{\theta}} \boldsymbol{h}^{\mathrm{T}}\ d\boldsymbol{\theta} = \mathbf{0}, \tag{3.31c}$$

$$\nabla_{\boldsymbol{x}} g_i^{\mathrm{T}}\ d\boldsymbol{x} + \nabla_{\boldsymbol{\theta}} g_i^{\mathrm{T}}\ d\boldsymbol{\theta} = 0, \quad \text{if } \lambda_i(\boldsymbol{\theta}) \neq 0,\ i \in I(\boldsymbol{x}(\boldsymbol{\theta})), \tag{3.31d}$$

$$\nabla_{\boldsymbol{x}} g_i^{\mathrm{T}}\ d\boldsymbol{x} + \nabla_{\boldsymbol{\theta}} g_i^{\mathrm{T}}\ d\boldsymbol{\theta} \leq 0, \quad \text{if } \lambda_i(\boldsymbol{\theta}) = 0,\ i \in I(\boldsymbol{x}(\boldsymbol{\theta})), \tag{3.31e}$$

$$-d\lambda_i \leq 0, \quad \text{if } \lambda_i(\boldsymbol{\theta}) = 0,\ i \in I(\boldsymbol{x}(\boldsymbol{\theta})), \tag{3.31f}$$

$$d\lambda_i[\nabla_{\boldsymbol{x}} g_i^{\mathrm{T}}\ d\boldsymbol{x} + \nabla_{\boldsymbol{\theta}} g_i^{\mathrm{T}}\ d\boldsymbol{\theta}] = 0, \quad \text{if } \lambda_i(\boldsymbol{\theta}) = 0,\ i \in I(\boldsymbol{x}(\boldsymbol{\theta})). \tag{3.31g}$$

Equations (3.31d)-(3.31g) specify the conditions that directly result from differentiating the inequality constraints, complementarity, and the nonnegativity of $\boldsymbol{\lambda}$'s components, this means in case of inequality constraints and complementarity from

$$\nabla_{\boldsymbol{x}} g_i^{\mathrm{T}}\ d\boldsymbol{x} + \nabla_{\boldsymbol{\theta}} g_i^{\mathrm{T}}\ d\boldsymbol{\theta} \leq 0, \quad \text{if } i \in I(\boldsymbol{x}(\boldsymbol{\theta})), \tag{3.32a}$$

$$(\lambda_i(\boldsymbol{\theta}) + d\lambda_i)(g_i + dg_i) = \lambda_i(\boldsymbol{\theta}) dg_i + d\lambda_i(g_i + dg_i) = 0, \quad \text{if } i \in I(\boldsymbol{x}(\boldsymbol{\theta})). \tag{3.32b}$$

Since $g_i(\boldsymbol{x}(\boldsymbol{\theta}), \boldsymbol{\theta}) = 0\ \forall\ i \in I(\boldsymbol{x}(\boldsymbol{\theta}))$, (3.32b) results in (3.31d) for $\lambda_i(\boldsymbol{\theta}) \neq 0$, and in (3.31g) for $\lambda_i(\boldsymbol{\theta}) = 0$. In case $\lambda_i(\boldsymbol{\theta}) \neq 0$, (3.31d) implies (3.32a), i.e. (3.32a) must be written only for $\lambda_i(\boldsymbol{\theta}) = 0$, which is done by (3.31e).
Note that equation (3.31g) is a second-order constraint. It forces that one of the inequality constraints (3.31e) and (3.31f) is active $\forall\ i \in I(\boldsymbol{x}(\boldsymbol{\theta}))$.
In matrix form, system (3.31a)-(3.31f) can be written as

$$\boldsymbol{P} \left(\begin{array}{c} d\boldsymbol{x} \\ \hline d\boldsymbol{\theta} \\ \hline d\boldsymbol{\mu} \\ \hline d\boldsymbol{\lambda} \\ \hline df^* \end{array}\right) := \left(\begin{array}{c|c|c|c|c} \nabla_{\boldsymbol{x}} f^{\mathrm{T}} & \nabla_{\boldsymbol{\theta}} f^{\mathrm{T}} & \boldsymbol{0} & \boldsymbol{0} & -1 \\ \hline \nabla^2_{\boldsymbol{x}\boldsymbol{x}} \mathcal{L} & \nabla^2_{\boldsymbol{x}\boldsymbol{\theta}} \mathcal{L} & \nabla_{\boldsymbol{x}} \boldsymbol{h} & \boldsymbol{G}_{\boldsymbol{x}}^{\mathrm{T}} & 0 \\ \hline \nabla_{\boldsymbol{x}} \boldsymbol{h}^{\mathrm{T}} & \nabla_{\boldsymbol{\theta}} \boldsymbol{h}^{\mathrm{T}} & \boldsymbol{0} & \boldsymbol{0} & 0 \\ \hline \boldsymbol{G}^1_{\boldsymbol{x}} & \boldsymbol{G}^1_{\boldsymbol{\theta}} & \boldsymbol{0} & \boldsymbol{0} & 0 \end{array}\right) \left(\begin{array}{c} d\boldsymbol{x} \\ \hline d\boldsymbol{\theta} \\ \hline d\boldsymbol{\mu} \\ \hline d\boldsymbol{\lambda} \\ \hline df^* \end{array}\right) = \boldsymbol{0}, \tag{3.33a}$$

$$\boldsymbol{Q} \left(\begin{array}{c} d\boldsymbol{x} \\ \hline d\boldsymbol{\theta} \\ \hline d\boldsymbol{\mu} \\ \hline d\boldsymbol{\lambda} \\ \hline df^* \end{array}\right) := \left(\begin{array}{c|c|c|c|c} \boldsymbol{G}^0_{\boldsymbol{x}} & \boldsymbol{G}^0_{\boldsymbol{\theta}} & \boldsymbol{0} & \boldsymbol{0} & 0 \\ \hline \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I}^0_{|I(\boldsymbol{x}(\boldsymbol{\theta}))|} & 0 \end{array}\right) \left(\begin{array}{c} d\boldsymbol{x} \\ \hline d\boldsymbol{\theta} \\ \hline d\boldsymbol{\mu} \\ \hline d\boldsymbol{\lambda} \\ \hline df^* \end{array}\right) \leq \boldsymbol{0}, \tag{3.33b}$$

where the vector $d\boldsymbol{\lambda}$ is of size $|I(\boldsymbol{x}(\boldsymbol{\theta}))|$ and $\boldsymbol{G}^1_{\boldsymbol{x}}$, $\boldsymbol{G}^1_{\boldsymbol{\theta}}$, $\boldsymbol{G}^0_{\boldsymbol{x}}$, and $\boldsymbol{G}^0_{\boldsymbol{\theta}}$ refer to certain submatrices of

$$\boldsymbol{G}_{\boldsymbol{x}} = \nabla_{\boldsymbol{x}} \boldsymbol{g}(\boldsymbol{x}(\boldsymbol{\theta}), \boldsymbol{\theta})^{\mathrm{T}} \qquad \text{and} \qquad \boldsymbol{G}_{\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} \boldsymbol{g}(\boldsymbol{x}(\boldsymbol{\theta}), \boldsymbol{\theta})^{\mathrm{T}}, \tag{3.34}$$

respectively. Namely, $\boldsymbol{G}^1_{\boldsymbol{x}}$ and $\boldsymbol{G}^1_{\boldsymbol{\theta}}$ denote the submatrices of $\boldsymbol{G}_{\boldsymbol{x}}$ and $\boldsymbol{G}_{\boldsymbol{\theta}}$, respectively, that are associated with the non-zero multipliers λ_i of active inequality constraints. Accordingly, $\boldsymbol{G}^0_{\boldsymbol{x}}$ and $\boldsymbol{G}^0_{\boldsymbol{\theta}}$ denote the submatrices of $\boldsymbol{G}_{\boldsymbol{x}}$ and $\boldsymbol{G}_{\boldsymbol{\theta}}$, respectively, associated with the zero multipliers λ_i of active inequality constraints. $\boldsymbol{I}^0_{|I(\boldsymbol{x}(\boldsymbol{\theta}))|}$ is the $|I(\boldsymbol{x}(\boldsymbol{\theta}))| \times |I(\boldsymbol{x}(\boldsymbol{\theta}))|$ unit matrix after removing all rows $i \in I(\boldsymbol{x}(\boldsymbol{\theta}))$ corresponding to non-zero multipliers $\lambda_i \neq 0$.[3]
To take into account the second order condition (3.31g), system (3.33) must be modified in the following way: for each active inequality constraint $i \in I(\boldsymbol{x}(\boldsymbol{\theta}))$, from (3.33b) either the row associated with the term $\boldsymbol{G}^0$ or the row associated with $-\boldsymbol{I}^0_{|I(\boldsymbol{x}(\boldsymbol{\theta}))|}$ is extracted and added to (3.33a). By this, one decides if an active inequality constraint i

[3] We do not need to consider inactive constraints $g_i, i \in I \setminus I(\boldsymbol{x}(\boldsymbol{\theta}))$, because their Lagrange multipliers are zero, and after a small perturbation they are still inactive and have zero multipliers.

with zero Lagrange multiplier λ_i must remain active after the perturbation or is allowed to become inactive. This means, in total, there are 2^{m_0} possible systems, where $m_0 := |\{i \in I(\boldsymbol{x}(\boldsymbol{\theta})) \mid \lambda_i = 0\}|$.
System (3.31), or, equivalently, the set of all possible modifications of system (3.33), defines the set of all feasible perturbations $(d\boldsymbol{x}, d\boldsymbol{\theta}, d\boldsymbol{\mu}, d\boldsymbol{\lambda}, df^*)^{\mathrm{T}}$ for moving from a KKT solution represented by $\boldsymbol{x}(\boldsymbol{\theta}), \boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{\lambda}(\boldsymbol{\theta}), f^*(\boldsymbol{\theta})$ to another KKT solution.
Let us discuss these results with regard to sensitivities, i.e., with regard to directional and partial derivatives. System (3.33) can be written as

$$\boldsymbol{U} \begin{pmatrix} d\boldsymbol{x} \\ \hline d\boldsymbol{\mu} \\ \hline d\boldsymbol{\lambda} \\ \hline df^* \end{pmatrix} = \boldsymbol{S}\, d\boldsymbol{\theta}, \qquad \boldsymbol{V} \begin{pmatrix} d\boldsymbol{x} \\ \hline d\boldsymbol{\mu} \\ \hline d\boldsymbol{\lambda} \\ \hline df^* \end{pmatrix} \leq \boldsymbol{T}\, d\boldsymbol{\theta}, \tag{3.35a}$$

with

$$\boldsymbol{U} := \left(\begin{array}{c|c|c|c} \nabla_{\boldsymbol{x}} f^{\mathrm{T}} & \boldsymbol{0} & \boldsymbol{0} & -1 \\ \hline \nabla^2_{\boldsymbol{x}\boldsymbol{x}} \mathcal{L} & \nabla_{\boldsymbol{x}} \boldsymbol{h} & \boldsymbol{G}_{\boldsymbol{x}}^{\mathrm{T}} & 0 \\ \hline \nabla_{\boldsymbol{x}} \boldsymbol{h}^{\mathrm{T}} & \boldsymbol{0} & \boldsymbol{0} & 0 \\ \hline \boldsymbol{G}^1_{\boldsymbol{x}} & \boldsymbol{0} & \boldsymbol{0} & 0 \end{array}\right), \quad \boldsymbol{S} := -\left(\begin{array}{c} \nabla_{\boldsymbol{\theta}} f^{\mathrm{T}} \\ \hline \nabla^2_{\boldsymbol{x}\boldsymbol{\theta}} \mathcal{L} \\ \hline \nabla_{\boldsymbol{\theta}} \boldsymbol{h}^{\mathrm{T}} \\ \hline \boldsymbol{G}^1_{\boldsymbol{\theta}} \end{array}\right), \tag{3.35b}$$

$$\boldsymbol{V} := \left(\begin{array}{c|c|c|c} \boldsymbol{G}^0_{\boldsymbol{x}} & \boldsymbol{0} & \boldsymbol{0} & 0 \\ \hline \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I}^0_{|I(\boldsymbol{x}(\boldsymbol{\theta}))|} & 0 \end{array}\right), \qquad \boldsymbol{T} := -\begin{pmatrix} \boldsymbol{G}^0_{\boldsymbol{\theta}} \\ \hline \boldsymbol{0} \end{pmatrix}. \tag{3.35c}$$

Considering condition (3.31g), several possible systems (3.35) may exist, as described above regarding system (3.33).
Each optimal solution of a parametric problem in $\boldsymbol{\theta}$, represented by $\boldsymbol{x}(\boldsymbol{\theta}), \boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{\lambda}(\boldsymbol{\theta})$, $f^*(\boldsymbol{\theta})$, falls into one of the following three categories:

Regular point: The gradient vectors of the active constraints are linearly independent. One can distinguish between

- *Nondegenerate regular point:* All Lagrange multipliers $\lambda_i, i \in I(\boldsymbol{x}(\boldsymbol{\theta}))$, of active inequality constraints are different from zero. Regarding system (3.35), there is no matrix $\boldsymbol{V}$ and the matrix $\boldsymbol{U}$ is invertible.
- *Degenerate regular point:* First case: All Lagrange multipliers $\lambda_i, i \in I(\boldsymbol{x}(\boldsymbol{\theta}))$, of active inequality constraints are different from zero. Regarding system (3.35), there is no matrix $\boldsymbol{V}$ and the matrix $\boldsymbol{U}$ is not invertible. Second case: Some Lagrange multipliers $\lambda_i, i \in I(\boldsymbol{x}(\boldsymbol{\theta}))$, of active inequality constraints are equal to zero. The matrix $\boldsymbol{U}$ is not invertible because it is not a square matrix.

Nonregular point: The gradient vectors of the active constraints are linearly dependent. The matrix $\boldsymbol{U}$ of system (3.35) is never invertible. The Lagrange multipliers are not unique but rather there exist infinite possible combinations of Lagrange multipliers. However, for given values of Lagrange multipliers, sensitivities can be analyzed using

system (3.35) as described below.

In case of a nondegenerate regular point, i.e., if $\boldsymbol{U}$ is invertible, all partial derivatives can be obtained at once by computing

$$\begin{pmatrix} \partial \boldsymbol{x}/\partial \boldsymbol{\theta} \\ \hline \partial \boldsymbol{\mu}/\partial \boldsymbol{\theta} \\ \hline \partial \boldsymbol{\lambda}/\partial \boldsymbol{\theta} \\ \hline \partial f^*/\partial \boldsymbol{\theta} \end{pmatrix} = \boldsymbol{U}^{-1}\boldsymbol{S}. \tag{3.36}$$

Otherwise, if $\boldsymbol{U}$ is not invertible, one can check if directional derivatives[4] exist by replacing $d\boldsymbol{\theta}$ by the corresponding unit vectors and solving the 2^{m_0} possible combinations of system (3.35a) that we described above. If for a certain direction a solution exists for at least one combination and if it is unique, then the corresponding directional derivative exists. Partial derivatives with respect to a component $\theta_j, j \in \{1, ..., l\}$, exist in this case if directional derivatives for $d\theta_j$ and $-d\theta_j$ exist and coincide by absolute value but not in sign. More explicitly, for this purpose, one solves the combinations of system (3.35a) for the perturbation vectors

$$\Big(\frac{\partial x_1}{\partial \theta_j^+}, ..., \frac{\partial x_n}{\partial \theta_j^+}, 0, ..., 0, 1, 0, ..., 0, \frac{\partial \lambda_1}{\partial \theta_j^+}, ..., \frac{\partial \lambda_{|I(\boldsymbol{x}(\boldsymbol{\theta}))|}}{\partial \theta_j^+}, \frac{\partial \mu_1}{\partial \theta_j^+}, ..., \frac{\partial \mu_p}{\partial \theta_j^+}, \frac{\partial f^*}{\partial \theta_j^+}\Big)^{\mathrm{T}} \tag{3.37a}$$

and

$$\Big(\frac{\partial x_1}{\partial \theta_j^-}, ..., \frac{\partial x_n}{\partial \theta_j^-}, 0, ..., 0, -1, 0, ..., 0, \frac{\partial \lambda_1}{\partial \theta_j^-}, ..., \frac{\partial \lambda_{|I(\boldsymbol{x}(\boldsymbol{\theta}))|}}{\partial \theta_j^-}, \frac{\partial \mu_1}{\partial \theta_j^-}, ..., \frac{\partial \mu_p}{\partial \theta_j^-}, \frac{\partial f^*}{\partial \theta_j^-}\Big)^{\mathrm{T}}. \tag{3.37b}$$

The nondegenerate regular case is the most common one. However, if a variation of the parameter $\boldsymbol{\theta}$ is considered, the regular degenerate and the nonregular case occur as important transition situations, for instance at active set changes. Illustrative examples for all cases can be found in [CCC+06, CCMGB06].

3.3 Duality

Duality theory provides a symmetry between a constrained optimization problem and its so-called dual problem. This symmetry turns out to be perfect for convex problems fulfilling Slater's condition. The variables of the dual problem are in this case the Lagrange multipliers of the primal problem given by the Karush-Kuhn-Tucker conditions of theorem 3.1.10. They often allow intuitive interpretations as prices for constrained resources as we will see later on regarding our supply-demand optimization model. The benefit of duality becomes apparent if the dual problem is easier to solve than the original problem, which is mostly the case for separable problems as we will show in section 3.5

[4] See definition 3.4.1.

about decomposition methods. Introductions to Lagrangian duality can be found, e.g., in [BV04, Lue08]. In this section, we summarize the most important results to which we will refer in the remaining part of this thesis.

Consider again the optimization problem in standard form with variable $\boldsymbol{x} \in \mathbb{R}^n$:

$$\begin{aligned} &\min_{\boldsymbol{x}} \ f(\boldsymbol{x}) \\ \text{such that} \quad &g_i(\boldsymbol{x}) \leq 0, \quad i \in I = \{1, ..., m\}, \\ &h_j(\boldsymbol{x}) = 0, \quad j \in J = \{1, ..., p\}. \end{aligned} \tag{3.38}$$

Let its domain $D \subseteq \mathbb{R}^n$ be open and nonempty, $f : D \to \mathbb{R}$, $g_i : D \to \mathbb{R}$ $(i \in I)$, $h_j : D \to \mathbb{R}$ $(j \in J)$, and $p \leq n$.

Definition 3.3.1. *The **Lagrange dual function** $d : \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}$ is defined as the minimum value of the Lagrangian over $\boldsymbol{x}$: for $\boldsymbol{\lambda} \in \mathbb{R}^m, \boldsymbol{\mu} \in \mathbb{R}^p$*

$$d(\boldsymbol{\lambda}, \boldsymbol{\mu}) := \inf_{\boldsymbol{x} \in D} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_{\boldsymbol{x} \in D} \big(f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i g_i(\boldsymbol{x}) + \sum_{i=1}^{p} \mu_i h_i(\boldsymbol{x}) \big). \tag{3.39}$$

*In this regard, the Lagrange multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ are also called **dual variables**.*

d is concave, even if problem (3.38) is not convex, because it is the pointwise infimum of affine functions in $(\boldsymbol{\lambda}, \boldsymbol{\mu})$.

Definition 3.3.2. *The **Lagrange dual problem** associated with problem* (3.38) *is defined as*

$$\max_{\boldsymbol{\lambda}, \boldsymbol{\mu}} \ d(\boldsymbol{\lambda}, \boldsymbol{\mu}) \qquad \textit{such that} \quad \lambda_i \geq 0 \qquad i = 1, ..., m. \tag{3.40}$$

By the definition of the Lagrange dual problem, its optimal value is the best lower bound on the primal optimal value that can be obtained from the Lagrange dual function:

Theorem 3.3.3 (**Weak Duality**)**.** *The optimal value q^* of the Lagrange dual problem* (3.40) *is a lower bound on the optimal value p^* of the primal problem* (3.38)*:*

$$q^* \leq p^*. \tag{3.41}$$

*The difference $p^* - q^*$ is called the **optimal duality gap**.*

Proof For every $(\boldsymbol{\lambda}, \boldsymbol{\mu})$ with $\lambda_i \geq 0, i \in I$, we have

$$\begin{aligned} d(\boldsymbol{\lambda}, \boldsymbol{\mu}) &= \inf_{\boldsymbol{x} \in D} \big(f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i g_i(\boldsymbol{x}) + \sum_{i=1}^{p} \mu_i h_i(\boldsymbol{x}) \big) \\ &\leq \inf_{\boldsymbol{x} \in D, \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0}, \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}} \big(f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i g_i(\boldsymbol{x}) + \sum_{i=1}^{p} \mu_i h_i(\boldsymbol{x}) \big) \\ &\leq \inf_{\boldsymbol{x} \in D, \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0}, \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}} f(\boldsymbol{x}) = p^*. \end{aligned} \tag{3.42}$$

Since this weak duality result is not based on any convexity assumption, it is of great use for nonconvex optimization problems whose dual problems can be solved efficiently. The following theorem states sufficient conditions on the primal problem for a zero duality gap.

Theorem 3.3.4 (**Strong Duality**). *If problem* (3.38) *is convex, and if Slater's condition, given by 3.1.12, is satisfied, the optimal value* q^* *of the dual problem* (3.40) *equals the optimal value* p^* *of the primal problem* (3.38)*:* $q^* = p^*$.

Proof A proof of this theorem including a geometric interpretation can be found in [BV04]. We shortly condense it:
Convexity of the primal problem means $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}$, see example 3.1.7. For simplification assume that $\operatorname{rank}(\boldsymbol{A}) = p$. The sets $\mathcal{A}$ and $\mathcal{B}$ defined by

$$\mathcal{A} := \{(\boldsymbol{u}, \boldsymbol{v}, t) \mid \exists \boldsymbol{x} \in D, g_i(\boldsymbol{x}) \leq u_i, i \in I, \boldsymbol{a}_j^{\mathrm{T}}\boldsymbol{x} - b_j = v_j, j \in J, f(\boldsymbol{x}) \leq t\}, \tag{3.43}$$

$$\mathcal{B} := \{(\boldsymbol{0}, \boldsymbol{0}, s) \in \mathbb{R}^m \times \mathbb{R}^p \times \mathbb{R} \mid s < p^*\} \tag{3.44}$$

are convex and do not intersect. According to the so-called separating hyperplane theorem, there exists $(\tilde{\boldsymbol{\lambda}}, \tilde{\boldsymbol{\mu}}, \nu) \neq \boldsymbol{0}$ and α such that

$$(\boldsymbol{u}, \boldsymbol{v}, t) \in \mathcal{A} \;\Rightarrow\; \tilde{\boldsymbol{\lambda}}^{\mathrm{T}}\boldsymbol{u} + \tilde{\boldsymbol{\mu}}^{\mathrm{T}}\boldsymbol{v} + \nu t \geq \alpha, \text{ and} \tag{3.45}$$

$$(\boldsymbol{u}, \boldsymbol{v}, t) \in \mathcal{B} \;\Rightarrow\; \tilde{\boldsymbol{\lambda}}^{\mathrm{T}}\boldsymbol{u} + \tilde{\boldsymbol{\mu}}^{\mathrm{T}}\boldsymbol{v} + \nu t \leq \alpha. \tag{3.46}$$

(3.45) implies $\tilde{\boldsymbol{\lambda}} \geq \boldsymbol{0}$ and $\nu \geq 0$. (3.46) means $\nu t \leq \alpha \;\forall\; t < p^*$, and hence, $\nu p^* \leq \alpha$. Summarizing this, for all $\boldsymbol{x} \in D$ we have

$$\sum_{i=1}^{m} \tilde{\lambda}_i g_i(\boldsymbol{x}) + \tilde{\boldsymbol{\mu}}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}) + \nu f(\boldsymbol{x}) \geq \alpha \geq \nu p^*. \tag{3.47}$$

In the case $\nu > 0$, we can divide (3.47) by ν and obtain for all $\boldsymbol{x} \in D$

$$\mathcal{L}(\boldsymbol{x}, \tilde{\boldsymbol{\lambda}}/\nu, \tilde{\boldsymbol{\mu}}/\nu) \geq p^*. \tag{3.48}$$

Minimizing this expression over $\boldsymbol{x}$ and defining $\boldsymbol{\lambda} := \tilde{\boldsymbol{\lambda}}/\nu$ and $\boldsymbol{\mu} := \tilde{\boldsymbol{\mu}}/\nu$, we get $d(\boldsymbol{\lambda}, \boldsymbol{\mu}) \geq p^*$. Together with the weak duality of theorem 3.3.3, i.e., $d(\boldsymbol{\lambda}, \boldsymbol{\mu}) \leq p^*$, this leads to $d(\boldsymbol{\lambda}, \boldsymbol{\mu}) = p^*$, which shows that the dual optimum is attained, and that strong duality holds in the case $\nu > 0$.
The case $\nu = 0$ leads to a contradiction to the assumption $\operatorname{rank}(\boldsymbol{A}) = p$.

Remark 3.3.5. (Refinement of Slater's condition) If some inequality constraints $g_i, i \in I_A \subseteq I$, are affine, strong duality holds already under the weaker condition that there exists a feasible point $\boldsymbol{x}$ for which none of the non-affine inequality constraints is active:

$$\exists\, \boldsymbol{x} \in \mathbb{R}^n \;:\; g_i(\boldsymbol{x}) < 0 \;(i \in I \setminus I_A),\; g_i(\boldsymbol{x}) \leq 0 \;(i \in I_A),\; h_j(\boldsymbol{x}) = 0 \;(j \in J), \tag{3.49}$$

cf. [BV04].

Numerical methods that find solutions of the primal problem (3.38) by solving the dual (3.40) are called *dual methods*. They are advantageous if the dual is easier to solve than the primal and if the dual function $d(\boldsymbol{\lambda}, \boldsymbol{\mu})$ is not identically $-\infty$. This approach is also referred to as *Lagrangian relaxation*. [Lem01] treats this topic providing several applications and numerical algorithms. *Primal-dual methods* approximate primal-dual solutions $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ of both problems at the same time, which proves particularly beneficial in conjunction with barrier methods. The result are so-called primal-dual interior point methods, cf. section 3.6.
One difficulty in solving dual problems is that in many types of problems, the dual function is not differentiable. In this case, theory and methods of nonsmooth convex optimization are needed.

3.4 Nonsmooth Convex Optimization

This section is a brief review of the necessary basic theory for solving non-differentiable convex problems. A famous classical book in this domain of convex analysis and subdifferential calculus is [Roc70]. Further remarkable references including comprehensible introductions are [Cla83, HUL93, Ber03]. The textbook [BV04] does not particularly address nonsmooth optimization, however it contains many important results on convex optimization in general. We introduce nonsmooth optimization theory because one of its important applications is in decomposition methods. Decomposition is used to solve large-scale optimization problems, and is notably connected to our model reduction approach. Within our model reduction method, non-differentiability appears in the optimal value functions of the parametric subproblems.

Definition 3.4.1. *$\boldsymbol{s} \in \mathbb{R}^n$ is a **subgradient** of $f : \mathbb{R}^n \to \mathbb{R}$ at $\boldsymbol{x}$ if*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{s}^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{x}) \qquad \forall\ \boldsymbol{y} \in \mathbb{R}^n. \tag{3.50}$$

*The closed, convex set of all subgradients of f at $\boldsymbol{x}$ is called the **subdifferential** of f at $\boldsymbol{x}$. It is denoted by $\partial f(\boldsymbol{x})$.*
*Defining the **directional derivative** of f at $\boldsymbol{x}$ in a direction $\boldsymbol{d} \in \mathbb{R}^n$ by*

$$f'(\boldsymbol{x}, \boldsymbol{d}) := \lim_{t \searrow 0} \frac{f(\boldsymbol{x} + t\boldsymbol{d}) - f(\boldsymbol{x})}{t}, \tag{3.51}$$

the subdifferential can be equivalently defined in terms of the directional derivative:

$$\partial f(\boldsymbol{x}) := \{\boldsymbol{s} \in \mathbb{R}^n |\ \boldsymbol{s}^{\mathrm{T}}\boldsymbol{d} \leq f'(\boldsymbol{x}, \boldsymbol{d})\ \forall\ \boldsymbol{d} \in \mathbb{R}^n\}. \tag{3.52}$$

The subdifferential is a generalization of the gradient to the set of convex functions:

Theorem 3.4.2. *For convex $f : \mathbb{R}^n \to \mathbb{R}$, we have*

a) $\partial f(\boldsymbol{x})$ is nonempty $\forall\ \boldsymbol{x} \in \mathrm{relint}(\mathrm{dom} f)$,

b) if f is differentiable at $\boldsymbol{x}$ then $\partial f(\boldsymbol{x}) = \{\nabla f(\boldsymbol{x})\}$,

c) if $\partial f(\boldsymbol{x}) = \{\boldsymbol{s}\}$ then f is differentiable at $\boldsymbol{x}$ with $\nabla f(\boldsymbol{x}) = \boldsymbol{s}$.

Proof Proofs of these statements can be found, e.g., in [Roc70].

The following theorem summarizes the basic properties of a subdifferential.

Theorem 3.4.3. *a) For $f_1, f_2 : \mathbb{R}^n \to \mathbb{R}$ and $t_1, t_2 \in \mathbb{R}^+$, we have*

$$\partial(t_1 f_1 + t_2 f_2)(\boldsymbol{x}) = t_1 \partial f_1(\boldsymbol{x}) + t_2 \partial f_2(\boldsymbol{x}). \tag{3.53}$$

b) Let $\boldsymbol{A} \in \mathbb{R}^m \times \mathbb{R}^n, \boldsymbol{b} \in \mathbb{R}^m$, $f : \mathbb{R}^m \to \mathbb{R}$ finite and convex, and let $g(\boldsymbol{x}) := f(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$. Then

$$\partial g(\boldsymbol{x}) = \boldsymbol{A}^{\mathrm{T}} \partial f(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}). \tag{3.54}$$

c) Let $f_j : \mathbb{R}^n \to \mathbb{R}\ \forall\ j \in J$ where J is a compact set in some metric space. Assume that $f(\boldsymbol{x}) := \sup_{j \in J} f_j(\boldsymbol{x}) < \infty\ \forall\ \boldsymbol{x} \in \mathbb{R}^n$, and assume that the functions $j \mapsto f_j(\boldsymbol{x})$ are upper semi-continuous on $J\ \forall\ \boldsymbol{x} \in \mathbb{R}^n$. Then

$$\partial f(\boldsymbol{x}) = \mathrm{co}\big(\bigcup \{\partial f_j(\boldsymbol{x}) \mid f_j(\boldsymbol{x}) = f(\boldsymbol{x})\} \big), \tag{3.55}$$

i.e., the convex hull of the union of subdifferentials of "active" functions at $\boldsymbol{x}$.

Proof These properties are proved, e.g., in [HUL93].

Knowing these basic definitions and results concerning subgradients and subdifferentials, we can start to formulate the fundamental results with regard to nonsmooth convex optimization:

Theorem 3.4.4. *For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, the following statements are equivalent:*

a) $\boldsymbol{x}^*$ is a global minimum of f over $\mathbb{R}^n$,

b) $\boldsymbol{0} \in \partial f(\boldsymbol{x}^*)$,

c) $f'(\boldsymbol{x}^*, \boldsymbol{d}) \geq 0\ \forall\ \boldsymbol{d} \in \mathbb{R}^n$.

Proof The equivalences *a)* $\Leftrightarrow$ *b)* and *b)* $\Leftrightarrow$ *c)* directly result from definition 3.4.1.

Regarding constrained convex optimization problems, one obtains a generalization of the KKT conditions of theorem 3.1.13:

Theorem 3.4.5 (**Generalization of KKT Conditions for Non-Differentiable Convex Problems**). *Assume that the convex minimization problem*

$$\begin{aligned} & \min_{\boldsymbol{x}} \ f(\boldsymbol{x}) \\ \textit{such that} \quad & g_i(\boldsymbol{x}) \leq 0, \qquad i \in I = \{1, ..., m\}, \\ & \boldsymbol{a}_j^{\mathrm{T}} \boldsymbol{x} - b_j = 0, \qquad j \in J = \{1, ..., p\}, \end{aligned} \tag{3.56}$$

from example 3.1.7 fulfills the weak[5] *Slater condition. Then the following statements about a feasible point* $\boldsymbol{x}^*$ *are equivalent*

a) $\boldsymbol{x}^*$ *is a global minimum,*

b) $\exists\ \boldsymbol{\lambda} \in \mathbb{R}^m, \boldsymbol{\lambda} \geq \boldsymbol{0}, \boldsymbol{\mu} \in \mathbb{R}^p$ *such that*

$$\boldsymbol{0} \in \partial f(\boldsymbol{x}^*) + \sum_{i=1}^{m} \lambda_i \partial g_i(\boldsymbol{x}^*) + \sum_{j=1}^{p} \mu_j \boldsymbol{a}_j^{\mathrm{T}} \boldsymbol{x}^* \quad \textit{and} \tag{3.57}$$

$$\boldsymbol{\lambda}^{\mathrm{T}} \boldsymbol{g}(\boldsymbol{x}^*) = \sum_{i=1}^{m} \lambda_i g_i(\boldsymbol{x}^*) = 0. \tag{3.58}$$

Proof A proof can be found, e.g., in [Roc70, HUL93].

Furthermore, one derives a generalization of corollary 3.2.12 *a)*:

Theorem 3.4.6. *Let* $f^*(\boldsymbol{k}, \boldsymbol{l})$ *be the optimal value function of the parametric convex problem*

$$\begin{aligned} & \min_{\boldsymbol{x}} \ f(\boldsymbol{x}) \\ \textit{such that} \quad & g_i(\boldsymbol{x}) \leq k_i, \qquad i \in I = \{1, ..., m\}, \\ & a_j^{\mathrm{T}} \boldsymbol{x} = l_j, \qquad j \in J = \{1, ..., p\}. \end{aligned} \tag{3.59}$$

Suppose that Slater's condition holds for a fixed parameter $(\bar{\boldsymbol{k}}, \bar{\boldsymbol{l}})$*. Then we have for the Lagrange multipliers* $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ *(which are the optimal dual variables) corresponding to an optimum* $\boldsymbol{x}^*(\bar{\boldsymbol{k}}, \bar{\boldsymbol{l}})$

$$f^*(\boldsymbol{k}, \boldsymbol{l}) \geq f^*(\bar{\boldsymbol{k}}, \bar{\boldsymbol{l}}) - \sum_{i=1}^{m} \lambda_i^* (k_i - \bar{k}_i) - \sum_{j=1}^{p} \mu_j^* (l_j - \bar{l}_j) \quad \forall\ \boldsymbol{k} \in \mathbb{R}^m, \boldsymbol{l} \in \mathbb{R}^p, \tag{3.60}$$

i.e., $-(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ *is a subgradient of* f^* *at* $(\bar{\boldsymbol{k}}, \bar{\boldsymbol{l}})$*:* $-(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) \in \partial f^*(\bar{\boldsymbol{k}}, \bar{\boldsymbol{l}})$.

[5] Here, the condition of definition 3.1.12 is weakened by requiring a feasible point at which only all the *non-affine* inequality constraints are strictly satisfied, cf. also remark 3.3.5.

Proof The dual problem of (3.59) can be written as

$$\begin{aligned} &\max_{\boldsymbol{\lambda},\boldsymbol{\mu}} \; d(\boldsymbol{\lambda},\boldsymbol{\mu}) - \sum_{i=1}^{m} \lambda_i k_i - \sum_{j=1}^{p} \mu_j l_j \\ &\text{such that} \quad \lambda_i \geq 0, \quad i \in I, \end{aligned} \tag{3.61}$$

where

$$d(\boldsymbol{\lambda},\boldsymbol{\mu}) = \inf_{\boldsymbol{x}} \big(f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i g_i(\boldsymbol{x}) + \sum_{j=1}^{p} \mu_j \boldsymbol{a}_j^{\mathrm{T}} \boldsymbol{x} \big). \tag{3.62}$$

By theorem 3.3.4, Slater's condition for the parameter $(\bar{\boldsymbol{k}}, \bar{\boldsymbol{l}})$ implies strong duality for problems (3.59) and (3.61) when $(\boldsymbol{k}, \boldsymbol{l}) = (\bar{\boldsymbol{k}}, \bar{\boldsymbol{l}})$. Let $(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ be an optimal solution of the dual (3.61) for $(\bar{\boldsymbol{k}}, \bar{\boldsymbol{l}})$.
Furthermore, let $\boldsymbol{x}$ be any feasible point of problem (3.59) for an arbitrary parameter $(\boldsymbol{k}, \boldsymbol{l}) \in \mathbb{R}^{m+p}$, i.e., $g_i(\boldsymbol{x}) \leq k_i, i \in I$, and $\boldsymbol{a}_j^{\mathrm{T}} \boldsymbol{x} = l_j, j \in J$. Then, by strong duality, we have

$$\begin{aligned} f^*(\bar{\boldsymbol{k}}, \bar{\boldsymbol{l}}) &= d(\boldsymbol{\lambda}^*, \boldsymbol{\mu}^*) - \sum_{i=1}^{m} \lambda_i^* \bar{k}_i - \sum_{j=1}^{p} \mu_j^* \bar{l}_j \\ &\leq f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i^* g_i(\boldsymbol{x}) + \sum_{j=1}^{p} \mu_j^* \boldsymbol{a}_j^{\mathrm{T}} \boldsymbol{x} - \sum_{i=1}^{m} \lambda_i^* \bar{k}_i - \sum_{j=1}^{p} \mu_j^* \bar{l}_j \\ &\leq f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i^* k_i + \sum_{j=1}^{p} \mu_j^* l_j - \sum_{i=1}^{m} \lambda_i^* \bar{k}_i - \sum_{j=1}^{p} \mu_j^* \bar{l}_j \\ &= f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i^* (k_i - \bar{k}_i) + \sum_{j=1}^{p} \mu_j^* (l_j - \bar{l}_j). \end{aligned} \tag{3.63}$$

Since this inequality holds for any feasible point $\boldsymbol{x}$ for the parametric problem in $(\boldsymbol{k}, \boldsymbol{l})$, it follows that

$$f^*(\bar{\boldsymbol{k}}, \bar{\boldsymbol{l}}) \leq f^*(\boldsymbol{k}, \boldsymbol{l}) + \sum_{i=1}^{m} \lambda_i^* (k_i - \bar{k}_i) + \sum_{j=1}^{p} \mu_j^* (l_j - \bar{l}_j), \tag{3.64}$$

which is (3.60). This proof generalizes the perturbation inequality deduced in [BV04].

Note that the dual variables of a parametric problem as (3.59) do not in general imply a descent direction of the optimal value function as we point out in the following remark.

Remark 3.4.7. $\boldsymbol{d} \in \mathbb{R}^n$ is a *descent direction* for a function $f : \mathbb{R}^n \to \mathbb{R}$ at $\boldsymbol{x}$ if $f'(\boldsymbol{x}, \boldsymbol{d}) < 0$. For differentiable f, $-\nabla f(\boldsymbol{x})$ is always a descent direction if it is non-zero. However, for non-differentiable convex f, $-\boldsymbol{d}$, with $\boldsymbol{d} \in \partial f(x)$, need not be a descent direction.

The following theorem is another useful result, which we will need in the context of dual decomposition.

Theorem 3.4.8. *Consider again a convex problem*

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}) \quad \textit{such that} \quad \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0}, \quad \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}. \tag{3.65}$$

Suppose that $\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})$ *minimizes the Lagrangian* $\mathcal{L}(\boldsymbol{x}, \bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})$ *for fixed* $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}}) \in \mathbb{R}^{m+p}$. *Then* $-\boldsymbol{g}(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})) - \boldsymbol{h}(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}}))$ *is a subgradient of the negative of the Lagrange dual function,* $-d$, *at* $(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})$, *i.e.,* $-\boldsymbol{g}(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})) - \boldsymbol{h}(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})) \in \partial\big(-d\big)(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})$.

Proof For all $(\boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{R}^{m+p}$, we have

$$\begin{aligned} d(\boldsymbol{\lambda}, \boldsymbol{\mu}) &= \inf_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_{\boldsymbol{x}} \big(f(\boldsymbol{x}) + \boldsymbol{\lambda}^{\mathrm{T}} \boldsymbol{g}(\boldsymbol{x}) + \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{h}(\boldsymbol{x})\big) \\ &\leq f(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})) + \boldsymbol{\lambda}^{\mathrm{T}} \boldsymbol{g}(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})) + \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{h}(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})) \\ &= f(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})) + \bar{\boldsymbol{\lambda}}^{\mathrm{T}} \boldsymbol{g}(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})) + (\boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}})^{\mathrm{T}} \boldsymbol{g}(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})) \\ &\quad + \bar{\boldsymbol{\mu}}^{\mathrm{T}} \boldsymbol{h}(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})) + (\boldsymbol{\mu} - \bar{\boldsymbol{\mu}})^{\mathrm{T}} \boldsymbol{h}(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})) \\ &= d(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}}) + (\boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}})^{\mathrm{T}} \boldsymbol{g}(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})) + (\boldsymbol{\mu} - \bar{\boldsymbol{\mu}})^{\mathrm{T}} \boldsymbol{h}(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})). \end{aligned} \tag{3.66}$$

Multiplying this inequality by "-1", we get that $-\boldsymbol{g}(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})) - \boldsymbol{h}(\boldsymbol{x}(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})) \in \partial\big(-d\big)(\bar{\boldsymbol{\lambda}}, \bar{\boldsymbol{\mu}})$.

Although subgradients do not necessarily imply descent directions, they are commonly used to solve non-differentiable convex optimization problems:

Definition 3.4.9. *A* ***subgradient method*** *is a basic algorithm to approximate a minimum* $\boldsymbol{x}^*$ *of a non-differentiable convex function* $f : \mathbb{R}^n \to \mathbb{R}$ *by choosing*

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha_k \boldsymbol{d}^{(k)}, \tag{3.67}$$

where $\boldsymbol{x}^{(k)} \in \mathbb{R}^n$ *is the* k*-th iterate,* $\boldsymbol{d}^{(k)} \in \mathbb{R}^n$ *is any subgradient of* f *at* $\boldsymbol{x}^{(k)}$, *and* $\alpha_k \in \mathbb{R}^+$ *is the* k*-th step size. Since this is not a descent method, one keeps track of the best point so far, i.e.,*

$$f_{\text{best}}^{(k)} = \min_{i=1,\ldots,k} f(\boldsymbol{x}^{(i)}). \tag{3.68}$$

Classically, step sizes of subgradient methods are fixed before the method is started, i.e., they do not depend on the iterates. This is an essential difference to standard descent methods, where the step size is determined at each iteration by current information. [Ber99, Ber03] introduce subgradient methods in the context of Lagrangian relaxation as the probably most important application of nonsmooth optimization. Since classical subgradient methods converge quite slowly in many examples, several advanced extensions have been proposed. For instance, [NB01] discusses incremental subgradient methods. These methods aim for quickly minimizing a convex function that is

the sum of a large number of component functions, which makes them appropriate for large-scale separable problems. The publication includes convergence results for different non-dynamic and dynamic step size rules. More recent research results related to subgradient methods are given by [KLL07, NO08]. A supplementary and periodically updated chapter 6 of [Ber09] about convex optimization algorithms is provided on the publisher's website. In addition to several advanced subgradient methods, it contains related cutting plane methods and bundle methods, which are also very popular to solve general convex problems.
Cutting plane methods for solving convex problems trace back to the early work [Kel60]. Their basic idea is to construct a piecewise-affine model for the objective f by

$$\hat{f}_k(\boldsymbol{x}) := \max_{i=1,\dots,k} \left(f(\boldsymbol{x}^{(i)}) + \boldsymbol{d}^{(i)\,\mathrm{T}}(\boldsymbol{x} - \boldsymbol{x}^{(i)}) \right), \tag{3.69}$$

where $\boldsymbol{x}^{(i)} \in \mathbb{R}^n$ is the i-th iterate and $\boldsymbol{d}^{(i)} \in \mathbb{R}^n$ is any subgradient of f at $\boldsymbol{x}^{(i)}$. Minimizing this model $\hat{f}_k$ on a predetermined convex compact set, gives a new iterate $\boldsymbol{x}^{(k+1)}$. It is easy to see that

$$\hat{f}_k \leq \hat{f}_{k+1} \quad \text{and} \quad \hat{f}_k \leq f \qquad \forall\, k, \tag{3.70}$$

which is, associated with the objective's convexity, essential to prove this methods' convergence.
Bundle methods improve the classical cutting plane method in view of stability. They keep memory of a bundle of past information, namely $\{f(\boldsymbol{x}^{(i)}), \boldsymbol{x}^{(i)}, \boldsymbol{d}^{(i)}, i = 1, ..., k\}$ and an additional value $\boldsymbol{y}^{(k)}$, the point with smallest objective value so far, called the center. Using this information for the minimization, the polyhedral approximation $\hat{f}_k$ of the objective f is adjusted by a penalizing term that avoids the new iterate $\boldsymbol{x}^{(k+1)}$ moving drastically away from the current center.
A discussion of *descent methods* for nonsmooth problems can be found in [HUL93]. It exhibits that the concept of steepest descent mostly does not lead to practically useful algorithms in the non-differentiable case.

3.5 Decomposition Methods

Decomposition methods are used for the numerical solution of programs that show a certain advantageous structure. They partition the original optimization problem into a sequence of N reduced-dimensional (or easier) local subproblems that are linked by a master problem. During an iteration, each subproblem as well as the master problem are solved once. The master problem determines adjusted values of certain variables, which are fixed when solving the local problems, where the latter can be done either sequentially or in parallel. The master problem is convex if the original problem is. However, it is not necessarily differentiable, even if it results from a differentiable original problem. For this reason, decomposition methods require techniques for optimizing

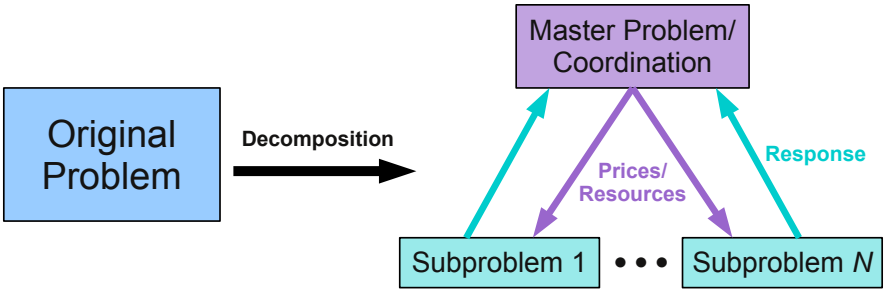

Figure 3.1: *Decomposition of an optimization problem*

nonsmooth problems. The overall procedure is schematically illustrated in figure 3.1. The alternate solving of subproblems and the master problem repeats until sufficient optimality criteria are approximately fulfilled. Decomposition methods are mainly used to solve optimization problems on large-scale systems. An early and influential book on optimization theory for such problems is [Las70], of which several chapters are fundamental and still usable today. For example, it discusses the historically important Dantzig-Wolfe decomposition [DW60] and Benders decomposition [Ben62], as well as the results of [Eve63] about optimal resource allocation. [Ber99, BGLS06, Lue08] include shorter introductions to several decomposition methods and important references. The textbook [CCMGB06] presents a practical approach to decomposition techniques and comprises many illustrative examples. [PC06] surveys alternative decompositions for utility maximization in large networks. [BXMM07] illustrates decomposition of problems with coupling constraints and coupling variables by numerical examples.
The two basic decomposition schemes are *primal decomposition* and *dual decomposition*. The master problem of primal decomposition is an optimization problem in some (possibly newly introduced) variables of the primal problem, the master problem of dual decomposition is in dual variables since it results from Lagrangian relaxation. All other decomposition methods base on one or both of these basic concepts. For example, the known *Benders decomposition* is used to solve special kinds of problems with complicating variables by a primal decomposition. More specifically, it is applied if variables can be partitioned into "easy" and "hard" ones, as is often the case for mixed integer or stochastic programs. [Geo72] enhances the original method [Ben62] to the so-called generalized Benders decomposition, which established one of the standard methods for solving MINLPs.
The following subsections discuss primal and dual decomposition for the most suitable and common problem structures.

a. Decoupling Complicating Constraints

Many practical optimization applications, as [KMT98, CMLW01, BLRS01], show a common block-separable structure, where the components of the variable vector $\boldsymbol{x} \in \mathbb{R}^n$ are

partitioned into N groups, i.e. $\boldsymbol{x} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_N)$, with possibly different numbers of components $n_1, ..., n_N$:

Definition 3.5.1. *An optimization problem is called* ***block-separable*** *if it has the form*

$$\min_{\boldsymbol{x}_1,...,\boldsymbol{x}_N} \sum_{i=1}^{N} f_i(\boldsymbol{x}_i) \qquad \text{such that} \quad \boldsymbol{g}(\boldsymbol{x}) = \sum_{i=1}^{N} \boldsymbol{g}_i(\boldsymbol{x}_i) \leq \boldsymbol{0}, \quad \boldsymbol{h}(\boldsymbol{x}) = \sum_{i=1}^{N} \boldsymbol{h}_i(\boldsymbol{x}_i) = \boldsymbol{0}, \tag{3.71}$$

where $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$, $\boldsymbol{g}_i : \mathbb{R}^{n_i} \to \mathbb{R}^m$, $\boldsymbol{h}_i : \mathbb{R}^{n_i} \to \mathbb{R}^p$, $\forall\ i = 1, ..., N$.[6]
This means that the Hessians of the $1 + m + p$ *functions*

$$f : \mathbb{R}^n \to \mathbb{R},\ f(\boldsymbol{x}) = \sum_{i=1}^{N} f_i(\boldsymbol{x}_i), \tag{3.72}$$

$$g_j : \mathbb{R}^n \to \mathbb{R},\ g_j(\boldsymbol{x}) = \sum_{i=1}^{N} g_{i,j}(\boldsymbol{x}_i),\ j = 1, ..., m,\ \text{and} \tag{3.73}$$

$$h_j : \mathbb{R}^n \to \mathbb{R},\ h_j(\boldsymbol{x}) = \sum_{i=1}^{N} h_{i,j}(\boldsymbol{x}_i),\ j = 1, ..., p, \tag{3.74}$$

where $n = \sum_{i=1}^{N} n_i$, *have a common block-diagonal structure.*
If $n_1 = ... = n_N = 1$, *i.e., each group consists of one variable, the problem is called* ***separable****.*

The $m + p$ constraints within this problem formulation are called *coupling (or complicating) constraints* because they prohibit to solve the overall problem by solving the subproblems of each variable $\boldsymbol{x}_i, i = 1, ..., N$, completely independently. However, despite these coupling constraints, it is possible to take much advantage of such a block-separable structure.
In the following, we present the basic schemes of dual and primal decomposition for block-separable problems with convex f_i, convex $\boldsymbol{g}_i$, and linear $\boldsymbol{h}_i\ \forall\ i = 1, ..., N$, i.e., we deal with convex minimization problems. If a convex problem fulfills in addition Slater's condition, by theorem 3.3.4, its optimal value is equal to the optimal value of its Lagrange dual problem. Otherwise, the optimal value of the dual problem is a lower bound on the optimal value of the primal problem, cf. theorem 3.3.3.

Dual decomposition methods take advantage of the fact that block-separable problems are ideally suited to dual methods because the Lagrangian dual function of a block-separable problem decomposes into N separate subproblems of smaller dimension than

[6] There can be further local constraints, i.e., it can be required that $\boldsymbol{x}_i \in \mathcal{F}_i \subset \mathbb{R}^{n_i}$. In this section, we do not consider this case for ease of notation.

the original problem:

$$\begin{aligned} d(\boldsymbol{\lambda},\boldsymbol{\mu}) &= \inf_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N} \sum_{i=1}^{N} f_i(\boldsymbol{x}_i) + \boldsymbol{\lambda}^{\mathrm{T}} \sum_{i=1}^{N} \boldsymbol{g}_i(\boldsymbol{x}_i) + \boldsymbol{\mu}^{\mathrm{T}} \sum_{i=1}^{N} \boldsymbol{h}_i(\boldsymbol{x}_i) \\ &= \sum_{i=1}^{N} \inf_{\boldsymbol{x}_i} f_i(\boldsymbol{x}_i) + \boldsymbol{\lambda}^{\mathrm{T}} \boldsymbol{g}_i(\boldsymbol{x}_i) + \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{h}_i(\boldsymbol{x}_i) := \sum_{i=1}^{N} d_i(\boldsymbol{\lambda},\boldsymbol{\mu}). \end{aligned} \tag{3.75}$$

The master program of the dual decomposition, given by

$$\max_{\boldsymbol{\lambda}\geq\boldsymbol{0},\boldsymbol{\mu}} d(\boldsymbol{\lambda},\boldsymbol{\mu}) = \max_{\boldsymbol{\lambda}\geq\boldsymbol{0},\boldsymbol{\mu}} \sum_{i=1}^{N} d_i(\boldsymbol{\lambda},\boldsymbol{\mu}), \tag{3.76}$$

chooses dual variables $(\boldsymbol{\lambda},\boldsymbol{\mu}) \in \mathbb{R}^{m+p}$ ("prices")[7], and sends them to the subproblems, which are given $\forall\ i = 1,...,N$ by

$$\inf_{\boldsymbol{x}_i} f_i(\boldsymbol{x}_i) + \boldsymbol{\lambda}^{\mathrm{T}} \boldsymbol{g}_i(\boldsymbol{x}_i) + \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{h}_i(\boldsymbol{x}_i). \tag{3.77}$$

The subproblems respond with their (approximated) optimal values $d_i(\boldsymbol{\lambda},\boldsymbol{\mu})$ and optimal solutions $\boldsymbol{x}_i(\boldsymbol{\lambda},\boldsymbol{\mu})$, $i = 1,...,N$, of these parametric problems in $(\boldsymbol{\lambda},\boldsymbol{\mu})$. This enables the master program to evaluate $d(\boldsymbol{\lambda},\boldsymbol{\mu}) = \sum_{i=1}^{N} d_i(\boldsymbol{\lambda},\boldsymbol{\mu})$ and a subgradient (cf., theorem 3.4.8):

$$-\sum_{i=1}^{N} \big(\boldsymbol{g}_i(\boldsymbol{x}_i(\boldsymbol{\lambda},\boldsymbol{\mu})) + \boldsymbol{h}_i(\boldsymbol{x}_i(\boldsymbol{\lambda},\boldsymbol{\mu}))\big) \in \partial\big(-d\big)(\boldsymbol{\lambda},\boldsymbol{\mu}). \tag{3.78}$$

Making use of this subgradient, adjusted values $(\boldsymbol{\lambda},\boldsymbol{\mu})$ can be determined by calculating one iterate of a nonsmooth optimization method (e.g., a subgradient method or cutting plane method) for solving the master problem (3.76) by minimizing $-d(\boldsymbol{\lambda},\boldsymbol{\mu})$ under the condition $\boldsymbol{\lambda} \geq \boldsymbol{0}$.
When the master problem is solved by a cutting plane method, this dual method turns out to be strongly related to the Dantzig-Wolfe decomposition for large-scale linear programs with a block angular structure.

To perform *primal decomposition* of problem (3.71), for each subproblem i, a variable $(\boldsymbol{t}_i,\boldsymbol{u}_i) \in \mathbb{R}^{m+p}$ is introduced that represents the amount of the resources allocated to this subproblem i. By this, the original problem (3.71) can be reformulated as

$$\begin{aligned} \min_{\substack{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N,\\ \boldsymbol{t}_1,\ldots,\boldsymbol{t}_N,\\ \boldsymbol{u}_1,\ldots,\boldsymbol{u}_N}} \sum_{i=1}^{N} f_i(\boldsymbol{x}_i) \qquad \text{such that} \qquad & \boldsymbol{g}_i(\boldsymbol{x}_i) \leq \boldsymbol{t}_i, \quad \boldsymbol{h}_i(\boldsymbol{x}_i) = \boldsymbol{u}_i \quad \forall\ i = 1,...,N, \\ & \sum_{i=1}^{N} \boldsymbol{t}_i \leq \boldsymbol{0}, \quad \sum_{i=1}^{N} \boldsymbol{u}_i = \boldsymbol{0}. \end{aligned} \tag{3.79}$$

[7] Due to the economical interpretation of dual variables as *shadow prices*, dual decomposition is also known as *price decomposition* in the context of coupling constraints.

The master program declares fixed resource allocations $(\boldsymbol{t}_i, \boldsymbol{u}_i) \in \mathbb{R}^{m+p}$ for which the subproblems are solved.[8] The corresponding optimal values of the subsystems are

$$\phi_i(\boldsymbol{t}_i, \boldsymbol{u}_i) := \min_{\boldsymbol{x}_i} f_i(\boldsymbol{x}_i) \qquad \text{such that} \quad \boldsymbol{g}_i(\boldsymbol{x}_i) \leq \boldsymbol{t}_i, \quad \boldsymbol{h}_i(\boldsymbol{x}_i) = \boldsymbol{u}_i \quad \forall\ i = 1, ..., N, \tag{3.80}$$

and the original problem is equivalent to the master problem

$$\min_{\substack{\boldsymbol{t}_1,...,\boldsymbol{t}_N,\\ \boldsymbol{u}_1,...,\boldsymbol{u}_N}} \sum_{i=1}^{N} \phi_i(\boldsymbol{t}_i, \boldsymbol{u}_i) \qquad \text{such that} \quad \sum_{i=1}^{N} \boldsymbol{t}_i \leq \boldsymbol{0}, \quad \sum_{i=1}^{N} \boldsymbol{u}_i = \boldsymbol{0}, \tag{3.81}$$

where $(\boldsymbol{t}_i, \boldsymbol{u}_i) \in \{(\boldsymbol{t}_i, \boldsymbol{u}_i) \in \mathbb{R}^{m+p} \mid \exists \boldsymbol{x}_i : \boldsymbol{g}_i(\boldsymbol{x}_i) \leq \boldsymbol{t}_i, \boldsymbol{h}_i(\boldsymbol{x}_i) = \boldsymbol{u}_i\}\ \forall\ i = 1, ..., N$, i.e., $(\boldsymbol{t}_i, \boldsymbol{u}_i)$ must be chosen such that each subproblem has at least one feasible solution.
Note that, even if the subproblems are differentiable, the optimal value functions ϕ_i need not be differentiable and thus, in general, the master problem is not differentiable. Under appropriate conditions on the original problem, however, the master problem is convex.
Let $\boldsymbol{\lambda}_i \in \mathbb{R}^m, \boldsymbol{\mu}_i \in \mathbb{R}^p$ be the optimal dual variables associated with the subproblem constraints in (3.80). Then, by theorem 3.4.6, $-(\boldsymbol{\lambda}_i, \boldsymbol{\mu}_i)$ is a subgradient of ϕ_i at $(\boldsymbol{t}_i, \boldsymbol{u}_i)$, and, therefore, $-(\boldsymbol{\lambda}_1, ..., \boldsymbol{\lambda}_N, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_N)^{\mathrm{T}} \in \partial\phi(\boldsymbol{t}_1, ..., \boldsymbol{t}_N, \boldsymbol{u}_1, ..., \boldsymbol{u}_N)$, where

$$\phi(\boldsymbol{t}_1, ..., \boldsymbol{t}_N, \boldsymbol{u}_1, ..., \boldsymbol{u}_N) := \sum_{i=1}^{N} \phi_i(\boldsymbol{t}_i, \boldsymbol{u}_i). \tag{3.82}$$

The master program now determines for all subproblems $i = 1, ..., N$ new, adjusted values $(\boldsymbol{t}_i, \boldsymbol{u}_i)$ by using this subgradient for computing one iterate of a nonsmooth optimization method with the objective to solve (3.81).

b. Decoupling Complicating Variables

In addition to the problem class of definition 3.5.1, there is another class of problems that are eminently suitable for decomposition. Namely, they are of the form

$$\min_{\boldsymbol{x}_1,...,\boldsymbol{x}_N,\boldsymbol{y}} \sum_{i=1}^{N} f_i(\boldsymbol{x}_i, \boldsymbol{y}) \quad \text{such that} \quad \boldsymbol{x}_i \in \mathcal{F}_i\ \forall\ i = 1, ..., N, \quad \boldsymbol{y} \in \mathcal{F}_y, \tag{3.83}$$

where $f_i : \mathbb{R}^{n_i+n_y} \to \mathbb{R}$, and the feasible sets $\mathcal{F}_i, i = 1, ..., N$, and $\mathcal{F}_y$ depend only on the corresponding variables $\boldsymbol{x}_i, \boldsymbol{y}$. We consider again convex problems, i.e., $\mathcal{F}_i$ and $\mathcal{F}_y$ are convex sets and f_i is convex over $\mathcal{F}_i \times \mathcal{F}_y\ \forall\ i = 1, ..., N$. Most appropriately, $\boldsymbol{x}_1, ..., \boldsymbol{x}_N$ have relatively high dimension and $\boldsymbol{y}$ has a relatively small one. For obvious reason,

[8] For this reason, primal decomposition is also known as *resource decomposition* in the context of coupling constraints.

the variable $\boldsymbol{y}$ is called *coupling (or complicating) variable*. As in the previous section, primal or dual decomposition can be applied.

Primal decomposition is performed by fixing $\boldsymbol{y}$ and defining each subproblem i by

$$\min_{\boldsymbol{x}_i} f_i(\boldsymbol{x}_i, \boldsymbol{y}) \quad \text{such that} \quad \boldsymbol{x}_i \in \mathcal{F}_i \tag{3.84}$$

with optimal value function $\phi_i(\boldsymbol{y})$. The original problem is then equivalent to the following master problem in variable $\boldsymbol{y}$

$$\min_{\boldsymbol{y}} \sum_{i=1}^{N} \phi_i(\boldsymbol{y}) \quad \text{such that} \quad \boldsymbol{y} \in \mathcal{F}_y. \tag{3.85}$$

Note that the functions $\phi_i(\boldsymbol{y})$ need not be differentiable. But at least, if the original problem is convex, so is the master problem. A subgradient of the master's objective $\phi(\boldsymbol{y}) := \sum_{i=1}^{N} \phi_i(\boldsymbol{y})$ is given by $\sum_{i=1}^{N} \boldsymbol{s}_i^y \in \partial\phi(\boldsymbol{y})$ with $(\boldsymbol{s}_i^x, \boldsymbol{s}_i^y) \in \partial f_i\big(\boldsymbol{x}_i(\boldsymbol{y}), \boldsymbol{y}\big)$, where $\boldsymbol{x}_i(\boldsymbol{y})$ solves the subproblem i for fixed $\boldsymbol{y}$.[9]

For a *dual decomposition*, problem (3.83) is reformulated with newly introduced variables $\boldsymbol{y}_i$ and consistency constraints:

$$\begin{aligned} \min_{\boldsymbol{x}_1,...,\boldsymbol{x}_N,\boldsymbol{y}_1,...,\boldsymbol{y}_N} \sum_{i=1}^{N} f_i(\boldsymbol{x}_i, \boldsymbol{y}_i) \quad \text{such that} \quad & \boldsymbol{x}_i \in \mathcal{F}_i, \quad \boldsymbol{y}_i \in \mathcal{F}_y \quad \forall\ i = 1, ..., N, \\ & \boldsymbol{y}_i = \boldsymbol{y}_N \quad \forall\ i = 1, ..., N-1. \end{aligned} \tag{3.86}$$

The Lagrange function

$$\mathcal{L}(\boldsymbol{x}_1, ..., \boldsymbol{x}_N, \boldsymbol{y}_1, ..., \boldsymbol{y}_N, \boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_{N-1}) = \sum_{i=1}^{N} f_i(\boldsymbol{x}_i, \boldsymbol{y}_i) + \sum_{i=1}^{N-1} \boldsymbol{\mu}_i^{\mathrm{T}}(\boldsymbol{y}_i - \boldsymbol{y}_N) \tag{3.87}$$

of this problem can be minimized over each $(\boldsymbol{x}_i, \boldsymbol{y}_i) \in \mathcal{F}_i \times \mathcal{F}_y$ separately. Hence, the dual function is given by

$$d(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_{N-1}) = \sum_{i=1}^{N-1} d_i(\boldsymbol{\mu}_i) + d_N(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_{N-1}), \tag{3.88}$$

where

$$\begin{aligned} d_i(\boldsymbol{\mu}_i) &= \inf_{\boldsymbol{x}_i,\boldsymbol{y}_i} \big(f_i(\boldsymbol{x}_i, \boldsymbol{y}_i) + \boldsymbol{\mu}_i^{\mathrm{T}} \boldsymbol{y}_i \big) \quad \forall\ i = 1, ..., N-1, \quad \text{and} \\ d_N(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_{N-1}) &= \inf_{\boldsymbol{x}_N,\boldsymbol{y}_N} \big(f_N(\boldsymbol{x}_N, \boldsymbol{y}_N) - \sum_{i=1}^{N-1} \boldsymbol{\mu}_i^{\mathrm{T}} \boldsymbol{y}_N \big). \end{aligned} \tag{3.89}$$

[9] In general, $\boldsymbol{s}_i^x \neq \boldsymbol{0}$ since each subproblem i can be constrained in $\boldsymbol{x}_i$ by its definition in (3.84).

These minimization problems define the independently solvable subproblems of the decomposition, while the master problem, which adjusts the values $\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_{N-1}$, is in this case the dual problem

$$\max_{\boldsymbol{\mu}_1,...,\boldsymbol{\mu}_{N-1}} d(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_{N-1}) = \max_{\boldsymbol{\mu}_1,...,\boldsymbol{\mu}_{N-1}} \sum_{i=1}^{N-1} d_i(\boldsymbol{\mu}_i) + d_N(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_{N-1}). \tag{3.90}$$

Let $(\boldsymbol{x}_i^*, \boldsymbol{y}_i^*), i = 1, ..., N$, be subproblems' solutions for fixed $(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_{N-1})$. Then, as in the proof of theorem 3.4.8, it can be easily seen that $\forall\, i = 1, ..., N-1 : \ -\boldsymbol{y}_i^* \in \partial\big(-d_i\big)(\boldsymbol{\mu}_i)$ and $(\boldsymbol{y}_N^*, ..., \boldsymbol{y}_N^*) \in \partial\big(-d_N\big)(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_{N-1})$. A subgradient to compute an iteration towards solving the master problem is, therefore, given by $(\boldsymbol{y}_N^* - \boldsymbol{y}_1^*, ..., \boldsymbol{y}_N^* - \boldsymbol{y}_{N-1}^*) \in \partial\big(-d\big)(\boldsymbol{\mu}_1, ..., \boldsymbol{\mu}_{N-1})$.

c. Remarks and Further Decomposition Structures

Primal and dual decomposition of problems with coupling constraints and coupling variables mainly differ in the computation of the subgradients that are relevant to solve the master problem. Indeed, we can transform these problems into each other. Consider, e.g., the reformulation of problem (3.83) that includes coupling variables to problem (3.86) with newly introduced variables. It leads to a block-separable problem with coupling constraints, as defined in (3.71).
This observation can be generalized to the fact that different representations of a certain optimization problem with a (block-)separable structure lead to different decomposition approaches, cf. also [PC06, CLCD07]. As a consequence, the question arises how to decide for a certain decomposition approach. To our knowledge and by [CLCD07], there is hardly any research in this direction, although this topic is of obvious interest, for example due to the influence on convergence speed and on memory requirements. [CLCD07] initialized an analysis by comparing alternative decompositions by two metrics: the

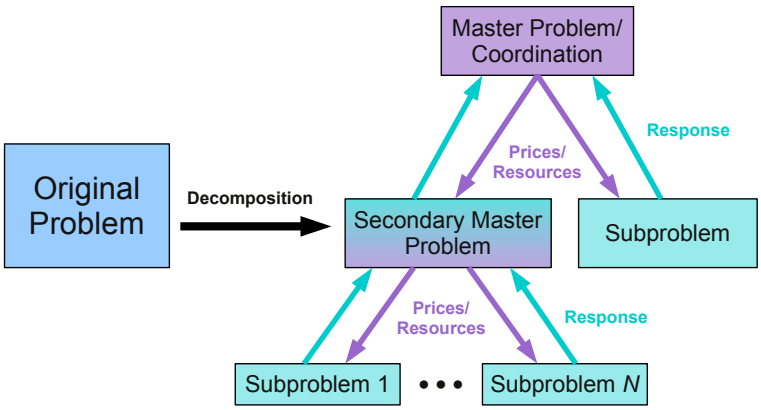

Figure 3.2: *Multilevel decomposition*

tradeoff between local computation and global communication through message passing, and the speed of convergence. However, a systematical exploration and comparison of alternative decompositions remains an open issue. In chapter 5, we discuss in what way concepts and methods from graph theory, as network connectivity and network partitioning, can help to decide for a favorable decomposition.
Advanced decomposition schemes are obtained by applying primal and dual decompositions recursively [PC06]. The result is a *multilevel decomposition* with smaller subproblems from level to level, which is schematically represented by figure 3.2 for the case of two levels.
Since subgradient methods for nonsmooth optimization are relatively slow, several advanced numerical decomposition methods have been proposed to solve large-scale separable convex problems. For example, [NS08, TDSD13, TDNSD13] combine dual decomposition with smoothing [Nes05b] (and excessive gap [Nes05a]), which achieves remarkable computational results.

3.6 Interior Point Methods

Interior point methods build one of the two most successful classes of methods for solving large inequality constrained optimization problems. The so-called active set methods of the other important class solve the problem in each iteration for a newly determined set of inequality constraints that are fixed to equality.[10] For problems that are constrained by a large number of inequalities, the update of this active set may become the bottleneck because of combinatorial complexity. In contrast, interior point methods compute in each iteration an equality-constrained optimum of a composite function that comprises the original objective function as well as the presence of inequality constraints. By this, they reach an optimal solution over a nonlinear path that starts at a strictly feasible point. This path is called the central path.
Numerical methods for large-scale differentiable optimization problems are reviewed in [GOT05]. The survey article [FGW02] summarizes the historical development of interior point methods, comprising classical barrier methods, and details major results, as convergence properties and properties of the central path. [BV04, NW06] include theoretical introductions as well as important references on interior point methods. One of the most essential historical references concerning barrier functions and interior point methods is [FM90], which includes the mathematical theory that was developed by Fiacco and McCormick during the 1960s.
This section outlines the *primal-dual interior point algorithm* with a *filter line-search*

[10] More specifically, in the context of SQP (sequential quadratic programming), one distinguishes between two kinds of active set methods: The first class updates the set of active inequality constraints in an outer loop, and solves equality constrained QPs in inner loop. In the methods of the more popular second class, the QP redetermines the active set at each SQP iteration, where the active set of the QP can be shown to converge to the active set of the original nonlinear program.

method that is implemented by the interior point solver IPOPT (Interior Point OPTimizer) for large-scale nonlinear programs. [Ipo] is the documentation of this solver. Mathematical details and numerical studies of the algorithm can be found in [Wäc02, WB06]. [WB05] analyzes its global convergence.
Consider a nonlinear optimization problem of the form

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) \qquad \text{such that} \qquad \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}, \qquad \boldsymbol{0} \leq \boldsymbol{x}, \tag{3.91}$$

where $f : \mathbb{R}^n \to \mathbb{R}$, $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^p$, with $p \leq n$, twice continuously differentiable. Inequality constraints can be included in this formulation by introducing bounded slack variables (e.g. for $\boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0}$ write $\boldsymbol{g}(\boldsymbol{x}) - \boldsymbol{x}_s = \boldsymbol{0}$ with $\boldsymbol{0} \leq -\boldsymbol{x}_s$).
As an interior point method, IPOPT solves a sequence of barrier problems

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \varphi_k(\boldsymbol{x}) := f(\boldsymbol{x}) - k\sum_{i=1}^{n} \ln(x_i) \qquad \text{such that} \qquad \boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0}, \tag{3.92}$$

for a decreasing sequence of barrier parameters k converging to 0. Since, for fixed $k > 0$, $\varphi_k(\boldsymbol{x}) \to \infty$ if and only if at least one of the variables x_i approaches its bound zero, any solution of problem (3.92) is a strictly feasible point of problem (3.91). Under LICQ and strong second order optimality conditions, the optimal solutions $\boldsymbol{x}^*(k)$ of (3.92) converge to an optimal solution of the original problem (3.91) as $k \to 0$:

Theorem 3.6.1. *Let $\boldsymbol{x}^*$ be a local minimizer of problem (3.91), where $f : \mathbb{R}^n \to \mathbb{R}$, $\boldsymbol{h} : \mathbb{R}^n \to \mathbb{R}^p$, are at least twice continuously differentiable, and where the corresponding feasible region has a strict interior. Let $\boldsymbol{x}^*$ be regular, and let the related KKT-point $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ fulfill strict complementarity (where $\boldsymbol{\lambda}^*$ are the bound multipliers, $\boldsymbol{\mu}^*$ the equality constraint multipliers). Assume furthermore that for the Lagrangian $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{\lambda} + \boldsymbol{h}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\mu}$, there exists $\omega > 0$ such that $\boldsymbol{q}^{\mathrm{T}}\nabla^2_{\boldsymbol{xx}}\mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)\boldsymbol{q} \geq \omega||\boldsymbol{q}||^2$ for all $\boldsymbol{q} \in \mathbb{R}^n \setminus \{\boldsymbol{0}\}$ in the null space of equality and active inequality constraints, i.e. $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}^*)$ fulfills the strong second order sufficient condition.*
Then, we have for the solutions of a sequence of problems (3.92) with decreasing $k_l \to 0$:

a) there exists a subsequence $\boldsymbol{x}(k_l)$ of minimizers of (3.92) converging to $\boldsymbol{x}^*$,

b) for any such convergent subsequence $\boldsymbol{x}(k_l)$, the corresponding sequences of barrier multipliers $\boldsymbol{\lambda}(k_l) = k_l\boldsymbol{X}^{-1}e$ and $\boldsymbol{\mu}(k_l)$ are bounded and converge to the multipliers $\boldsymbol{\lambda}^*$ and $\boldsymbol{\mu}^*$, respectively,

c) a unique, continuously differentiable vector function $\boldsymbol{x}(k)$ of minimizers of problem (3.92) exists for $k > 0$ in a neighborhood of $k = 0$, with $\lim_{k\to 0^+} \boldsymbol{x}(k) = \boldsymbol{x}^*$, and $||\boldsymbol{x}(k_l) - \boldsymbol{x}^*|| = O(k_l)$,

where $\boldsymbol{X} := \mathrm{diag}(\boldsymbol{x})$ and $\boldsymbol{e} := (1 \cdots 1)^{\mathrm{T}}$.

Proof See [FM90, FGW02, Bie10].

Therefore, a possible solution of (3.91) can be approximated by solving the barrier problem (3.92) repeatedly for decreasing k, until a point is found that satisfies the first order optimality conditions up to user tolerance. Such a method splits into inner iterations and outer iterations. The inner iterations solve problem (3.92) for a fixed value of k (where the required accuracy increases for decreasing k), the outer iterations test for convergence and adjust k [FGW02].
Defining $\boldsymbol{\Lambda} := \operatorname{diag}(\boldsymbol{\lambda})$, the first order optimality conditions for the original problem (3.91) and the barrier problem (3.92) can be both expressed by only one equation system

$$\nabla f(\boldsymbol{x}) + \nabla \boldsymbol{h}(\boldsymbol{x})\boldsymbol{\mu} - \boldsymbol{\lambda} = \boldsymbol{0} \tag{3.93a}$$
$$\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{0} \tag{3.93b}$$
$$\boldsymbol{X}\boldsymbol{\Lambda}\boldsymbol{e} - k\boldsymbol{e} = \boldsymbol{0} \tag{3.93c}$$
$$\boldsymbol{x}, \boldsymbol{\lambda} \geq \boldsymbol{0}. \tag{3.93d}$$

For $k = 0$, (3.93) are the KKT conditions of problem (3.91), where $\boldsymbol{\lambda} \in \mathbb{R}^n$ and $\boldsymbol{\mu} \in \mathbb{R}^p$ are the Lagrange multipliers corresponding to bounds and equality constraints. For $k > 0$, the KKT conditions of problem (3.92) are obtained by rewriting (3.93c) by $\boldsymbol{\lambda} = k\boldsymbol{X}^{-1}\boldsymbol{e}$ and plugging this in (3.93a). In addition, sequentially solving (3.93) for decreasing values of $k \to 0$ can be regarded as a homotopy method applied to the primal-dual equations of (3.91) with complementarity conditions relaxed by a homotopy parameter k.
To solve (3.93) for fixed $\boldsymbol{\mu}$, Newton's method can be applied. At a given Newton iterate $(\boldsymbol{x}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)})$, with $\boldsymbol{x}^{(i)}, \boldsymbol{\lambda}^{(i)} \geq \boldsymbol{0}$, search directions $(\boldsymbol{d}_{\boldsymbol{x}}^{(i)}, \boldsymbol{d}_{\boldsymbol{\lambda}}^{(i)}, \boldsymbol{d}_{\boldsymbol{\mu}}^{(i)})$ can be computed by linearizing (3.93):

$$\begin{pmatrix} \boldsymbol{W}^{(i)} & \nabla \boldsymbol{h}(\boldsymbol{x}^{(i)}) & -\boldsymbol{I} \\ \nabla \boldsymbol{h}(\boldsymbol{x}^{(i)})^{\mathrm{T}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{\Lambda}^{(i)} & \boldsymbol{0} & \boldsymbol{X}^{(i)} \end{pmatrix} \begin{pmatrix} \boldsymbol{d}_{\boldsymbol{x}}^{(i)} \\ \boldsymbol{d}_{\boldsymbol{\mu}}^{(i)} \\ \boldsymbol{d}_{\boldsymbol{\lambda}}^{(i)} \end{pmatrix} = - \begin{pmatrix} \nabla f(\boldsymbol{x}^{(i)}) + \nabla \boldsymbol{h}(\boldsymbol{x}^{(i)})\boldsymbol{\mu}^{(i)} - \boldsymbol{\lambda}^{(i)} \\ \boldsymbol{h}(\boldsymbol{x}^{(i)}) \\ \boldsymbol{X}^{(i)}\boldsymbol{\Lambda}^{(i)}\boldsymbol{e} - k\boldsymbol{e} \end{pmatrix}, \tag{3.94}$$

where $\boldsymbol{W}^{(i)}$ is the Hessian of the Lagrange function $\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) - \boldsymbol{x}^{\mathrm{T}}\boldsymbol{\lambda} + \boldsymbol{h}(\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\mu}$ of the original problem (3.91), i.e.,

$$\boldsymbol{W}^{(i)} := \nabla^2_{\boldsymbol{x}\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}^{(i)}, \boldsymbol{\lambda}^{(i)}, \boldsymbol{\mu}^{(i)}) = \nabla^2 f(\boldsymbol{x}^{(i)}) + \sum_{j=1}^{p} \mu_j^{(i)} \nabla^2 h_j(\boldsymbol{x}^{(i)}). \tag{3.95}$$

Instead of solving the nonlinear system (3.94) directly, the IPOPT algorithm computes solutions by solving the smaller, symmetric system

$$\begin{pmatrix} \boldsymbol{W}^{(i)} + \boldsymbol{X}^{(i)^{-1}}\boldsymbol{\Lambda}^{(i)} + \delta\boldsymbol{I} & \nabla \boldsymbol{h}(\boldsymbol{x}^{(i)}) \\ \nabla \boldsymbol{h}(\boldsymbol{x}^{(i)})^{\mathrm{T}} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{d}_{\boldsymbol{x}}^{(i)} \\ \boldsymbol{d}_{\boldsymbol{\mu}}^{(i)} \end{pmatrix} = - \begin{pmatrix} \nabla \varphi_k(\boldsymbol{x}^{(i)}) + \nabla \boldsymbol{h}(\boldsymbol{x}^{(i)})\boldsymbol{\mu}^{(i)} \\ \boldsymbol{h}(\boldsymbol{x}^{(i)}) \end{pmatrix}, \tag{3.96}$$

and then computing the direction $\boldsymbol{d}_{\boldsymbol{\lambda}}^{(i)}$ by

$$\boldsymbol{d}_{\boldsymbol{\lambda}}^{(i)} = k{\boldsymbol{X}^{(i)}}^{-1}\boldsymbol{e} - \boldsymbol{\lambda}^{(i)} - {\boldsymbol{X}^{(i)}}^{-1}\boldsymbol{\Lambda}^{(i)}\boldsymbol{d}_{\boldsymbol{x}}^{(i)}. \tag{3.97}$$

The Hessian regularization parameter $\delta \geq 0$ is introduced to ensure a certain descent property of the direction $\boldsymbol{d}_{\boldsymbol{x}}^{(i)}$. The algorithm adjusts δ such that the upper left block of the matrix in (3.96), projected onto the null space of the constraint Jacobian $\nabla \boldsymbol{h}(\boldsymbol{x}^{(i)})$, is positive definite.
To compute next iterates

$$\boldsymbol{x}^{(i+1)} := \boldsymbol{x}^{(i)} + \alpha_i \boldsymbol{d}_{\boldsymbol{x}}^{(i)}, \tag{3.98a}$$
$$\boldsymbol{\mu}^{(i+1)} := \boldsymbol{\mu}^{(i)} + \alpha_i \boldsymbol{d}_{\boldsymbol{\mu}}^{(i)}, \tag{3.98b}$$
$$\boldsymbol{\lambda}^{(i+1)} := \boldsymbol{\lambda}^{(i)} + \alpha_i^{\boldsymbol{\lambda}} \boldsymbol{d}_{\boldsymbol{\lambda}}^{(i)}, \tag{3.98c}$$

step sizes $\alpha_i, \alpha_i^{\boldsymbol{\lambda}} \in (0, 1]$ are determined by first computing

$$\alpha_i^{\max} := \max\{\ \alpha \in (0,1] \mid \boldsymbol{x}^{(i)} + \alpha \boldsymbol{d}_{\boldsymbol{x}}^{(i)} \geq (1-\tau)\boldsymbol{x}^{(i)}\ \}, \tag{3.99a}$$
$$\alpha_i^{\boldsymbol{\lambda}} := \max\{\ \alpha \in (0,1] \mid \boldsymbol{\lambda}^{(i)} + \alpha \boldsymbol{d}_{\boldsymbol{\lambda}}^{(i)} \geq (1-\tau)\boldsymbol{\lambda}^{(i)}\ \}, \tag{3.99b}$$

with $\tau = \min\{0.99, k\}$, which ensures $\boldsymbol{x}^{(i+1)}, \boldsymbol{\lambda}^{(i+1)} > \boldsymbol{0}$. To attain global convergence, $\alpha_i \in (0, \alpha_i^{\max}]$ is then specified by a backtracking line-search method: A decreasing sequence of trial step sizes $\alpha_{i,l} = 2^{-l}\alpha_i^{\max}$, with $l = 0, 1, 2, ...$, is explored, until a step size $\alpha_{i,l}$ is found that leads to sufficient progress toward a solution of (3.92) compared to the current iterate. More precisely, a trial point $\boldsymbol{x}^{(i+1)}(\alpha_{i,l})$ is accepted if it either improves feasibility, i.e., sufficiently decreases $||\boldsymbol{h}(\boldsymbol{x})||$, or if it sufficiently decreases the objective function $\varphi_k(\boldsymbol{x})$. This idea traces back to [FL02]. To avoid cycles, the new iterate must also show an improvement compared to some previous iterates, which build the "filter". Global convergence of the filter method that is implemented in IPOPT is proved under mild assumptions in [WB05].
In the case that it is not possible to find a step size $\alpha_{i,l}$ that implies an acceptable point $\boldsymbol{x}^{(i+1)}(\alpha_{i,l})$, the IPOPT algorithm switches to the so-called feasibility restoration phase. During this phase, the objective function is ignored, and the algorithm focuses on minimizing the constraint violation $||\boldsymbol{h}(\boldsymbol{x})||$, while diverging as few as possible from the point at which the restoration phase was started.
Furthermore, the IPOPT solver includes second-order corrections in order to improve a proposed step $\tilde{\boldsymbol{d}}_{\boldsymbol{x}}^{(i)} = \alpha_{i,0}\boldsymbol{d}_{\boldsymbol{x}}^{(i)}$ if a trial point $\boldsymbol{x}^{(i+1)} = \boldsymbol{x}^{(i)} + \tilde{\boldsymbol{d}}_{\boldsymbol{x}}^{(i)}$ has been rejected. To reduce infeasibility, the algorithm applies an additional Newton step for the constraints at the point $\boldsymbol{x}^{(i)} + \tilde{\boldsymbol{d}}_{\boldsymbol{x}}^{(i)}$, using the Jacobian $\nabla \boldsymbol{h}(\boldsymbol{x}^{(i)})$.
Once, the barrier problem (3.92) is solved for a fixed value of k, k is decreased. Global convergence of the overall method results from global convergence of the filter method for each barrier problem. If k is updated appropriately, primal-dual interior point

methods can achieve superlinear local convergence under certain standard assumptions [BLN98, GOST01]. The approach implemented in IPOPT follows strategy 2 of [BLN98]. Another widely used solver for large nonlinear optimization problems is KNITRO, which proved robust and efficient in many applications. Its interior point algorithms are described in [BHN99], the theoretical background is given in [BGN00].
Polynomial-time complexity of interior point methods was derived for very general classes of convex optimization problems by [NN94], using the theory of self-concordance. Recent references on this topic are [NT08, BTN].

3.7 Least Squares Methods for Parameter Estimation

The method of least squares is mostly used for parameter estimation in regression analysis. A wide overview of regression models can be found, e.g., in [Fox08]. In the simplest case, the linear regression, we have the following model:
Dependent variables $y_1, ..., y_n \in \mathbb{R}$ (regressands) and independent variables $\boldsymbol{x}_1, ..., \boldsymbol{x}_p \in \mathbb{R}^n$ (regressors) are connected by

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \tag{3.100}$$

where $\boldsymbol{y} = (y_1, ..., y_n)^{\mathrm{T}} \in \mathbb{R}^n$, $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_p) \in \mathbb{R}^{n \times p}$, $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)^{\mathrm{T}} \in \mathbb{R}^p$ and $\boldsymbol{\epsilon} = (\epsilon_1, ..., \epsilon_n)^{\mathrm{T}} \in \mathbb{R}^n$ with $\mathbb{E}(\boldsymbol{\epsilon}) = \boldsymbol{0}$.

The *classical linear model* assumes that

$$\mathrm{Cov}(\boldsymbol{\epsilon}) = \sigma^2 \boldsymbol{I}_n, \quad \text{i.e.,} \quad \mathrm{Var}(\epsilon_i) = \sigma^2, \quad \mathrm{Cov}(\epsilon_i, \epsilon_j) = 0, \text{ for } i \neq j. \tag{3.101}$$

The *general linear model* only assumes

$$\mathrm{Cov}(\boldsymbol{\epsilon}) = \sigma^2 \boldsymbol{\Sigma}, \tag{3.102}$$

where $\boldsymbol{\Sigma}$ is positive-semidefinite and symmetric. Furthermore, in both cases the variable $\boldsymbol{X}$ is assumed to be deterministic. It is also called design matrix. The random variable $\boldsymbol{\epsilon}$ is not observable. $\boldsymbol{\beta}$ and σ^2 are unknown model parameters. $\boldsymbol{\Sigma}$ is sometimes supposed to be known, but could also include unknown parameters. Often, we only have some information about the structure of $\boldsymbol{\Sigma}$.

Usually, we search for estimators of $\boldsymbol{\beta}$ and σ^2. In the classical linear model, $\boldsymbol{\beta}$ is estimated by the method of least squares. Thereby, the estimator $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ is determined by minimizing the quadratic error:

$$\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2 = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^{\mathrm{T}}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = \boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{\epsilon} = \sum_{i=1}^{n} \epsilon_i^2 \;\rightarrow\; \min_{\boldsymbol{\beta}}. \tag{3.103}$$

The solution of this problem is the least squares estimator

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}. \tag{3.104}$$

The difference $\boldsymbol{y} - \hat{\boldsymbol{y}}$ between observed values $\boldsymbol{y} = (y_1, ..., y_n)^{\mathrm{T}}$ and estimated values $\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$ (also named *fit*) is called *residual*.
Defining the *residual sum of squares (RSS)* by

$$\mathrm{RSS} := (\boldsymbol{y} - \hat{\boldsymbol{y}})^{\mathrm{T}}(\boldsymbol{y} - \hat{\boldsymbol{y}}), \tag{3.105}$$

σ^2 is estimated by

$$\hat{\sigma}^2 = \frac{\mathrm{RSS}}{n - \mathrm{rank}(\boldsymbol{X})} \ . \tag{3.106}$$

Properties of the estimators $\hat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ are summarized in the following theorem:

Theorem 3.7.1 (**Gauß-Markov-Theorem**). *In the classical linear model, we have:*

1) The least squares estimator $\hat{\boldsymbol{\beta}}$ is unbiased, i.e., $\mathbb{E}(\hat{\boldsymbol{\beta}}) = \boldsymbol{\beta}$.

2) $\mathrm{Cov}(\hat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X})^{-1}$.

3) $\hat{\boldsymbol{\beta}}$ is the best linear unbiased estimator (BLUE), i.e., $\hat{\boldsymbol{\beta}}$ has minimal variance of all linear unbiased estimators of the form $\tilde{\boldsymbol{\beta}} = \boldsymbol{A}\boldsymbol{y}$ with $\mathbb{E}(\tilde{\boldsymbol{\beta}}) = \boldsymbol{\beta}$:
$\mathrm{Var}(\hat{\beta}_i) \leq \mathrm{Var}(\tilde{\beta}_i)$, for $i = 1, ..., p$.

4) $\hat{\sigma}^2$ is unbiased, i.e., $\mathbb{E}(\hat{\sigma}^2) = \sigma^2$.

Proof A proof can be found, e.g., in [FHT96].

A more general *nonlinear regression model* for a one-dimensional dependent variable $y \in \mathbb{R}$ and an independent variable $\boldsymbol{x} \in \mathbb{R}^p$ is given by

$$y = f(\boldsymbol{x}, \boldsymbol{\theta}) + \epsilon, \quad \mathbb{E}(\epsilon) = 0, \tag{3.107}$$

where $\boldsymbol{\theta}$ is an unknown parameter of an open subset Θ of an m-dimensional real space. The function f is supposed to be at least once continuously differentiable in $\boldsymbol{\theta}$.
Given n observations $y_1, ..., y_n$ for deterministic $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$, we have

$$y_i = f(\boldsymbol{x}_i, \boldsymbol{\theta}) + \epsilon_i, \quad \text{for} \quad i = 1, ..., n. \tag{3.108}$$

A least squares estimator for the parameter $\boldsymbol{\theta}$ is then given by a solution of the optimization problem

$$\min_{\boldsymbol{\theta}} \ \sum_{i=1}^{n} (y_i - f(\boldsymbol{x}_i, \boldsymbol{\theta}))^2. \tag{3.109}$$

Under the further assumption that $\mathrm{Cov}(\boldsymbol{\epsilon}) = \sigma^2\boldsymbol{I}$, we obtain analogously to (3.106) as estimator for σ^2

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} (y_i - f(\boldsymbol{x}_i, \hat{\boldsymbol{\theta}}))^2}{n - m} \ . \tag{3.110}$$

The least squares estimator as solution of (3.109) is in general computed numerically, see also [BKS07]. To prove the usual asymptotic properties of the least squares estimator in the nonlinear regression setting, we have to suppose several sophisticated conditions. They are, e.g., informally discussed in [FHT96].

4 Graph Theory: Selected Concepts and Algorithms

This chapter gives a basic overview of important definitions and concepts of graph theory. Our special focus is on providing a basis for (hyper-)graph partitioning, since, in the following chapters, it will be applied in order to reveal the structure of separable optimization problems with coupling constraints. Further methods and algorithms for the analysis of network data can be found in [Kol09, BE05, New10]. These textbooks also include many illustrating graph visualizations. [BMMN95, Ber98] give surveys of network models and network optimization problems.
Sections 4.1 and 4.2 include basic definitions and algorithms. Section 4.3 introduces the most important centrality measures. In section 4.4, we discuss different notions of connectivity and respective algorithms for computing connectivity. Section 4.5 overviews concepts and methods for graph partitioning. It includes a special subsection that covers hypergraph partitioning. Finally, section 4.6 presents some measures for graph fragmentation.

4.1 Basic Definitions

Definition 4.1.1. *A **graph** $G = (V, E)$ is a mathematical structure that consists of a set V of **vertices** (also called **nodes**) and a set E of **edges**, where an edge $e \in E$ is an unordered pair $\{u, v\}$ of distinct vertices $u, v \in V$. The two vertices defining an edge are said to be **adjacent** to each other, and are also called **neighbors**.*
*A graph is called **directed graph** or **digraph** if its edges have an ordering to their vertices, i.e., if (u, v) is distinct from (v, u). Such edges are called **directed edges** or **arcs**.*
*The format of a graph can be enriched by a **weight function** $w : E \to \mathbb{R}$ that adds to each edge $e \in E$ a weight $w(e)$.*
*A graph $G_1 = (V_1, E_1)$ is a **subgraph** of another graph $G_2 = (V_2, E_2)$ if $V_1 \subseteq V_2$ and $E_1 \subseteq E_2$.*
*Let $V' \subseteq V$, then the **subgraph of G induced by** V' is given by $G' = (V', E')$ where $E' = \{e \in E \mid e = \{v_i, v_j\} \text{ with } v_i, v_j \in V'\}$.*
For a subset $V' \subset V$, $G - V'$ denotes the subgraph of G induced by $V \setminus V'$.
*If the edge set E contains the same edge several times, which is called **multiedges**, we call G a **multigraph**. Otherwise, if each of its edges is contained in E only once, we call G **simple**.*

Extending the notion of edge weights to all pairs of vertices, we can represent the edge

set E of an unweighted graph through a set $\{w(e)|e \in V \times V\}$ of weights by defining

$$w(e) := \begin{cases} 1 & \text{if } e \in E \\ 0 & \text{if } e \notin E \end{cases} . \tag{4.1}$$

Similarly, we define the following matrix that captures the connectivity of a graph:

Definition 4.1.2. *The **adjacency matrix** of a simple directed graph $G = (V, E)$ is defined as the $N_V \times N_V$ matrix $\boldsymbol{A}$ with entries*

$$A_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases} , \tag{4.2}$$

where $N_V := |V|$ and the integers $1, ..., N_V$ denote the elements of V. The adjacency matrix of an undirected graph is symmetric and has $A_{i,j} = 1$ if v_i and v_j are neighbors. If G is weighted, the non-zero entries are $w(v_i, v_j)$ rather than 1.

Definition 4.1.3. *The **degree** d_v of a vertex v is the number of edges containing v. For a vertex v of a directed graph, the **in-degree** d_v^{in} and the **out-degree** d_v^{out} count the number of edges pointing in towards and out from a vertex, respectively.*
*The **maximum** and **minimum degree** of an undirected graph $G = (V, E)$ are given by $\Delta(G) := \max_{v \in V} d_v$ and $\delta(G) := \min_{v \in V} d_v$, respectively.*

Definition 4.1.4. *On a graph G, a **walk** from v_0 to v_n is an alternating sequence $\{v_0, e_1, v_1, e_2, ..., v_{n-1}, e_n, v_n\}$, where $e_i = \{v_{i-1}, v_i\}$ if G is undirected, and $e_i = (v_{i-1}, v_i)$ if G is directed. A walk without repeated edges is called a **trail**, and a trail without repeated vertices is a **path**. The **length of a walk/trail/path** is defined as the number of the edges it passes (or the sum of its edge weights, in case of a weighted graph).*

Definition 4.1.5. *The **distance** $\text{dist}(v, u)$ of any two vertices v and u of a graph $G = (V, E)$ is the length of the shortest path(s) between them. It is set to infinity if no path between the two vertices exists.*

Definition 4.1.6. *A graph $G = (V, E)$ is called **connected** if it is possible to establish a path between every two vertices $u, v \in V$. A **component** of a graph G is a connected subgraph G_c of G for which the addition of any vertex $v \in V \setminus V_c$ would destroy G_c's connectivity.*

Definition 4.1.7. *An undirected graph is called **complete** if every pair of vertices is connected by an edge. A **clique** is a complete subgraph of an undirected graph.*

The following definition generalizes the definition of a graph by allowing edges to connect more than two nodes, which enables to store more detailed information about the connectivity structure of nodes.

Definition 4.1.8. *A **hypergraph** $H = (V, N)$ is a mathematical structure that consists of a set V of vertices (nodes) and a set N of nonempty subsets of V called **hyperedges** or **nets**.*
*The nodes connected by a net $n_i \in N$ are called **Pins** of n_i and are denoted as $\mathrm{Pins}(n_i)$.*
For $N' \subseteq N$, we define $\mathrm{Pins}(N') := \bigcup_{n' \in N'} \mathrm{Pins}(n')$.
*The **induced graph** $G_{ind} = (V, E)$ of a hypergraph is obtained by connecting each two vertices $v_i, v_j \in V$ that are contained in the same hyperedge.*
*The **net intersection graph (NIG)** $G_{NIG} = (V_{NIG}, E_{NIG})$ of a hypergraph H represents each net $n_i \in N$ as a vertex $v_i \in V_{NIG}$, and includes an edge $e_{ij} = \{v_i, v_j\} \in E_{NIG}$ if and only if the corresponding nets $n_i, n_j \in N$ share at least one vertex of H, i.e., if $\mathrm{Pins}(n_i) \cap \mathrm{Pins}(n_j) \neq \emptyset$. This means that each subset of nets that share a vertex of H is a clique in G_{NIG}.*[1]

Hypergraphs are usually used to represent the connection of vertices by common membership of groups of some kind. Such membership of groups can equally be described by a bipartite graph, which is a graph with two kinds of vertices, one representing the original vertices and the other representing the groups to which they belong. Each edge connects a vertex of the first set with a vertex of the second set:

Definition 4.1.9. *A **bipartite graph** is a graph $G = (V, E)$ whose vertex set V may be partitioned into two disjoint sets, $V = V_1 \dot{\bigcup} V_2$, such that each edge in E has one endpoint in V_1 and the other in V_2.*
*A bipartite graph implies two **induced graphs** G_1 and G_2: $G_1 = (V_1, E_1)$ is obtained by defining an edge $e_1 = \{v_i, v_j\}$ between two vertices $v_i, v_j \in V_1$ if and only if v_i and v_j have edges in E to at least one common vertex in V_2. $G_2 = (V_2, E_2)$ is defined analogously.*

Remark 4.1.10. Let a hypergraph $H = (V, N)$ be represented by a bipartite graph $G = (V_1 \dot{\bigcup} V_2, E)$ where $V_1 = V$, V_2 represents the net set N, and two nodes $v_1 \in V_1$ and $v_2 \in V_2$ are connected if the net of H corresponding to v_2 contains the vertex v_1. Then the induced graphs of G are given by the induced graph and the NIG of H: $G_1 = G_{ind}$ and $G_2 = G_{NIG}$.

In general, an induced graph contains less information than the original bipartite graph or hypergraph. Nevertheless, such a reduced graph representation is the most common way to analyze these sophisticated network structures, due to high computational costs of algorithms for hypergraphs or bipartite graphs.
There are some few kinds of networks that are most suitably represented by a directed hypergraph or a directed bipartite graph. However, the edges' directions in an original graph cannot be represented in its induced graph.

[1] Visualizations of a hypergraph's NIG can be found in [KPcA12].

4.2 Basic Graph Algorithms

Given a graph G, e.g., in a simple case in form of its adjacency matrix, efficient algorithms are desired to analyze its structure and to answer certain questions about its properties. Thereby, a basic issue is how to move from a given source vertex through a whole graph within the meaning of visiting all of its vertices. This can be done, e.g., by two common algorithms, Breadth-First Search (BFS) and Depth-First Search (DFS), which are the basis of more sophisticated algorithms developed to solve specific problems in graph theory. In this work, this concerns mainly the detection of (bi-, tri-)connected components as well as the calculation of centralities for which we need, e.g., the computation of shortest paths distances between two vertices. BFS and DFS differ in the order in which the vertices are discovered. Which algorithm is more useful to perform a specific task depends on its context. BFS, DFS, Dijkstra's shortest path algorithm, as well as many other useful algorithms are provided by the Boost Graph Library for C++, cf. [SLL02].
Breadth-First Search (BFS) visits the vertices according to their distance to the source vertex. Closer vertices are discovered prior to more distant vertices, i.e., BFS starts by visiting all neighbors of the source. In contrast, *Depth-First Search (DFS)* visits, beginning at the source vertex, an undiscovered neighbor until it reaches a vertex that has no undiscovered adjacent vertex. If it reaches such a vertex, the algorithm goes back to the previous vertex and checks it for neighbors that are not yet visited. Both of these algorithms run in $O(N_V + N_E)$ time, where $N_V := |V|$ is the number of vertices in G and $N_E := |E|$ is G's number of edges. *Dijkstra's single source shortest path algorithm* for computing the shortest-path distances between one specific source vertex and all other vertices of a graph runs, under certain conditions on its implementation, in time $O(N_E + N_V \log N_V)$ [Dij59, FT87]. The lengths of the shortest paths between all pairs of vertices in a graph can be calculated by the *Floyd-Warshall algorithm* that achieves an $O({N_V}^3)$ running time [Flo62].

4.3 Centrality Measures

A *vertex centrality* is a function $c : V \to \mathbb{R}$. The higher the value, which the function assigns to a vertex, the more 'important' the vertex is for the network in terms of the corresponding definition of centrality. In the following, we present four well-established centrality measures. For this, let $G = (V, E)$ be an undirected connected graph:

Definition 4.3.1. *The **degree centrality** of a vertex $v \in V$ is defined as*

$$c_D(v) := d_v. \tag{4.3}$$

There are several different definitions of closeness centrality. The following classic one that is useful for connected graphs was introduced by [Sab66].

Definition 4.3.2. *The **closeness centrality** of a vertex $v \in V$ is defined as*

$$c_C(v) := \frac{1}{\sum_{u \in V} \mathrm{dist}(v,u)}. \tag{4.4}$$

[Bon72] introduced a measure based on the spectral decomposition of the adjacency matrix $\boldsymbol{A}$ of G:

Definition 4.3.3. *The **eigenvector centrality** of the vertices $v \in V$ is defined by the following system of equations*

$$c_E(v) = \frac{1}{\lambda} \sum_{\{v,u\} \in E} c_E(u), \tag{4.5}$$

where λ is the largest eigenvalue of $\boldsymbol{A}$.

This means, the vector $\boldsymbol{c}_E = (c_E(v_1), ..., c_E(v_{N_V}))^{\mathrm{T}}$ is the eigenvector corresponding to the eigenvalue λ, fulfilling $\boldsymbol{A}\boldsymbol{c}_E = \lambda \boldsymbol{c}_E$, where $\boldsymbol{A}$ is the adjacency matrix of G.[2]

The concept of betweenness centrality was introduced by [Fre77].

Definition 4.3.4. *The **(absolute) betweenness centrality** of a vertex $v \in V$ is defined as*

$$c_B(v) := \sum_{s \neq t \neq v \in V} \frac{\sigma(s,t|v)}{\sigma(s,t)}, \tag{4.6}$$

where $\sigma(s,t|v)$ is the total number of shortest paths between vertices s and t that pass through v, and $\sigma(s,t) := \sum_v \sigma(s,t|v)$.

That means, if all shortest paths of a graph are unique, $c_B(v)$ just counts the number of shortest paths going through v. The absolute betweenness centrality $c_B(v)$ is scaled to the *relative betweenness centrality* $c'_B(v)$ by the factor $\frac{2}{N_V^2 - 3N_V + 2}$, where N_V is the total number of vertices in the graph. $\frac{N_V^2 - 3N_V + 2}{2}$ is the upper limit of $c_B(v)$ for all v. A natural measure for the dominance of the most central point v^* of a graph is given by

Definition 4.3.5. *The **central point dominance** of a graph is defined as*

$$c'_B := \frac{\sum_{v \in V} c'_B(v^*) - c'_B(v)}{N_V - 1}, \tag{4.7}$$

where $c'_B(v)$ is the relative betweenness centrality of a vertex v, and v^* is the most central point of the graph.

[2] One variant of the eigenvector centrality is PageRank, which is used by the search engine Google to analyze Web page linking.

The central point dominance is zero for all graphs for which the betweenness centralities of all points are equal. Its value is 1 only for a wheel or a star.
The betweenness centrality of individual vertices was naturally extended to betweenness centrality of groups by [EB99]:

Definition 4.3.6. *The **group betweenness centrality** of a set of vertices $V' \subseteq V$ is defined as*

$$c_{GB}(V') := \sum_{s \neq t \in V \setminus V'} \frac{\sigma(s,t|V')}{\sigma(s,t)}, \tag{4.8}$$

where $\sigma(s,t|V')$ is the number of shortest paths between s and t that traverse at least one vertex of the set V'.

[PED07] proposes a fast algorithm for successive computation of group betweenness centrality to find the most prominent group of vertices in a graph.

4.4 Connectivity

A survey of results and algorithms concerning graph connectivity is given by chapter 7 of [BE05]. We summarize the most important results with regard to this thesis. Therefore, we consider an undirected graph $G = (V, E)$ in this section.

Definition 4.4.1. *G is called k-**vertex-connected** if $|V| > k$ and $G - V'$ is connected for every $V' \subset V$ with $|V'| < k$. The **vertex-connectivity** $\kappa(G)$ is the largest $k \in \mathbb{N}_0$ such that G is k-vertex-connected.*
*G is k-**edge-connected** if $|V| \geq 2$ and $G' = (V, E \setminus E')$ is connected for every $E' \subseteq E$ with $|E'| < k$. The **edge-connectivity** $\lambda(G)$ is the largest $k \in \mathbb{N}_0$ such that G is k-edge-connected.*
*The **local vertex-connectivity** $\kappa_G(s,t)$ of two non-adjacent vertices $s, t \in V$ is defined as the minimum number of vertices in $V \setminus \{s,t\}$ that must be removed to destroy all paths between s and t. For adjacent vertices s, t, $\kappa_G(s,t)$ is set to $N_V - 1$. Similarly, the **local edge-connectivity** $\lambda_G(s,t)$ of $s, t \in V$ is the minimum number of edges of G that must be removed to destroy all paths between s and t.*

Definition 4.4.2. *A subset $V' \subset V$ with $|V'| = n$ ($E' \subseteq E$ with $|E'| = n$) of a connected graph G is called an (n-)**vertex-cut (edge-cut)**, or an (n-)**vertex-separator (edge-separator)**, if its removal from G disconnects G.*
*Let $s, t \in V$ and $V' \subseteq V \setminus \{s,t\}$. Then V' is called an s-t-**vertex-separator** or (**-cut**) if s and t are in the same connected component of G, but in different connected components of $G - V'$. An s-t-**edge-separator** (**-cut**) is defined analogously.*

Theorem 4.4.3. *For all non-trivial graphs G it holds that $\kappa(G) \leq \lambda(G) \leq \delta(G)$.*

Proof See, e.g., [BE05].

The following theorem is one of the most fundamental statements in graph theory. It results from Karl Menger's early work on general curve theory [Men27].

Theorem 4.4.4 (**Menger's Theorem**). *Let $s, t \in V$ be two non-adjacent vertices of G. Then the maximum number of vertex-disjoint paths between s and t is equal to the minimum cardinality of an s-t-vertex-separator.*

A similar result holds for edge-cuts. It is often called the edge version of Menger's Theorem although it was first explicitly stated several years later, cf. e.g. [FF56]:

Theorem 4.4.5. *For $s, t \in V$, the maximum number of edge-disjoint paths between s and t is equal to the minimum cardinality of an s-t-edge-separator.*

A global version of Menger's Theorem is given by

Theorem 4.4.6. *A non-trivial graph G is k-vertex-connected (k-edge-connected) if and only if all pairs of distinct vertices can be connected by k vertex-disjoint (edge-disjoint) paths.*

Proof The vertex version is proved in [Whi32]. The edge version follows directly from theorem 4.4.5.

Since many algorithms for the computation of vertex- and edge-connectivity work by solving max-flow-problems to compute local connectivities, the following definition is necessary to understand them:

Definition 4.4.7. *A **flow network** is a directed graph $G = (V, E)$ with a function $u : E \to \mathbb{R}_0^+$, which assigns non-negative capacities to the edges, and two distinct vertices $s, t \in V$ constituting the **source** and **sink**, respectively.*
*An **s-t-flow** is a function $f : E \to \mathbb{R}_0^+$ satisfying capacity and balance constraints*

$$0 \leq f(e) \leq u(e) \qquad \forall\, e \in E \qquad \text{and} \tag{4.9a}$$

$$\sum_{e \in \Gamma^-(v)} f(e) = \sum_{e \in \Gamma^+(v)} f(e) \quad \forall\, v \in V \setminus \{s, t\}, \tag{4.9b}$$

*where $\Gamma^+(v)$ is the set of edges with origin v, and $\Gamma^-(v)$ is the set of edges with destination v. The **value** of a flow f is defined as*

$$\sum_{e \in \Gamma^+(s)} f(e) - \sum_{e \in \Gamma^-(s)} f(e). \tag{4.10}$$

*It represents the amount of flow passing from the source to the sink. The problem of computing a flow of maximum value is called the **max-flow problem**.*

Many polynomial-time algorithms have been developed for solving the max-flow problem. For example, [GT88] provides an algorithm that solves this problem on a graph $G = (V, E)$ in $O(N_V N_E \log(N_V^2/N_E))$ time, where $N_V = |V|$ and $N_E = |E|$. A list of alternative algorithms can be found in [BE05]. The Boost Graph Library (BGL) [SLL02] includes a selection of them with different requirements on the input graph.
The following classical result concerning network flows traces back to [FF56]. It is closely related to theorem 4.4.5 and the reason why vertex- and edge-connectivity of a graph are most often computed by solving a sequence of max-flow problems.

Theorem 4.4.8 (**Max-Flow Min-Cut Theorem**). *The value of a maximum s-t-flow is equal to the capacity*[3] *of a minimum s-t-edge-cut.*

[GH61] shows that $N_V - 1$ calls to a max-flow algorithm suffice to compute a maximum flow (and minimum edge-cut) between all pairs of vertices. A classical detailed reference on network flows is [AMO93].

Algorithms for vertex- and edge-connectivity

Algorithms for vertex- and edge-connectivity usually exploit that

$$\kappa(G) = \min_{s \neq t \in V} \kappa_G(s,t) \qquad \text{and} \qquad \lambda(G) = \min_{s \neq t \in V} \lambda_G(s,t). \tag{4.11}$$

A popular simple algorithm that computes a minimum edge-cut corresponding to $\lambda(G)$ was developed by [SW97]. It is implemented in the Boost Graph Library (BGL) [SLL02]. In contrast to most other connectivity algorithms, it is not based on the computation of maximum flows. [BE05] includes an overview of several max-flow based methods and their time complexities, where algorithms that check k-vertex/edge-connectivity of a graph G for $k \in \mathbb{N}$ are distinguished from algorithms that compute the vertex/edge-connectivity $\kappa(G)$ and $\lambda(G)$, respectively. For undirected graphs, these problems are usually solved by constructing suitable directed graphs on which the local connectivity between distinct vertices s and t is computed by solving the related max-flow problem. For example, [Eve79] proves that the local vertex-connectivity $\kappa_G(s,t)$ for vertices $s, t \in V$ of an undirected graph $G = (V, E)$ can be computed by defining the following directed graph $\widetilde{G} = (\widetilde{V}, \widetilde{E})$ with $|\widetilde{V}| = 2N_V$ and $|\widetilde{E}| = 2N_E + N_V$: Each vertex $v \in V$ is replaced by two vertices $v', v'' \in \widetilde{V}$ which are connected by a directed edge $e_v = (v', v'') \in \widetilde{E}$. Furthermore, every edge $e = \{v, u\} \in E$ is replaced by two edges $e' = (v'', u'), e'' = (u'', v') \in \widetilde{E}$. The maximum flow from source s'' to the target t' in $\widetilde{G}$ with unit capacities for all edges is then equal to the local vertex-connectivity $\kappa_G(s,t)$. Usually, the output of connectivity algorithms does not include the related vertex- and edge-separators. However, some max-flow algorithms can be easily extended to additionally find particular edges that constitute a minimum edge-separator. Chapter 10 of

[3] The capacity of an s-t-edge-cut is defined as the sum of capacities of the edges with origin in the component including the vertex s and destination in the component including t.

[New10], for instance, gives an explanation about how this can be done in case of the augmenting path algorithm [FF56].

Graphs that do not include any 1-vertex-cut (also called cut-vertex) or even any 2-vertex-cut (also called separation pair) are said to be biconnected or triconnected, respectively:

Definition 4.4.9. *A connected graph $G = (V, E)$ is called **biconnected** if each subgraph induced by a vertex set $V \setminus \{v_i\}$, $v_i \in V$, is connected.*

Definition 4.4.10. *A biconnected graph $G = (V, E)$ is called **triconnected** if each subgraph induced by a vertex set $V \setminus \{v_i, v_j\}$, $v_i, v_j \in V$, is connected.*

Algorithms for biconnected and triconnected components

The following algorithms are based on DFS. To compute biconnected components of a graph, [Tar72] proposes a linear time algorithm, which is implemented, e.g., in the Boost Graph Library (BGL) [SLL02]. Triconnected components result from the representation of a biconnected graph by a so-called SPQR-tree. An algorithm that constructs the SPQR-tree of a graph in linear time was originally proposed by [HT73]. However, it included faulty parts that were later on revealed and corrected by [GM01]. There is not any SPQR-tree algorithm implemented in the BGL until now, but in the Open Graph Drawing Framework (OGDF), which is a self-contained C++ class library for the automatic layout of graphs [CGJ+13][4].

[Oel87] proposes another concept that is interesting in the context of dividing a graph into more than two components:

Definition 4.4.11. *For an integer $l \geq 2$, the **l-connectivity** of a graph $G = (V, E)$ is the minimum number of vertices whose removal from G produces a disconnected graph with at least l components or a graph with fewer than l vertices. For $n \geq 0$, G is called **(n, l)-connected** if its l-connectivity is at least n.*

4.5 Graph Partitioning

In this section, we focus on the partitioning of undirected graphs. However, many concepts similarly exist for directed graphs. Basically, one distinguishes between two kinds of partitioning, namely graph partitioning by edge-separators (GPES), also called edge-cut partitioning, and graph partitioning by vertex-separators (GPVS), also called vertex-cut partitioning, cf. [KC12, KPcA12]. GPES is more often used, and leads to a partition of a graph $G = (V, E)$ in the original sense:

Definition 4.5.1. *A **(k-way) partition** of a graph $G = (V, E)$ is a decomposition $\Pi = \{V_1, ..., V_k\}$ of its set of vertices V into k disjoint, nonempty subsets V_i such that $\bigcup_{i=1}^{k} V_i = V$.*

[4]The OGDF can be obtained from the website: http://www.ogdf.net

The classical GPES problem consists in detecting a partition $\Pi = \{V_1, ..., V_k\}$ of a graph G's set of vertices in such a manner that the sets $E(V_i, V_j)$ of edges connecting elements of V_i to those of V_j are relatively small in size compared to the sets $E(V_i, V_i)$ connecting vertices within the V_i. In other words, one aims for an edge-separator of small size whose removal decomposes the graph into components of high connectivity. We denote the partition that an edge-separator E_s induces by $\Pi_{E_s}(G)$.
In contrast, the partition induced by a vertex-separator V_s consists of the components $V_1, ..., V_k$ of $G - V_s$ plus the vertex-cut itself, i.e., $\Pi_{V_s}(G) = \{V_1, ..., V_k; V_s\}$.

Definition 4.5.2. *We call a separator that induces a k-way partition **narrow** if no subset of it induces a k-way partition, and **wide** otherwise.*

Furthermore, we define the connectivity of an element of a vertex-separator by

Definition 4.5.3. *The number of parts $V_i, i \in \{1, ..., k\}$, that a vertex $v_s \in V_s$ connects is called the **connectivity** of v_s. We denote it by $\lambda(v_s)$.*

Lemma 4.5.4. *A vertex-cut V_s inducing a vertex partition $\Pi_{V_s}(G) = \{V_1, ..., V_k; V_s\}$ is narrow if and only if every vertex $v_s \in V_s$ connects at least two parts, i.e., $\lambda(v_s) \geq 2$.*

Proof a) Let us assume that there is a vertex $v_s \in V_s$ with $\lambda(v_s) < 2$. If $\lambda(v_s) = 1$, we can place v_s to the part $V_i, i \in \{1, ..., k\}$ that v_s connects. Otherwise, if $\lambda(v_s) = 0$, we can place it to any part $V_i, i \in \{1, ..., k\}$, and $V_s \setminus \{v_s\}$ is a valid separator. Thus V_s is not narrow.
b) Let us assume that V_s is not narrow. Then there exists a valid separator $V_s' \subset V_s$. Assume that $\lambda(v_s') \geq 2$ for any $v_s' \in V_s \setminus V_s'$. Then there are two parts $V_i, V_j, i \neq j \in \{1, ..., k\}$ that are connected by v_s', which contradicts the validity of the separator V_s'.

Obviously, we are interested in partitions that are induced by narrow separators. The request for a small separator can be further specified using one of the following cut-size definitions:

Definition 4.5.5. *Basically, the **cut-size** of a partition $\Pi_{E_s}(G)$ or $\Pi_{V_s}(G)$ is defined as the size $|E_s|$ of the edge-separator or the size $|V_s|$ of the vertex-separator, respectively. If edges or nodes of a graph are accompanied by specific costs (e.g. weights) $c(e) \geq 0$ or $c(v) \geq 0$, the cut-sizes are given as $\sum_{e_s \in E_s} c(e_s)$ and $\sum_{v_s \in V_s} c(v_s)$, respectively.*
Taking further account of the connectivity of all $v_s \in V_s$, the cut-size of a partition $\Pi_{V_s}(G) = \{V_1, ..., V_k; V_s\}$ can alternatively be defined as $\sum_{v_s \in V_s} c(v_s)(\lambda(v_s) - 1)$, cf. also definition 4.5.15.

Besides the request for a small cut-size, further requirements on GPES and GPVS can be a predetermined number of components into which the graph must at least decompose, or that the set of components $\{G(V_1, E_1), ..., G(V_k, E_k)\}$ satisfies some prespecified criterion, where $E_i = \{e \in E \mid e = \{v_l, v_m\} \text{ with } v_l, v_m \in V_i\}, i = 1, ..., k$. For instance, such a quality criterion quantifies the balance of a partition or graph fragmentation, see below.

These quality criteria are introduced to control size and structure of the components, and to preclude separators that cut off only small parts of the graph. Which special requirements a partition should fulfill, and which criterion should be used to compare different partitions of a graph strongly depends on the respective application.

a. Quality Criteria for Graph Partitions

This section presents the two quality measures modularity and expansion. Both concern balance of a partition and were originally introduced for GPES. [KC12] adapts them to GPVS. Furthermore, there are measures for graph fragmentation introduced by [Bor06] to detect vertex-separators that induce partitions with low intra-component connectivity, cf. section 4.6.

Definition 4.5.6. *The **modularity** of a partition* $\Pi_{E_s}(G) = \{V_1, ..., V_k\}$ *induced by an edge-cut* E_s *is defined as*

$$\mathrm{m}_{ES}(\Pi_{E_s}) = \sum_{i=1}^{k} \left(\frac{|E(V_i, V_i)|}{|E|} - \left(\sum_{j \neq i} \frac{|E(V_i, V_j)|}{|E|} \right)^2 \right), \tag{4.12}$$

where $E(V_i, V_j), i \neq j \in \{1, ..., k\}$*, is the set of edges connecting elements of* V_i *to those of* V_j *and* $E(V_i, V_i), i \in \{1, ..., k\}$*, is the set of edges connecting vertices within* V_i*.*

This definition traces back to [GN04]. The modularity measures the fraction of the edges that connect vertices of the same component minus the expected value of the same quantity in a graph with the same vertex partition but random connections between the vertices. If the number of intra-component edges is no better than random, we will get $\mathrm{m}_{ES}(\Pi_{E_s}) \leq 0$. The maximal value of $\mathrm{m}_{ES}(\Pi_{E_s})$ is 1, which is reached if $E_s = \emptyset$. The problem of maximizing modularity over all possible partitions of a network has been shown to be $\mathcal{NP}$-complete [BDG+08]. A fast heuristic algorithm for maximizing modularity by spectral analysis of the so-called modularity matrix is given by [New06]. Alternative methods are, for example, simulated annealing [DDGDA05] and greedy algorithms [New04].
[KC12] proposes two possible adaptions of the modularity measure to partitions induced by vertex-separators.[5] These two definitions arise since either the number of vertices or the number of edges within a component can be counted for the first term of the formula:

[5] Note that [KC12] defines the connected components of a vertex-cut based partition in an augmented way: in distinction from our definition, a component i additionally includes the edges in E_s incident to the vertices V_i and also the vertices in V_s adjacent to V_i. As a result, the components are overlapping. To keep consistency, we will handle this discrepancy by defining additional sets of the controversially treated elements in definitions 4.5.7 and 4.5.10.

Definition 4.5.7. *The **modularity** of a partition $\Pi_{V_s}(G) = \{V_1, ..., V_k; V_s\}$ induced by a vertex-cut V_s can be measured by*

$$\mathrm{m1}_{VS}(\Pi_{V_s}) = \sum_{i=1}^{k} \left(\frac{|V_i| + |V_{s,i}|}{|V|} - \left(\sum_{j \neq i} \frac{|V_{s,ij}|}{|V|} \right)^2 \right), \tag{4.13}$$

or

$$\mathrm{m2}_{VS}(\Pi_{V_s}) = \sum_{i=1}^{k} \left(\frac{|E(V_i)|}{|E|} - \left(\sum_{j \neq i} \frac{|V_{s,ij}|}{|V|} \right)^2 \right), \tag{4.14}$$

where $V_{s,i}$ is the set of vertices $v_s \in V_s$ adjacent to V_i, $V_{s,ij} := V_{s,i} \cap V_{s,j}$ is the set of vertices $v_s \in V_s$ that connect the sets V_i and V_j, and $E(V_i) := \{e \in E \mid e \cap V_i \neq \emptyset\}$.

[KC12] presents a vertex-cut based partitioning algorithm on the basis of structural balance vertices, called SBV-cut. It performs well, both in terms of the modularity measures and in terms of expansion, see definition 4.5.10.
Expansion was originally defined by [KVV04] for bipartitions induced by edge-cuts, and accordingly for graphs. The expansion of a bipartition measures its relative cut-size:

Definition 4.5.8. *The **expansion of a bipartition** $\Pi_{E_s^b}(G) = \{V_1, V_2\}$ induced by an edge-cut E_s^b is defined as*

$$\mathrm{e}^b_{ES}(\Pi_{E_s^b}) = \frac{|E_s^b|}{\min\{|V_1|, |V_2|\}} = \frac{|E(V_1, V_2)|}{\min\{|V_1|, |V_2|\}}. \tag{4.15}$$

*The **expansion of a graph** G is the minimum expansion over all edge-cuts of the graph. We denote it by $\mathrm{e}_{graph}(G)$.*

A general partition $\Pi_{E_s}(G) = \{V_1, ..., V_k\}$ is preferable if the expansion of its components $V_i, i = 1, ..., k$, is high, because this indicates a high intra-component connectivity [KVV04, FTT04].
Otherwise, a partition can be considered to be good if either the average expansion or the maximum expansion, defined by

$$1/k \sum_{1 \leq i \leq k} \mathrm{e}^b_{ES}(V_i, V \setminus V_i) \qquad \text{and} \qquad \max_{1 \leq i \leq k} \mathrm{e}^b_{ES}(V_i, V \setminus V_i), \tag{4.16}$$

is low, because this indicates relatively low cut-sizes [KC12].
Enhancing the definition of expansion, the conductance of a graph gives greater importance to vertices with higher degree [KVV04]:

Definition 4.5.9. *The **conductance of a bipartition** $\Pi_{E_s^b}(G) = \{V_1, V_2\}$ induced by an edge-cut E_s^b is defined as*

$$\mathrm{c}^b_{ES}(\Pi_{E_s^b}) = \frac{|E_s^b|}{\min\{\sum_{v_1 \in V_1} d_{v_1}, \sum_{v_2 \in V_2} d_{v_2}\}} = \frac{|E(V_1, V_2)|}{\min\{\sum_{v_1 \in V_1} d_{v_1}, \sum_{v_2 \in V_2} d_{v_2}\}}. \tag{4.17}$$

*The **conductance of a graph** G is the minimum conductance over all edge-cuts of the graph. We denote it by $\mathrm{c}_{graph}(G)$.*

As in case of expansion, a partition $\Pi_{E_s}(G) = \{V_1, ..., V_k\}$ is preferable if the conductance of its components $V_i, i = 1, ..., k$, is high. However, both, expansion and conductance, are insufficient by themselves as quality criteria because they take neither inter-component connectivity nor the relative size of components into account. To overcome this drawback, [KVV04, FTT04] propose different partitioning methods that combine the above criteria with the cut-size criterion capturing inter-component connectivity.
The following two adaptions of the expansion measure to partitions induced by vertex-separators are proposed by [KC12].

Definition 4.5.10. *The **expansion** of a bipartition $\Pi_{V_s^b}(G) = \{V_1, V_2; V_s^b\}$ induced by a vertex-cut V_s^b can be measured by*

$$\mathrm{e1}_{VS}(\Pi_{V_s^b}) = \frac{|V_s^b|}{\min\{|V_1| + |V_{s,1}^b|, |V_2| + |V_{s,2}^b|\}}, \quad \textit{or} \quad \mathrm{e2}_{VS}(\Pi_{V_s}) = \frac{|V_s^b|}{\min\{|E(V_1)|, |E(V_2)|\}}, \tag{4.18}$$

where $E(V_i) = \{e \in E \mid e \cap V_i \neq \emptyset\}$.

Note that for a narrow vertex-separator, we have $|V_s^b| = |V_{s,1}^b| = |V_{s,2}^b|$ by lemma 4.5.4.

b. Typical Problem Formulations

There are many different formulations of partitioning problems of which we present the most common ones in a general form. We consider, as usual, a connected undirected graph $G = (V, E)$ with $N_V := |N|$ and $N_E := |E|$. Since bipartitioning, also called bisection, is the basis for many general partitioning algorithms, we start with the respective formulations of edge-cut based bipartitioning and vertex-cut based bipartitioning.

Basic edge-cut based bipartitioning [BJ92]: Let $\beta(N_V) \leq N_V$ be a positive integer. The balanced edge-separator problem is to find an edge-separator E_s of G inducing a bipartition $\Pi_{E_s} = \{V_1, V_2\}$ such that
i) there is no edge between V_1 and V_2,
ii) $\max\{|V_1|, |V_2|\} \leq \beta(N_V)$, and
iii) $|E_s|$ is minimized subject to i) and ii).

Basic vertex-cut based bipartitioning [BJ92, dSB05, BM11]: Let $\beta(N_V)$ be a positive integer. The balanced vertex-separator problem is to find a vertex-separator V_s of G inducing a bipartition $\Pi_{V_s} = \{V_1, V_2; V_s\}$ such that
i) there is no edge between V_1 and V_2,
ii) $\max\{|V_1|, |V_2|\} \leq \beta(N_V)$, and
iii) $|V_s|$ is minimized subject to i) and ii).

Both of these problems are $\mathcal{NP}$-hard [BJ92]. Instead of $|E_s|$ and $|V_s|$, in iii) one could also use another cut-size definition, cf. definition 4.5.5, and [KL70, dSB05]. A further kind of bipartitioning is to approximate a solution of the following problem, where a balance criterion is integrated in the objective function:

Minimum ratio vertex-cuts [FHL08]: Let G be a graph with positive vertex weights $w(v)$, $\forall\ v \in V$. The minimum ratio cut bipartitioning problem is to find a vertex-separator V_s of G inducing a partition $\Pi_{V_s} = \{V_1, V_2; V_s\}$ such that
i) there is no edge between V_1 and V_2, and
ii) the *sparsity* of the separator[6], defined by

$$\frac{\sum_{v_s \in V_s} w(v_s)}{\min_{i \in \{1,2\}} \{\sum_{v_i \in V_i} w(v_i)\} + \sum_{v_s \in V_s} w(v_s)}, \tag{4.19}$$

is minimized subject to i).

The following problem formulations of GPES and GPVS generalize the first two definitions above:

Edge-cut based k-way partitioning [KL70, KPcA12]: Let $k \leq N_V$ be a positive integer. The GPES problem is to find an edge-separator E_s of G inducing a k-way partition $\Pi_{E_s} = \{V_1, ..., V_k\}$ such that
i) there is no edge between V_i and V_j $\forall\ i \neq j \in \{1, ..., k\}$,
ii) Π_{E_s} fulfills a certain balance criterion, and
iii) the cut-size of Π_{E_s} is minimized subject to i) and ii).

Vertex-cut based k-way partitioning [KPcA12]: Let $k \leq N_V$ be a positive integer. The GPVS problem is to find a vertex-separator V_s of G inducing a k-way partition $\Pi_{V_s} = \{V_1, ..., V_k; V_s\}$ such that
i) there is no edge between V_i and V_j $\forall\ i \neq j \in \{1, ..., k\}$,
ii) Π_{V_s} fulfills a certain balance criterion, and
iii) the cut-size of Π_{V_s} is minimized subject to i) and ii).

Possible *balance criteria* in ii) are:

- *Balanced component weights* [KPcA12]: Let G be a graph with positive vertex weights $w(v)$, $\forall\ v \in V$, and let be given a maximum allowable imbalance ratio $\epsilon > 0$. A partition Π_{E_s}, or Π_{V_s}, is constrained to fulfill

$$\max_{1 \leq i \leq k} \left\{ \sum_{v_i \in V_i} w(v_i) \right\} \leq (1 + \epsilon) \frac{\sum_{i=1}^{k} \sum_{v_i \in V_i} w(v_i)}{k}. \tag{4.20}$$

- *Lower bounded expansion of components* [KVV04]: Given a minimum allowable expansion $\alpha > 0$, a partition Π_{E_s}, or Π_{V_s}, is constrained to fulfill

$$\min_{1 \leq i \leq k} \mathrm{e}_{graph}(G(V_i, E_i)) \geq \alpha. \tag{4.21}$$

[6] Note the similarity of this definition to the expansion $\mathrm{e1}_{VS}$ in definition 4.5.10.

- *Lower bounded conductance of components* [KVV04]: Given a minimum allowable expansion $\alpha > 0$, a partition Π_{E_s}, or Π_{V_s}, is constrained to fulfill

$$\min_{1 \leq i \leq k} \mathrm{c}_{graph}(G(V_i, E_i)) \geq \alpha. \tag{4.22}$$

Partitions of maximum modularity [New04, DDGDA05, BDG+08, New06]: The problem is to maximize modularity over all possible partitions of a graph G by an edge-separator:

$$\max_{E_s \subseteq E} \mathrm{m}_{ES}(\Pi_{E_s}). \tag{4.23}$$

Graph fragmentation by removing m "key players" [Bor06], cf. section 4.6: Let $m < N_V$ be a positive integer. The key player problem is to find a set $V_s \subset V$ with $|V_s| = m$ whose removal from G results in a graph that maximizes some graph fragmentation measure F:

$$\max_{V_s \subset V} F(G - V_s). \tag{4.24}$$

c. Graph Partitioning by Edge-Separator (GPES)

Many graph partitioning methods based on the detection of small edge-cuts have been developed in the last decades. [Kol09, New10, BE05] sketch several of them. Most of the common methods fall into one of the following basic groups.

Hierarchical partitioning

Hierarchical partitioning is mostly based on the definition of (dis)similarity measures between sets of vertices. It results in an entire hierarchy of nested partitions between the trivial partitions $\{\{v_1\}, \{v_2\}, ..., \{v_{N_V}\}\}$ and V. Two opposed concepts can be distinguished:
Agglomerative algorithms start with the trivial partition $\{\{v_1\}, \{v_2\}, ..., \{v_{N_V}\}\}$ and accomplish coarsening through successive merging of partitions. For example, [New04] proposes a well performing method based on a greedy algorithm to maximize modularity. Its worst-case running time is $O((N_E + N_V)N_V)$, or $O(N_V^2)$ on a sparse graph.
On the contrary, *divisive algorithms* start with the trivial partition V and proceed by successive refinement of partitions through splitting. There are many examples of divisive methods: Recursive bipartitioning can be performed, e.g., using the famous Kernighan-Lin algorithm [KL70] for approximations of minimum edge-cuts. It is a simple but rather slow heuristic algorithm for graph bisection. Integrating some improvements, this algorithm has an overall running time of $O(N_V^3)$. Faster bisection can be achieved, for instance, by applying spectral partitioning with a running time of $O(\sqrt{N_V} N_E)$, cf. [PSL90], or by the $O(\sqrt{\log(N_V)})$-approximation algorithm for a sparsest cut of [ARV04] and their related approximation of a balanced edge-separator. Two newer common approaches are a method based on minimum cut and expansion criteria introduced by [FTT04], as well

as an algorithm proposed by Girvan and Newman [GN02, GN04]: It iteratively removes edges of a graph based on their betweenness centrality, which is defined similar to the betweenness centrality of vertices in definition 4.3.4, and runs in $O(N_E^2 N_V)$ worst case time, and in $O(N_V^3)$ time on sparse graphs.

Spectral partitioning

The following methods for graph partitioning exploit that the connectivity of a graph is associated with the eigendecomposition of matrices as its adjacency matrix $\boldsymbol{A}$ or its Laplacian $\boldsymbol{L}$, defined below, cf. for example [KVV04, vL07, MC09].

The first approach is to conduct a *spectral analysis of the adjacency matrix* $\boldsymbol{A}$ of a graph that results in N_V pairs $(\lambda_i, \boldsymbol{x}_i) \in \mathbb{R} \times \mathbb{R}^{N_V}$ of eigenvalues and eigenvectors, for which $\boldsymbol{A}\boldsymbol{x}_i = \lambda_i \boldsymbol{x}_i$, and which are ordered by their eigenvalues $\lambda_1 \leq ... \leq \lambda_{N_V}$. Then, starting with eigenvectors $\boldsymbol{x}_i$ corresponding to large (absolute) eigenvalues λ_i, certain vertices $v_j \in V$, and their immediate neighbors, are declared to build a cluster V_i if they have in common particularly large positive or negative entries x_{ij} in $\boldsymbol{x}_i$. Some fundamental properties of the graph spectrum $\lambda_1, ..., \lambda_{N_V}$ are pointed out in [BE05]. Results and a discussion of this approach can be found in [GMZ03].

The much more popular approach is to perform iterative bisection of a graph according to *spectral properties of the graph Laplacian* $\boldsymbol{L}$ defined by $\boldsymbol{L} := \boldsymbol{D} - \boldsymbol{A}$, where $\boldsymbol{D} = \mathrm{diag}[(d_{v_1}, ..., d_{v_{N_V}})]$ is the diagonal matrix of vertex degrees. If G is a simple undirected graph, the entries l_{ij} of $\boldsymbol{L}$ are given by

$$l_{ij} = \begin{cases} -1 & \text{if } \{v_i, v_j\} \in E, \\ d_{v_i} & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases} \tag{4.25}$$

One can easily show that G consists of k connected components if and only if the k smallest eigenvalues of $\boldsymbol{L}$ are zero, $\lambda_1(\boldsymbol{L}) = ... = \lambda_k(\boldsymbol{L}) = 0$, and $\lambda_{k+1}(\boldsymbol{L}) > 0$, cf. e.g. [BE05]. The smallest eigenvalue $\lambda_1(\boldsymbol{L})$ is identically equal to zero with corresponding eigenvector $\boldsymbol{x}_1(\boldsymbol{L}) = (1, ..., 1)^{\mathrm{T}}$. A graph G of 'nearly' two components, i.e., suitable for bisection, can be assumed to have $\lambda_2(\boldsymbol{L})$ close to zero. A common and successful approach to graph bisection is to partition vertices according to the sign of their entries in the corresponding eigenvector $\boldsymbol{x}_2(\boldsymbol{L})$ by defining

$$V_1 := \{v_i \in V \mid x_{2i}(\boldsymbol{L}) \geq 0\} \qquad \text{and} \qquad V_2 := \{v_i \in V \mid x_{2i}(\boldsymbol{L}) < 0\}. \tag{4.26}$$

This method traces back to Fiedler [Fie73], and is further studied and advanced by [PSL90, NJW02, CSB+11].

An alternative approach is to study the spectrum of the normalized Laplacian $\boldsymbol{D}^{-1}\boldsymbol{L}$. A recent tutorial about spectral partitioning including a review of algorithms and many references is given by [vL07]. The time complexity of spectral partitioning is mainly due to the calculation of eigenvalues and their corresponding eigenvectors. Some of the largest or smallest eigenvalues of a large, sparse, and symmetric matrix can be efficiently

calculated by variants of the iterative Lanczos algorithm [GvL96]. Its convergence speed depends on the size of the spectral gap of the matrix. In the case of spectral bisection, for example, its time complexity is $O(N_E/(\lambda_3 - \lambda_2))$.

d. Graph Partitioning by Vertex-Separator (GPVS)

GPVS is much less common and studied than GPES, although these problems are closely related. Therefore, an indirect approach to GPVS is to compute an edge-separator and to translate it to a vertex-separator: A narrow vertex-separator can be found by solving a minimum vertex cover problem on the bipartite graph induced by the edge-cut and adjacent vertices [PSL90]. If this method is successful and efficient, depends on the structure of the graph. Important aspects in which edge- and vertex-separators differ are discussed in [FHL08, KC12].
Direct methods to detect favored vertex-separators are proposed by [dSB05, FHL08, BM11, KC12]. [dSB05] provides a mixed integer programming formulation of the vertex-cut based bipartitioning problem, and proposes different version of a branch-and-cut algorithm to approximate solutions. The same problem is studied by [BM11] from a polyhedral point of view, which enables to solve all the instances generated by [dSB05] to optimality in small time without using any sophisticated MIP algorithm. Adapting [ARV04]'s techniques developed in the context of edge-separators, [FHL08] obtains an $O(\sqrt{\log(N_V)})$ approximation for minimum ratio vertex-cuts, and exhibits a pseudo-approximation for finding balanced vertex-separators in general graphs with running time of the same order. This improved the previous best approximation ratio for vertex-separators of $O(\log(N_V))$ based on [LR99]. [KC12] presents a vertex-cut based partitioning algorithm based on structural balance vertices, called SBV-cut, which performs well, both in terms of the modularity measures and in terms of expansion.
Section 4.6 presents the problem of graph fragmentation by removing key players [Bor06], which is in a sense related to GPVS.

e. Hypergraph Partitioning

Algorithms on hypergraphs are in general much more expensive in terms of computational complexity then algorithms on graphs. Therefore, research on hypergraphs often attempts to answer certain questions by solving problems on suitably related graphs.
[KPcA12] presents an effective method that solves the problem of hypergraph partitioning by finding vertex-separators on an undirected graph, namely by solving the GPVS problem on the net intersection graph (NIG) of a hypergraph. A vertex-separator on this graph defines a net-separator for the hypergraph. Since this method exactly addresses our problem of determining advantageous decompositions of a block-separable optimization problem with coupling constraints of section 5.2, this section is mainly a sketch of [KPcA12]. For an illustrating visualization of the method, we refer to this publication.

Furthermore, at the end of this section, we give a definition of a vertex-separator on a hypergraph. To our knowledge, there are no publications about hypergraph partitioning by vertex-separators, but we show that such vertex-separators are identical to vertex-separators on the induced graph of a hypergraph by our definition.

Definition 4.5.11. *A net subset $N_s \subseteq N$ of a hypergraph $H = (U, N)$ is called a* ***k-way net-separator****, or a* ***k-way net-cut****, if its removal disconnects the hypergraph into at least k connected components. The induced k-way partition of the vertices is denoted by $\Pi_{N_s}(H) = \{U_1, ..., U_k\}$. Simultaneously, the net-separator N_s induces a partition of the net set $\widetilde{\Pi}_{N_s}(H) = \{N_1, ..., N_k; N_s\}$, where N_i denotes the set of internal nets of a vertex set U_i, $i = 1, ..., k$.*

Theorem 4.5.12. *A vertex partition $\Pi_{V_s}(G_{NIG}) = \{V_1, ..., V_k; V_s\}$ of the net intersection graph G_{NIG} of a hypergraph $H = (V, N)$ by a narrow vertex-separator $V_s \subset V_{NIG}$ induces a net partition $\widetilde{\Pi}_{N_s}(H) = \{N_1 \equiv V_1, ..., N_k \equiv V_k; N_s \equiv V_s\}$ of H by a net-separator N_s, which fulfills that $\text{Pins}(N_i) \cap \text{Pins}(N_j) = \emptyset \ \forall \ i \neq j \in \{1, ..., k\}$.*

Proof By definition of a vertex-separator $V_s \subset V_{NIG}$ it does not exist any edge between any two vertices $v_i \in V_i$ and $v_j \in V_j$ for $i \neq j \in \{1, ..., k\}$, which implies that $\text{Pins}(N_i) \cap \text{Pins}(N_j) = \emptyset \ \forall \ i \neq j \in \{1, ..., k\}$: If we had $\text{Pins}(n_i) \cap \text{Pins}(n_j) \neq \emptyset$ for two nets $n_i \in N_i$ and $n_j \in N_j$, then there would be an edge e_{ij} between vertices $v_i \in V_i$ and $v_j \in V_j$ which would contradict that $V_s \subset V_{NIG}$ is a vertex-separator of G_{NIG}.

However, this net partition $\widetilde{\Pi}_{N_s}(H)$ induces only a partial vertex partition:

Corollary 4.5.13. *A net partition of H induced by a net-separator N_s*

$$\widetilde{\Pi}_{N_s}(H) = \{N_1 \equiv V_1, ..., N_k \equiv V_k; N_s \equiv V_s\} \tag{4.27}$$

induces a k-way partial node partition of H

$$\Pi_{N_s}^{partial}(H) = \{U_1^{partial} = \text{Pins}(N_1), ..., U_k^{partial} = \text{Pins}(N_k)\}. \tag{4.28}$$

The remaining nodes are those which are only connected by nets of the separator N_s. In the following, they are called *free nodes*, and U_F denotes their assemblage in the set

$$U_F = U \ \setminus \ \bigcup_{i=1}^{k} U_i^{partial} = \{u_i \in U \mid u_i \in n, n \in N \Rightarrow n \in N_s\}.$$

A complete node partition can be constructed by assigning each of the free nodes to a set $U_i^{partial}, i \in \{1, ..., k\}$:

$$\Pi_{N_s}(H) = \{U_1 \supseteq U_1^{partial}, ..., U_k \supseteq U_k^{partial}\}. \tag{4.29}$$

Let $\lambda(n_s)$ denote the *connectivity* of a net $n_s \in N_s$, which is defined as the number of parts $U_i, i \in \{1, ..., k\}$, of $\Pi_{N_s}(H)$ that are connected by n_s.

Theorem 4.5.14. *Let V_s be a narrow vertex-separator that induces a vertex partition $\Pi_{V_s}(G_{NIG})$. Then any node partition $\Pi_{N_s}(H) = \{U_1, ..., U_k\}$ constructed according to (4.29) induces the net partition $\widetilde{\Pi}_{N_s}(H)$ given by (4.27) where the connectivity of each cut-net $n_s \in N_s$ is greater than or equal to the connectivity of the corresponding vertex $v_s \in V_s$: $\lambda(n_s) \geq \lambda(v_s)$.*

Proof See [KPcA12].

Note that $\lambda(n_s)$ may be greater than $\lambda(v_s)$ only due to the assignment of the free nodes. Defining the following cut-size metrics, theorem 4.5.14 entails corollary 4.5.16.

Definition 4.5.15. *Let $c(n) \geq 0$ be some predetermined cost of a net $n \in N$. Then, the **cut-size** of a node partition $\Pi_{N_s}(H) = \{U_1, ..., U_k\}$ induced by a net-separator N_s (and a certain assignment of free nodes) can be defined, for example, by*
*a) the **cut-net metric***

$$\textit{cut-size}_a(\Pi_{N_s}) = \sum_{n_s \in N_s} c(n_s) \qquad \textit{or} \tag{4.30}$$

*b) the **connectivity metric***

$$\textit{cut-size}_b(\Pi_{N_s}) = \sum_{n_s \in N_s} c(n_s)(\lambda(n_s) - 1). \tag{4.31}$$

One can observe that the free node assignment has no influence on the cut-net metric, however, it affects the connectivity metric. Defining analogously to definition 4.5.15 the cut-size of a vertex partition $\Pi_{V_s}(G_{NIG}) = \{V_1, ..., V_k; V_s\}$, we get

Corollary 4.5.16. *The cut-size of a vertex partition $\Pi_{V_s}(G_{NIG})$ is equal to the cut-size of any induced node partition $\Pi_{N_s}(H)$ according to the cut-net metric: $\textit{cut-size}_a(\Pi_{V_s}) = \textit{cut-size}_a(\Pi_{N_s})$. According to the connectivity metric, the cut-size of the vertex partition $\Pi_{V_s}(G_{NIG})$ only approximates the cut-size of a node partition $\Pi_{N_s}(H)$: $\textit{cut-size}_b(\Pi_{V_s}) \leq \textit{cut-size}_b(\Pi_{N_s})$.*

However, it is easy to see that in the case of bipartitioning by a narrow separator, equality of cut-sizes holds no matter which metric is used, since $\forall\ v_s \equiv n_s \in V_s \equiv N_s$: $\lambda(v_s) = \lambda(n_s) = 2$:

Corollary 4.5.17. *The cut-size of a vertex bipartition $\Pi_{V_s}(G_{NIG}) = \{V_1, V_2; V_s\}$ by a narrow separator V_s is equal to the cut-size of any induced node partition $\Pi_{N_s}(H) = \{U_1, U_2\}$ according to both cut-net metrics:*
$\textit{cut-size}_a(\Pi_{V_s}) = \textit{cut-size}_a(\Pi_{N_s}) = \textit{cut-size}_b(\Pi_{V_s}) = \textit{cut-size}_b(\Pi_{N_s})$.

This result forms the basis for hypergraph partitioning by recursive bipartitioning of the NIG according to the connectivity and the cut-net metrics as proposed by [KPcA12].

Finally, we define vertex-separators of a hypergraph H. By this definition, they are identical to vertex-separators on the induced graph G_{ind} of the hypergraph.

Definition 4.5.18. *A subset $U_s \subset U$ of a hypergraph $H = (U, N)$ is called a **k-way vertex-separator**, or a **k-way vertex-cut**, if its removal disconnects the hypergraph into at least k connected components, where the net set of the hypergraph $H - U_s$ is defined as $N_{U_s} := \{n \setminus U_s \mid n \in N, |n \setminus U_s| \geq 2\}$, i.e., N_{U_s} basically contains all nets of N that connect at least two vertices of $U \setminus U_s$, but each such net without the potentially included elements of U_s. The induced k-way partition of the vertices is denoted by $\Pi_{U_s}(H) = \{U_1, ..., U_k; U_s\}$.*

Theorem 4.5.19. *A subset $U_s \subset U$ is a k-way vertex-separator of a hypergraph $H = (U, N)$ if and only if U_s is a k-way vertex-separator of H's induced graph $G_{ind} = (U, E)$.*

Proof a) Let U_s be a vertex-separator of H inducing a partition $\{U_1, ..., U_k; U_s\}$, i.e., $\nexists\, n \in N_{U_s} : n \cap U_i \neq \emptyset, n \cap U_j \neq \emptyset$ for $i \neq j \in \{1, ..., k\}$. Assuming that $G_{ind} - U_s$ does not decompose into k components with nodes $U_1, ..., U_k$, there exists an edge $e = \{u_i, u_j\} \in E, u_i \in U_i, u_j \in U_j$, for $i \neq j \in \{1, ..., k\}$. But this means that H, and, therefore, $H - U_s$, must include a net n with $n \cap U_i \neq \emptyset$ and $n \cap U_j \neq \emptyset$, which contradicts that U_s is a k-way vertex-separator of H.
b) Let U_s be a vertex-separator of G_{ind} inducing a partition $\{U_1, ..., U_k; U_s\}$, i.e., $\nexists\, e = \{u_i, u_j\} \in E, u_i \in U_i, u_j \in U_j$, for $i \neq j \in \{1, ..., k\}$. Assuming that $H - U_s$ does not decompose into k components with nodes $U_1, ..., U_k$, there exists a net $n \in N_{U_s}$: $n \cap U_i \neq \emptyset, n \cap U_j \neq \emptyset$ for $i \neq j \in \{1, ..., k\}$. However, this means that G_{ind}, and, therefore, $G_{ind} - U_s$, includes an edge $e = \{u_i, u_j\}$, which contradicts that U_s is a k-way vertex-separator of G_{ind}.

4.6 Measuring Graph Fragmentation

This section gives an overview of possibilities to measure the fragmentation of a network. Such measures are basic for the task of disrupting or fragmenting a network by removing its key nodes. The fragmentation problem is quite similar to graph partitioning by vertex-separators, however, it differs in terms of the related quality measures. In fact, one aims to identify a small set of vertices whose removal decomposes the graph into as many as possible components of low intra-component connectivity. This is in contrast to GPVS, which aims for high intra-component connectivity. [Bor06] defines the *problem of identifying the m key players* as follows:
Given a network represented by an undirected graph $G = (V, E)$, find a set $V_s \subset V$ of m nodes such that removing this set would result in a residual network with the least possible cohesion, which means with the highest possible fragmentation.

The solution of this problem obviously depends on the underlying measure. [Bor06] proposes the following fragmentation measures for a graph $G = (V, E)$ consisting of k components $(V_i, E_i), i = 1, ..., k$. Besides number and size of components, the last of these measures, F^D, captures also intra-component connectivity.

- F^{comp} counts the *number k of network components*, normalized by dividing by the number of nodes
$$F^{comp} = \frac{k}{|V|}. \tag{4.32}$$

- F counts the *number of pairs of nodes $v_i \neq v_j \in V$ that are disconnected from each other*. With $r_{ij} = 1$ if v_i can reach v_j and $r_{ij} = 0$ otherwise, F is defined by
$$F = 1 - \frac{2\sum_{j<i\leq|V|} r_{ij}}{|V|(|V|-1)}. \tag{4.33}$$
This expression is relatively expensive to compute, but it can be rewritten in terms of size $|V_i|$ of each component
$$F = 1 - \frac{\sum_{1\leq i\leq k} |V_i|(|V_i|-1)}{|V|(|V|-1)}. \tag{4.34}$$
Remark 4.6.1. Note that the quantity $\sum_{j<i\leq|V|} r_{ij} = \sum_{1\leq i\leq k} \frac{|V_i|(|V_i|-1)}{2}$ is known as *total pairwise connectivity* of a graph. The minimization of this quantity under certain side conditions is the objective of problems as detecting critical nodes of a sparse network or assessing network vulnerability. The resulting optimization problems are $\mathcal{NP}$-complete but can be solved approximately by efficient heuristics [ACEP09, DXT+10].

- F is similar to the *diversity measure H, the heterogeneity* (also known as concentration ratio or Hirschman-Herfindahl index), which is in this context defined by
$$H = 1 - \sum_{1\leq i\leq k} \Big(\frac{|V_i|}{|V|}\Big)^2. \tag{4.35}$$
H can maximally achieve the value $1 - (1/|V|)$ (if all nodes are isolated). Normalizing H, we obtain the measure F:
$$\begin{aligned} H^* &= \frac{1 - \sum_{1\leq i\leq k}(\frac{|V_i|}{|V|})^2}{1 - |V|^{-1}} = \frac{|V| - \sum_{1\leq i\leq k} \frac{|V_i|^2}{|V|}}{|V| - 1} = 1 - \frac{-|V| + \sum_{1\leq i\leq k} |V_i|^2}{|V|(|V|-1)} \\ &= 1 - \frac{\sum_{1\leq i\leq k} |V_i|(|V_i|-1)}{|V|(|V|-1)} = F \end{aligned} \tag{4.36}$$

- The *information entropy* E is in the context of network fragmentation defined as

$$E = -\sum_{1 \leq i \leq k} \frac{|V_i|}{|V|} \ln\Big(\frac{|V_i|}{|V|}\Big). \tag{4.37}$$

  Normalizing E yields

$$E^* = \frac{\sum_{1 \leq i \leq k} \frac{|V_i|}{|V|} \ln(\frac{|V_i|}{|V|})}{\ln(\frac{1}{|V|})}. \tag{4.38}$$

- A fragmentation measure that *captures not only the number and size of components* in which a graph breaks down *but also the relative cohesion of each component* is

$$F^D = 1 - \frac{2\sum_{1 \leq j < i \leq |V|} \frac{1}{\mathrm{dist}(v_i, v_j)}}{|V|(|V|-1)}. \tag{4.39}$$

  F^D is equal to F if each component is complete, i.e., if there is an edge between each two vertices of a component.

[Bor06] handles the above defined key player problem via combinatorial optimization by a greedy algorithm. Alternative algorithms to solve the combinatorial problem are tabu-search, the Kernighan-Lin algorithm, and simulated annealing. A different heuristic approach would be to choose m vertices that maximize some centrality measure, cf. section 4.3, and to seek modifications of this set that address some of its weak points. The most appropriate centrality measure for this approach would obviously be the betweenness centrality.

5 A Graph-Theoretical Approach to Decomposing Constrained Nonlinear Programs

Large-scale optimization problems arise in many application areas, especially in operations research. Examples are, among many others, optimal distribution of natural gas [RMWSB02, MHH+04, BSH11], optimization of water distribution systems [BGS09, ZSZ13, ZSZD13], utility maximization in communication networks [PC06, CLCD07], optimal flow in multi-commodity networks [OMV00, Min01, GK07], as well as diverse problems in chemical engineering [FW08, ZLB08, Bie10]. Some of these problems are solved by applying reduction methods that also ease problem analysis and the interpretation of optimal solutions. However, most reduction methods are problem-specific and cannot readily be applied to more general nonlinear programs.
Moreover, there are decomposition methods, which are used for the numerical solution of large-scale programs with a general block-separable structure, cf. section 3.5. They are quite popular because a large number of optimization problems shows a natural block-separable structure that results from the components of the underlying model. For example, exploiting this structure proved to yield efficient algorithms used for solving large-scale nonlinear programs resulting from optimal control problems [BP84, LBBS03] and from large-scale parameter estimation problems [BKS07, ZLB08]. If a MINLP is not block-separable, or if it has some large blocks that should be subdivided, splitting-schemes often succeed in transforming it into a suitable block-separable MINLP by introducing proper additional variables and constraints [Now05]. A further benefit of decomposition methods lies in their intuitively accessible way of proceeding: Primal and dual decomposition of problems with coupling constraints can illustratively be interpreted as resource and price decomposition, respectively. However, decomposition methods work on a predetermined problem structure, which is not necessarily useful for model analysis and possibly does not even lead to efficient numerical optimization. [Now05] traces this inefficiency back to the size of blocks, and suggests balancing the size and number of subproblems, instead of solving either one large problem or many small ones. His suggestion is due to the fact that, to obtain quickly solvable subproblems, each of which includes only few variables, Lagrangian relaxation must be applied to many constraints, which leads, in case of nonconvex optimization, to a larger duality gap. Both goals, fast computation and good approximation, should be counter-balanced. Balanced size and number of subproblems are also favorable to problem analyses. In addition, the number of variables connecting submodels should be as small as possible. As mentioned at the end of section 3.5, [CLCD07] emphasizes the fact that different representations of one optimization problem, with a (block-)separable structure, lead to different decom-

position schemes, which are difficult to compare and rate before numerical optimization is accomplished. Furthermore, a systematical exploration of possible decompositions is an open issue.
In this chapter, we present a new approach to expose the overall structure of constrained optimization problems and to explore and compare different decompositions. We delineate a graph that captures the connectivity of primal and dual model variables. This graph can be analyzed and beneficially partitioned using appropriate graph-theoretical methods. Resulting graph decompositions are transferred to the original problem, leading to different decompositions of the optimization problem. The idea of visualizing the structure of an optimization problem with complicating variables or constraints on a graph is in a similar way suggested by [BXMM07]. However, the authors do not propose to apply graph-theoretical methods to analyze the problem and reveal advantageous decompositions.
Note that the objective of our approach does not merely lie in an efficient numerical solution of a problem but rather in a convenient problem formulation that eases analysis and allows tailored model reduction. More specifically, we aim for

- determining weakly connected model components with only few variables describing the connection between subproblems and the master problem,
- balancing the size and number of subproblems,
- approximating the optimal value functions of subproblems within the relevant domains.

The resulting decomposition of the overall problem into subproblems with fewer variables and constraints facilitates

- the solution of subproblems independently and in parallel,
- (nearly) independent sensitivity analyses within subproblems,
- global analyses of subproblems, instead of only local analyses of the overall problem, in order to classify parameters into more and less important ones, which reduces the effort of data procurement in a problem with a high number of parameters,
- a reduced model formulation: assuming fixed parameters of certain submodels, submodel approximations can be used to solve the overall problem in order to reduce computation times.

Section 5.1 shows how the connectivity of variables in a constrained optimization problem that has a block-separable structure can be represented on a graph. In section 5.2, we propose a method for revealing possible decompositions of a block-separable program and for determining particularly favorable ones using graph theory. This approach is

based on the fact that vertex-cuts of a certain hypergraph correspond to different primal decompositions and that net-cuts of this hypergraph imply different dual decompositions. The subproblems of a decomposition are parametric problems in the links to the master problem. In section 5.3, we discuss the approximation of their optimal value functions, which capture the dependence of a subproblem's optimal objective value on the links.

5.1 Representing the Structures of Block-separable Programs on Graphs

In section 3.5, problem formulation (3.86) reveals that a problem with coupling variables can be transformed into one with coupling constraints. Therefore, we consider in the following only the block-separable optimization problems with the coupling constraints of definition 3.5.1:

$$\min_{\boldsymbol{x}_1,...,\boldsymbol{x}_N} \sum_{i=1}^{N} f_i(\boldsymbol{x}_i) \qquad \text{such that} \qquad \sum_{i=1}^{N} \boldsymbol{g}_i(\boldsymbol{x}_i) \leq \boldsymbol{0}, \qquad \sum_{i=1}^{N} \boldsymbol{h}_i(\boldsymbol{x}_i) = \boldsymbol{0}, \tag{5.1}$$

and $\boldsymbol{x}_i \in \mathcal{F}_i \subseteq \mathbb{R}^{n_i}$, where $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$, $\boldsymbol{g}_i : \mathbb{R}^{n_i} \to \mathbb{R}^m$, $\boldsymbol{h}_i : \mathbb{R}^{n_i} \to \mathbb{R}^p$, $\forall\, i = 1, ..., N$. Here, $\mathcal{F}_i$ represents the local feasible set of variable $\boldsymbol{x}_i$, $\forall\, i = 1, ..., N$.

The Lagrangian dual function of (5.1) decomposes into N separate subproblems, each of which is in one variable $\boldsymbol{x}_i$:

$$d(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \inf_{\boldsymbol{x}_1,...,\boldsymbol{x}_N} \sum_{i=1}^{N} f_i(\boldsymbol{x}_i) + \boldsymbol{\lambda}^{\mathrm{T}} \boldsymbol{g}_i(\boldsymbol{x}_i) + \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{h}_i(\boldsymbol{x}_i) = \sum_{i=1}^{N} d_i(\boldsymbol{\lambda}, \boldsymbol{\mu}). \tag{5.2}$$

Assumption 5.1.1. We suppose in the following that trivial or redundant constraints are removed from the problem. In a numerical application, this can be accomplished by using presolve methods, cf. remark 2.4.5. Furthermore, we assume that the feasible set of problem (5.1) is nonempty.

The problem formulation (5.1) represents the general case that each of the $m + p$ constraints affects each subproblem i of variable $\boldsymbol{x}_i$, $i \in \{1, ..., N\}$. However, in many applications, most coupling constraints $g_j(\boldsymbol{x}) = \sum_{i=1}^{N} g_{i,j}(\boldsymbol{x}_i)$, $j \in \{1, ..., m\}$, and $h_{j-m}(\boldsymbol{x}) = \sum_{i=1}^{N} h_{i,j-m}(\boldsymbol{x}_i)$, $j \in \{m+1, ..., m+p\}$, affect only few subproblems, say the subset $I_j \subset \{1, ..., N\}$, respectively.[1] This means, actually, there do not exist functions $g_{i,j}(\boldsymbol{x}_i)$ (or $h_{i,j}(\boldsymbol{x}_i)$) for $i \in \{1, ..., N\} \setminus I_j$ or, if existing, they are constant. In the following general context of problem (5.1), we refer to this situation by writing $g_{i,j}(\boldsymbol{x}_i) \equiv 0$ (or $h_{i,j}(\boldsymbol{x}_i) \equiv 0$) $\forall\, i \in \{1, ..., N\} \setminus I_j$.[2] In other words, we consider in the following the

[1] In the following, we use the index set $\{1, ..., m+p\}$ to refer to the union of inequality and equality coupling constraints and to related dual variables.

[2] In general, one ought to write $g_{i,j}(\boldsymbol{x}_i) \equiv k$ for some constant $k \in \mathbb{R}$ if $i \in \{1, ..., N\} \setminus I_j$, instead of $g_{i,j}(\boldsymbol{x}_i) \equiv 0$. However, for simplicity, we assume constant terms to be added to a non-constant function $g_{i,j}(\boldsymbol{x}_i), i \in I_j$.

situation that for most $i \in \{1, ..., N\}$ many components $g_{i,j}$ and $h_{i,j}$ of $\boldsymbol{g}_i$ and of $\boldsymbol{h}_i$, respectively, are identically zero. Thus, regarding (5.2), only few subproblems depend on a certain component of the dual variable $(\boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{R}^{m+p}$.[3]

Under the assumption of a loosely coupled system, and if $m+p$ (= number of coupling constraints = number of dual variables) or N (= number of subproblems) are relatively large numbers, e.g., if these numbers are close to n ($= \sum_{i=1}^{N} n_i$ = number of primal variables), we propose to properly aggregate subsets of subproblems that are more strongly coupled among each other than with the remaining subproblems. To detect such subsets of subproblems, we propose to represent the structure of the overall optimization problem (5.1) by means of certain graphs, and to apply methods for graph partitioning. The aggregation of subproblems can turn coupling constraints of the original problem into local constraints of the newly defined subproblem, which in turn reduces the dimension of the dual function $d(\boldsymbol{\lambda}, \boldsymbol{\mu})$'s domain. The result is a partial dual function $\widetilde{d}(\widetilde{\boldsymbol{\lambda}}, \widetilde{\boldsymbol{\mu}}) = \sum_{i=1}^{\widetilde{N}} \widetilde{d}_i(\widetilde{\boldsymbol{\lambda}}, \widetilde{\boldsymbol{\mu}})$ with $\widetilde{m} \leq m$, $\widetilde{p} \leq p$, and with fewer ($\widetilde{N} < N$) but slightly larger subproblems.[4]
Very suited representations of the overall problem can be obtained by passing through the following steps:

1. *Build a bipartite graph* with first vertex set V_s of subproblems, $|V_s| = N$, and second vertex set V_d of dual variable components, $|V_d| = m + p$. Two vertices $v_i \in V_s$ and $v_j \in V_d$ are connected if the corresponding subproblem i depends on the j-th component of the dual variable $(\boldsymbol{\lambda}, \boldsymbol{\mu})$. For example, in case of $j \leq m$, v_i and v_j are connected if $g_{i,j}(\boldsymbol{x}_i) \not\equiv 0$.
   Subproblem dimensions $n_i, i = 1, ..., N$, can be assigned to the nodes $v_i \in V_s$ as weights $w(v_i) := n_i \ \forall\ i \in \{1, ..., N\}$. The weight of each $v \in V_d$ is set to one.

2. *Merge* those *nodes* of each vertex set V_s and V_d that are connected to exactly the same nodes of the other set. We call the resulting reduced bipartite graph G_b, and its vertex sets V_s^{red} and V_d^{red}, where $|V_s^{\text{red}}| \leq N$ and $|V_d^{\text{red}}| \leq m + p$. This step illustrates a basic structuring of the optimization problem. As we will show in example 5.1.2, the described merging of nodes $v_s \in V_s$ leads to multilevel decompositions.
   The weight of a vertex v in G_b is defined as the added weights of those vertices in V_s or in V_d that are represented by v.

[3] More specifically, let us consider a subproblem i given by

$$\min_{\boldsymbol{x}_i} f_i(\boldsymbol{x}_i) + \boldsymbol{\lambda}^{\mathrm{T}} \boldsymbol{g}_i(\boldsymbol{x}_i) + \boldsymbol{\mu}^{\mathrm{T}} \boldsymbol{h}_i(\boldsymbol{x}_i) = \min_{\boldsymbol{x}_i} f_i(\boldsymbol{x}_i) + \sum_{j=1}^{m} \lambda_j g_{i,j}(\boldsymbol{x}_i) + \sum_{j=m+1}^{m+p} \mu_{j-m} h_{i,j-m}(\boldsymbol{x}_i) = d_i(\boldsymbol{\lambda}, \boldsymbol{\mu}).$$

Since $g_{i,j}(\boldsymbol{x}_i) \equiv 0$ for $i \in \{1, ..., N\} \setminus I_j$, the function d_i's domain is actually of dimension $|\{j \in \{1, ..., m+p\} | i \in I_j\}| < m + p$ if $I_j \subset \{1, ..., N\}$.

[4] Note that the components of $(\widetilde{\boldsymbol{\lambda}}, \widetilde{\boldsymbol{\mu}})$ are identically equal to some components of $(\boldsymbol{\lambda}, \boldsymbol{\mu})$, i.e., they have similar meaning and value in an optimal solution of the overall problem.

3. In most cases, the *bipartite graph* G_b can equally be *represented by a hypergraph* $H = (V_s^{\text{red}}, N_d)$ with node set V_s^{red} of subproblems and net set $N_d \equiv V_d^{\text{red}}$ of coupling constraints, which illustrates the structure of the optimization problem very well.[5] However, since merging nodes can turn coupling constraints to local constraints, it can happen that a node $v_d \in V_d^{\text{red}}$ is only connected to one node in V_s^{red}. Such a node v_d will not be represented in H. In this case, we have $|N_d| < |V_d^{\text{red}}|$.

4. *Build the net intersection graph G_{NIG} and the induced graph G_{ind} of H.* They are *subgraphs of the projections of G_b onto one of its two vertex sets*, which are obtained by defining an edge between each two elements of one vertex set that are connected to the same node of the other vertex set in the bipartite graph. The projections onto the vertex sets are also referred to as induced graphs, cf. definition 4.1.9. We name them G_s and G_d, respectively. We have $G_{NIG} \subseteq G_d$ and $G_{ind} = G_s$.
   The graph G_s of subproblems includes an edge between each two subproblems that are directly coupled by a constraint. The graph G_d of dual variable components includes an edge between each two components that appear in the same subproblem.

5. *Search for a vertex-separator* (subset of nodes, cf. definition 4.4.2) that beneficially decomposes G_{NIG} or G_{ind}. All vertex-separators of these graphs imply partitions of the hypergraph H induced by net- and vertex-separators, respectively, cf. section 4.5.

The following examples illustrate steps 1 to 4. Section 5.2 elaborates step 5.

Example 5.1.2 (Constructing the bipartite graph G_b and the hypergraph H, as well as their induced graphs G_s, G_d, G_{NIG}, and G_{ind}, ex. a)). Let us consider a small problem ($N = 4$) with six coupling constraints and without local constraints, i.e., $\mathcal{F}_i = \mathbb{R}^{n_i}, \forall\ i = 1, ..., 4$:

$$\min_{\boldsymbol{x}_1,...,\boldsymbol{x}_4} \sum_{i=1}^{4} f_i(\boldsymbol{x}_i) \qquad \text{such that} \qquad \sum_{i=1}^{4} \boldsymbol{g}_i(\boldsymbol{x}_i) \leq \boldsymbol{0}, \qquad \sum_{i=1}^{4} \boldsymbol{h}_i(\boldsymbol{x}_i) = \boldsymbol{0}, \tag{5.3}$$

where $n_1 = n_3 = n_4 = 1$, $n_2 = 2$, and $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$, $\boldsymbol{g}_i : \mathbb{R}^{n_i} \to \mathbb{R}^4$, $\boldsymbol{h}_i : \mathbb{R}^{n_i} \to \mathbb{R}^2$, $\forall\ i = 1, ..., 4$. Let the structure of $\boldsymbol{g}_i$ and $\boldsymbol{h}_i, i = 1, ..., 4$, be given by

$$\boldsymbol{g}_1(\boldsymbol{x}_1) = \begin{pmatrix} g_{1,1}(\boldsymbol{x}_1) \\ g_{1,2}(\boldsymbol{x}_1) \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{g}_2(\boldsymbol{x}_2) = \begin{pmatrix} g_{2,1}(\boldsymbol{x}_2) \\ g_{2,2}(\boldsymbol{x}_2) \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{g}_3(\boldsymbol{x}_3) = \begin{pmatrix} 0 \\ 0 \\ g_{3,3}(\boldsymbol{x}_3) \\ g_{3,4}(\boldsymbol{x}_3) \end{pmatrix}, \boldsymbol{g}_4(\boldsymbol{x}_4) = \begin{pmatrix} 0 \\ 0 \\ g_{4,3}(\boldsymbol{x}_4) \\ g_{4,4}(\boldsymbol{x}_4) \end{pmatrix},$$

[5] Here, we introduce two graphical representations of a constrained optimization problem. This is because, as we will discuss later on, it is useful to consider, besides the hypergraph H, also the bipartite graph G_b, in view of exploring the most useful decompositions, cf. remark 5.2.10.

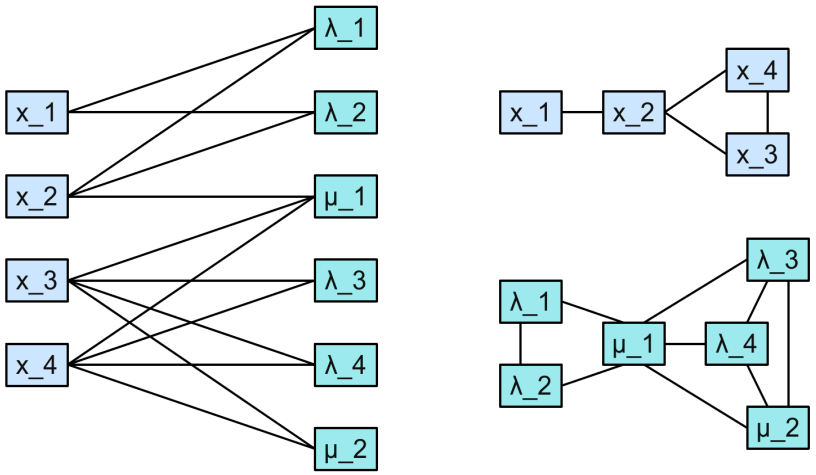

Figure 5.1: *Structure of the optimization problem (ex. 5.1.2)*

$$\boldsymbol{h}_1(\boldsymbol{x}_1) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \boldsymbol{h}_2(\boldsymbol{x}_2) = \begin{pmatrix} h_{2,1}(\boldsymbol{x}_2) \\ 0 \end{pmatrix}, \boldsymbol{h}_3(\boldsymbol{x}_3) = \begin{pmatrix} h_{3,1}(\boldsymbol{x}_3) \\ h_{3,2}(\boldsymbol{x}_3) \end{pmatrix}, \boldsymbol{h}_4(\boldsymbol{x}_4) = \begin{pmatrix} h_{4,1}(\boldsymbol{x}_4) \\ h_{4,2}(\boldsymbol{x}_4) \end{pmatrix}, \tag{5.4}$$

where we assume that the functions $g_{i,j}(\boldsymbol{x}_i)$ and $h_{i,j}(\boldsymbol{x}_i)$ are not identically zero.[6]
The Lagrangian dual function of this problem reads

$$\begin{aligned} d(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \mu_1, \mu_2) = & \inf_{\boldsymbol{x}_1} f_1(\boldsymbol{x}_1) + \lambda_1 g_{1,1}(\boldsymbol{x}_1) + \lambda_2 g_{1,2}(\boldsymbol{x}_1) \\ & + \inf_{\boldsymbol{x}_2} f_2(\boldsymbol{x}_2) + \lambda_1 g_{2,1}(\boldsymbol{x}_2) + \lambda_2 g_{2,2}(\boldsymbol{x}_2) + \mu_1 h_{2,1}(\boldsymbol{x}_2) \\ & + \inf_{\boldsymbol{x}_3} f_3(\boldsymbol{x}_3) + \lambda_3 g_{3,3}(\boldsymbol{x}_3) + \lambda_4 g_{3,4}(\boldsymbol{x}_3) + \mu_1 h_{3,1}(\boldsymbol{x}_3) + \mu_2 h_{3,2}(\boldsymbol{x}_3) \\ & + \inf_{\boldsymbol{x}_4} f_4(\boldsymbol{x}_4) + \lambda_3 g_{4,3}(\boldsymbol{x}_4) + \lambda_4 g_{4,4}(\boldsymbol{x}_4) + \mu_1 h_{4,1}(\boldsymbol{x}_4) + \mu_2 h_{4,2}(\boldsymbol{x}_4) \\ = & \; d_1(\lambda_1, \lambda_2) + d_2(\lambda_1, \lambda_2, \mu_1) + d_3(\lambda_3, \lambda_4, \mu_1, \mu_2) + d_4(\lambda_3, \lambda_4, \mu_1, \mu_2), \end{aligned} \tag{5.5}$$

where $d_i, i = 1, ..., 4$, are the optimal value functions of the relaxed subproblems, which are parametric in the related components of the dual variable $(\boldsymbol{\lambda}, \boldsymbol{\mu})$.
The left hand-side of figure 5.1 shows a bipartite graph corresponding to the structure of this optimization problem with one node for each subproblem and one for each dual variable, i.e., before merging nodes with identical connections. On the right, you see the projections onto the vertex sets of subsystems and dual variables, respectively. Each vertex of these graphs has weight 1, except for x_2 which has $w(\text{x\_2}) = 2$.
Figure 5.2 shows the bipartite graph and the induced graphs after merging nodes that have exactly the same connections within the bipartite graph of figure 5.1. The resulting

[6] We do not assume problem (5.3) to be convex, and do not impose any further conditions on the problem defining functions, because, at the moment, we are only interested in problem structure.

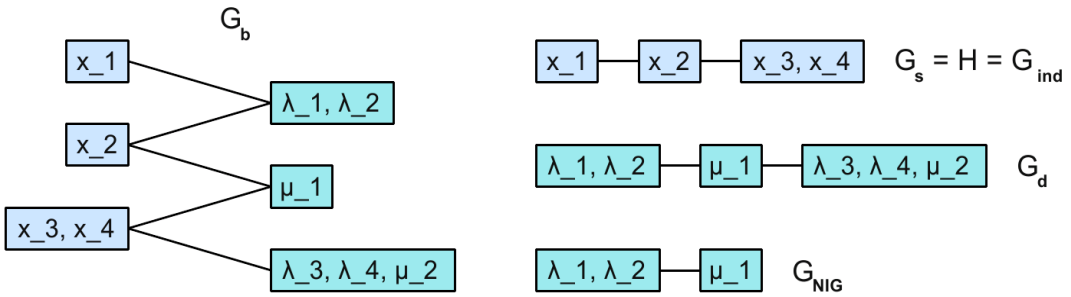

Figure 5.2: *Structure of the optimization problem represented by the bipartite graph G_b, the hypergraph H, and their induced graphs after merging nodes (ex. 5.1.2)*

graphs are called G_b, G_s, and G_d.[7] We have

$$\begin{aligned} V_s^{\text{red}} &= \{\text{x\_1}, \text{x\_2}, [\text{x\_3,x\_4}]\} \quad \text{and} \\ V_d^{\text{red}} &= \{[\lambda\text{\_1}, \lambda\text{\_2}], \mu\text{\_1}, [\lambda\text{\_3}, \lambda\text{\_4}, \mu\text{\_2}]\} \supset \{[\lambda\text{\_1}, \lambda\text{\_2}], \mu\text{\_1}\} = N_d. \end{aligned} \tag{5.6}$$

The weights of the nodes are as follows:
$w(\text{x\_1}) = 1$, $w(\text{x\_2}) = 2$, $w([\text{x\_3,x\_4}]) = 2$, $w([\lambda\text{\_1}, \lambda\text{\_2}]) = 2$, $w(\mu\text{\_1}) = 1$, $w([\lambda\text{\_3}, \lambda\text{\_4}, \mu\text{\_2}]) = 3$.
In this small example, the hypergraph $H = (V_s^{\text{red}}, N_d)$, with node set V_s^{red} of subproblems and net set N_d of coupling constraints, is just a simple graph and identically equal to the graph G_s and also to the induced graph G_{ind} of H, since, in G_b, each vertex of V_d^{red} connects at most two vertices of V_s^{red}. As mentioned above, the net intersection graph G_{NIG} of H is identically equal to G_d or a subgraph of G_d. In this example, it is the subgraph of G_d, given by the nodes $[\lambda\text{\_1}, \lambda\text{\_2}]$ and $\mu\text{\_1}$ and their connecting edge $\{[\lambda\text{\_1}, \lambda\text{\_2}], \mu\text{\_1}\}$. The constraints corresponding to the dual variables λ_3, λ_4, and μ_2 turn to local constraints of the aggregated subproblem in variables $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$.
Merging some nodes $v_s \in V_s$, here x_3 and x_4 into [x_3,x_4], illustrates the first structuring of the optimization problem that leads to a coarser decomposition. The corresponding partial dual function of the overall problem is given by

$$d_{basic}^{part}(\lambda_1, \lambda_2, \mu_1) = d_1(\lambda_1, \lambda_2) + d_2(\lambda_1, \lambda_2, \mu_1) + d_{3,4}(\mu_1), \tag{5.7}$$

where

$$d_{3,4}(\mu_1) = \inf_{\substack{\boldsymbol{x}_3, \boldsymbol{x}_4 \\ g_3(\boldsymbol{x}_3, \boldsymbol{x}_4) \leq 0 \\ g_4(\boldsymbol{x}_3, \boldsymbol{x}_4) \leq 0 \\ h_2(\boldsymbol{x}_3, \boldsymbol{x}_4) = 0}} f_3(\boldsymbol{x}_3) + f_4(\boldsymbol{x}_4) + \mu_1(h_{3,1}(\boldsymbol{x}_3) + h_{4,1}(\boldsymbol{x}_4)). \tag{5.8}$$

Of course, problem (5.8) could again be solved by a decomposition method, which would result in a multilevel decomposition as illustrated in figure 3.2.

[7] For example, [x_3,x_4] denotes the new node obtained by merging the two similarly connected nodes x_3 and x_4.

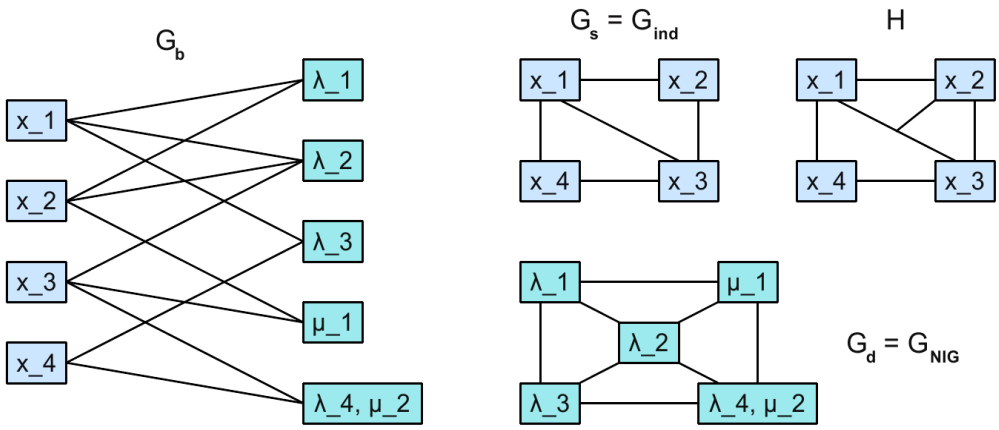

Figure 5.3: *Structure of the optimization problem represented by the bipartite graph G_b, the hypergraph H, and their induced graphs after merging nodes (ex. 5.1.3)*

The above graphs contain all information that is relevant to reveal different decompositions of problem (5.3) with constraints (5.4), as we will show in the following section.

Before we discuss possible decompositions, we present a second example with the same number of subproblems and constraints but with a quite different structure:

Example 5.1.3 (Constructing the bipartite graph G_b and the hypergraph H, as well as their induced graphs G_s, G_d, G_{NIG}, and G_{ind}, ex. b)). Consider again problem (5.3), but with differently structured functions $\boldsymbol{g}_i$ and $\boldsymbol{h}_i, i = 1, ..., 4$:

$$\boldsymbol{g}_1(\boldsymbol{x}_1) = \begin{pmatrix} g_{1,1}(\boldsymbol{x}_1) \\ g_{1,2}(\boldsymbol{x}_1) \\ g_{1,3}(\boldsymbol{x}_1) \\ 0 \end{pmatrix}, \boldsymbol{g}_2(\boldsymbol{x}_2) = \begin{pmatrix} g_{2,1}(\boldsymbol{x}_2) \\ g_{2,2}(\boldsymbol{x}_2) \\ 0 \\ 0 \end{pmatrix}, \boldsymbol{g}_3(\boldsymbol{x}_3) = \begin{pmatrix} 0 \\ g_{3,2}(\boldsymbol{x}_3) \\ 0 \\ g_{3,4}(\boldsymbol{x}_3) \end{pmatrix}, \boldsymbol{g}_4(\boldsymbol{x}_4) = \begin{pmatrix} 0 \\ 0 \\ g_{4,3}(\boldsymbol{x}_4) \\ g_{4,4}(\boldsymbol{x}_4) \end{pmatrix},$$

$$\boldsymbol{h}_1(\boldsymbol{x}_1) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \boldsymbol{h}_2(\boldsymbol{x}_2) = \begin{pmatrix} h_{2,1}(\boldsymbol{x}_2) \\ 0 \end{pmatrix}, \boldsymbol{h}_3(\boldsymbol{x}_3) = \begin{pmatrix} h_{3,1}(\boldsymbol{x}_3) \\ h_{3,2}(\boldsymbol{x}_3) \end{pmatrix}, \boldsymbol{h}_4(\boldsymbol{x}_4) = \begin{pmatrix} 0 \\ h_{4,2}(\boldsymbol{x}_4) \end{pmatrix}, \tag{5.9}$$

where the functions $g_{i,j}(\boldsymbol{x}_i)$ and $h_{i,j}(\boldsymbol{x}_i)$ are not identically zero.
The Lagrangian dual function of this problem has the form

$$d(\lambda_1, \lambda_2, \lambda_3, \lambda_4, \mu_1, \mu_2) = d_1(\lambda_1, \lambda_2, \lambda_3) + d_2(\lambda_1, \lambda_2, \mu_1) + d_3(\lambda_2, \lambda_4, \mu_1, \mu_2) + d_4(\lambda_3, \lambda_4, \mu_2). \tag{5.10}$$

Figure 5.3 shows the connectivity structure of the variables by the corresponding bipartite graph G_b and the induced graphs G_s and G_d, where the latter are identically equal to the induced graph G_{ind} and the net intersection graph G_{NIG} of the hypergraph H. H

contains one net, which is not a simple edge, between the nodes x_1, x_2, and x_3. It corresponds to the second inequality constraint $g_2(\boldsymbol{x}) = g_{1,2}(\boldsymbol{x}_1)+g_{2,2}(\boldsymbol{x}_2)+g_{3,2}(\boldsymbol{x}_3) \leq 0$. All nodes of the graphs have weight 1, except for the two nodes x_2 and [λ_4,μ_2], which have weight 2.

5.2 Determining Advantageous Decompositions

As described in section 3.5, the decomposition of optimization problems is realized by fixing appropriate primal or dual variables. We call these variables links. The subproblems, into which the original problem decomposes by fixing the links, can be solved independently. To solve the overall problem, the master problem, which is usually a nonsmooth problem in the links, adjusts the values of the links for which the subproblems are solved. If the links are primal variables, the method is referred to as primal decomposition, if they are dual variables, the method is called dual decomposition. If the overall problem is convex and fulfills Slater's condition, then, by theorem 3.3.4, its optimal value equals the optimal value of its dual problem. Otherwise, the optimal value of the dual problem is a lower bound on the optimal value of the primal problem, cf. theorem 3.3.3.
Conventionally, problem (5.1) is partitioned by dual decomposition into N subproblems connected by a master problem of $m+p$ links. This is the most granular decomposition of the problem in this general formulation. Further dual decompositions can be defined by aggregating some subproblems, or in other words, by different partitions of the N-element set of subproblems.[8] In case graph G_b, which represents the connection between primal and dual variables, is sparse, aggregating subproblems can reduce the number of links. This reduction occurs if global coupling constraints change into local ones.
The subproblems can be regarded as parametric problems in the links. If the optimal value functions of the subproblems were known, the original problem could be solved just by solving the master problem. Considering the approximation of optimal value functions of one or several subproblems, it would be preferable to determine subproblems of an appropriate size, which depend on as few links as possible. Furthermore, decompositions that result in subproblems of low intra-component connectivity would be preferable, since it eases their solution. If subproblems are supposed to be solved in parallel, a balanced size of components and a balanced intra-component connectivity is desirable. To analyze different decompositions of a block-separable problem (5.1), we use the graphical representation of connections between primal and dual variables, which we introduced in the previous section. We reveal that vertex-cuts of the hypergraph H correspond to different primal decompositions and that net-cuts of H imply different dual decompositions. The elements of a cut represent the links of a decompo-

[8] The total number of partitions of an n-element set is the Bell number B_n. Bell numbers satisfy the recursion $B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$. The first Bell numbers are $B_1 = 1, B_2 = 2, B_3 = 5, B_4 = 15, B_5 = 52, B_6 = 203, B_7 = 877, B_8 = 4140$.

sition, which are the variables of the master problem and which are fixed for solving the subproblems. Nets within the resulting components of the hypergraph represent those coupling constraints of the overall problem that change to local subproblem constraints of the related decomposition.
Before formalizing these results and the systematical exploration of decompositions of a certain problem, we continue with examples 5.1.2 and 5.1.3.

Example 5.2.1 (**Continuation of example 5.1.2: different decompositions**). Decomposition by fixing primal variables and decomposition by fixing dual variables are both possible. We discuss *dual decomposition* first because it is more convenient for coupling constraints. The dual function (5.5) represents the most granular decomposition of problem (5.3) with constraints (5.4). It has four subproblems coupled by six links. A coarser decomposition is obtained by aggregating subproblems x_3 and x_4. The related partial dual function (5.7) exhibits three subproblems coupled by three links of which two, λ_1 and λ_2, couple the same subproblems, namely those in $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$.
After merging nodes as described above and starting from the resulting coarser decomposition, all possible dual decompositions are obtained by different net-cuts of the hypergraph H shown in figure 5.2. The mentioned decomposition with three links, λ_1, λ_2, and μ_1, which is related to the partial dual function (5.7), corresponds to the largest net-cut consisting of two nets, one with weight 1 and one with weight 2. There are two different decompositions, the first has links λ_1 and λ_2, the second has link μ_1.
The decomposition with links λ_1 and λ_2 corresponds to an aggregation of the subproblems in variables $\boldsymbol{x}_2$, $\boldsymbol{x}_3$, and $\boldsymbol{x}_4$. In addition to the three local constraints in $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$, the first equality constraint changes into a local constraint of the subproblem in $\boldsymbol{x}_2$, $\boldsymbol{x}_3$, and $\boldsymbol{x}_4$ so that μ_1 vanishes from the master problem.
The decomposition with sole link μ_1 corresponds to an aggregation of the subproblems in variables $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. It turns the first two coupling constraints into local constraints of the subproblem in $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ so that λ_1 and λ_2 vanish from the master problem.
Summarizing, there are the following convenient dual decompositions, represented by the corresponding Lagrangian (partial) dual functions:

$$\begin{aligned} d(\lambda_1,\lambda_2,\lambda_3,\lambda_4,\mu_1,\mu_2) &= d_1(\lambda_1,\lambda_2)+d_2(\lambda_1,\lambda_2,\mu_1)+d_3(\lambda_3,\lambda_4,\mu_1,\mu_2)+d_4(\lambda_3,\lambda_4,\mu_1,\mu_2),\\ d^{part}_{basic}(\lambda_1,\lambda_2,\mu_1) &= d_1(\lambda_1,\lambda_2)+d_2(\lambda_1,\lambda_2,\mu_1)+d_{3,4}(\mu_1),\\ d^{part}_{A}(\lambda_1,\lambda_2) &= d_1(\lambda_1,\lambda_2)+d_{2,3,4}(\lambda_1,\lambda_2),\\ d^{part}_{B}(\mu_1) &= d_{1,2}(\mu_1)+d_{3,4}(\mu_1), \end{aligned} \tag{5.11}$$

where

$$d_{1,2}(\mu_1) = \inf_{\substack{\boldsymbol{x}_1,\boldsymbol{x}_2\\ g_1(\boldsymbol{x}_1,\boldsymbol{x}_2)\le 0\\ g_2(\boldsymbol{x}_1,\boldsymbol{x}_2)\le 0}} f_1(\boldsymbol{x}_1)+f_2(\boldsymbol{x}_2)+\mu_1 h_{2,1}(\boldsymbol{x}_2), \tag{5.12a}$$

$$d_{3,4}(\mu_1) = \inf_{\substack{\boldsymbol{x}_3,\boldsymbol{x}_4 \\ g_3(\boldsymbol{x}_3,\boldsymbol{x}_4)\leq 0 \\ g_4(\boldsymbol{x}_3,\boldsymbol{x}_4)\leq 0 \\ h_2(\boldsymbol{x}_3,\boldsymbol{x}_4)=0}} f_3(\boldsymbol{x}_3) + f_4(\boldsymbol{x}_4) + \mu_1(h_{3,1}(\boldsymbol{x}_3) + h_{4,1}(\boldsymbol{x}_4)), \tag{5.12b}$$

$$d_{2,3,4}(\lambda_1,\lambda_2) = \inf_{\substack{\boldsymbol{x}_2,\boldsymbol{x}_3,\boldsymbol{x}_4 \\ g_3(\boldsymbol{x}_3,\boldsymbol{x}_4)\leq 0 \\ g_4(\boldsymbol{x}_3,\boldsymbol{x}_4)\leq 0 \\ h_1(\boldsymbol{x}_2,\boldsymbol{x}_3,\boldsymbol{x}_4)=0 \\ h_2(\boldsymbol{x}_3,\boldsymbol{x}_4)=0}} f_2(\boldsymbol{x}_2) + f_3(\boldsymbol{x}_3) + f_4(\boldsymbol{x}_4) + \lambda_1 g_{2,1}(\boldsymbol{x}_2) + \lambda_2 g_{2,2}(\boldsymbol{x}_2). \tag{5.12c}$$

The master problem of a certain decomposition consists in maximizing the respective dual function in (5.11) under the condition that $\lambda_i \geq 0 \ \forall\ i$.
Besides the three subproblem aggregations of (5.12), and the trivial one of aggregating all subproblems, there is no other possibility to reduce the number of links by aggregating subproblems, cf. also theorem 5.2.3. As mentioned above, the aggregated problems of (5.12) can in turn be solved by a decomposition method which results in multilevel decompositions.
Note that the constraint $h_1(\boldsymbol{x}) = h_{2,1}(\boldsymbol{x}_2) + h_{3,1}(\boldsymbol{x}_3) + h_{4,1}(\boldsymbol{x}_4) \leq 0$ is the only constraint whose sole relaxation decomposes the optimization problem. This can also be seen in the hypergraph H in figure 5.2: The net between x_2 and [x_3,x_4], which corresponds to μ_1, is the only net-cut of H with lowest possible weight 1. This cut induces also a balanced partition with vertex weights of components given by $w(\{\text{x\_1, x\_2}\}) = 3$ and $w([\text{x\_3,x\_4}]) = 2$, and net weights of components given by $w([\lambda\_1, \lambda\_2]) = 2$ and $w([\lambda\_3, \lambda\_4, \mu\_2]) = 3$.
Let us now consider *primal decomposition*. $\boldsymbol{x}_2$ is the only primal variable whose fixing decomposes the optimization problem. Namely, by fixing $\boldsymbol{x}_2 = t$ in the subproblems, we can formulate problem (5.3) with constraints (5.4) by

$$\min_t f_2(t) + \Big(\inf_{\substack{\boldsymbol{x}_1 \\ g_1(\boldsymbol{x}_1,t)\leq 0 \\ g_2(\boldsymbol{x}_1,t)\leq 0}} f_1(\boldsymbol{x}_1) + \inf_{\substack{\boldsymbol{x}_3,\boldsymbol{x}_4 \\ g_3(\boldsymbol{x}_3,\boldsymbol{x}_4)\leq 0 \\ g_4(\boldsymbol{x}_3,\boldsymbol{x}_4)\leq 0 \\ h_1(t,\boldsymbol{x}_3,\boldsymbol{x}_4)=0 \\ h_2(\boldsymbol{x}_3,\boldsymbol{x}_4)=0}} f_3(\boldsymbol{x}_3) + f_4(\boldsymbol{x}_4) \Big) = \min_t f_2(t) + \phi_1(t) + \phi_{3,4}(t), \tag{5.13}$$

where $\phi_1(t)$ and $\phi_{3,4}(t)$ are the optimal value functions of the related subproblems. Fixing further primal variables does not result in any further decomposition of problem (5.13). However, the second subproblem, in variables $\boldsymbol{x}_3$ and $\boldsymbol{x}_4$, could be solved by dual decomposition, which would result in a multilevel decomposition.

Example 5.2.2 (**Continuation of example 5.1.3: different decompositions**)**.** Let us again start with *dual decomposition*. All possible dual decompositions are obtained by different net-cuts of the hypergraph H shown in figure 5.3. The dual function (5.10),

$$d(\lambda_1,\lambda_2,\lambda_3,\lambda_4,\mu_1,\mu_2) = d_1(\lambda_1,\lambda_2,\lambda_3) + d_2(\lambda_1,\lambda_2,\mu_1) + d_3(\lambda_2,\lambda_4,\mu_1,\mu_2) + d_4(\lambda_3,\lambda_4,\mu_2), \tag{5.14}$$

represents the most granular decomposition of problem (5.3) with constraints (5.9). It has four subproblems coupled by six links and corresponds to the largest net-cut of H, which consists of five nets, four of which have weight 1 and one has weight 2. There are ten further convenient dual decompositions, which can be represented by the following partial dual functions:

$$\begin{aligned}
d_A^{part}(\lambda_2,\lambda_3,\lambda_4,\mu_1,\mu_2) &= d_{1,2}(\lambda_2,\lambda_3,\mu_1) + d_3(\lambda_2,\lambda_4,\mu_1,\mu_2) + d_4(\lambda_3,\lambda_4,\mu_2),\\
d_B^{part}(\lambda_1,\lambda_2,\lambda_4,\mu_1,\mu_2) &= d_{1,4}(\lambda_1,\lambda_2,\lambda_4,\mu_2) + d_2(\lambda_1,\lambda_2,\mu_1) + d_3(\lambda_2,\lambda_4,\mu_1,\mu_2),\\
d_C^{part}(\lambda_1,\lambda_2,\lambda_3,\lambda_4,\mu_2) &= d_1(\lambda_1,\lambda_2,\lambda_3) + d_{2,3}(\lambda_1,\lambda_2,\lambda_4,\mu_2) + d_4(\lambda_3,\lambda_4,\mu_2),\\
d_D^{part}(\lambda_1,\lambda_2,\lambda_3,\mu_1) &= d_1(\lambda_1,\lambda_2,\lambda_3) + d_2(\lambda_1,\lambda_2,\mu_1) + d_{3,4}(\lambda_2,\lambda_3,\mu_1),
\end{aligned} \tag{5.15}$$

$$\begin{aligned}
d_E^{part}(\lambda_2,\lambda_3,\mu_1) &= d_{1,2}(\lambda_2,\lambda_3,\mu_1) + d_{3,4}(\lambda_2,\lambda_3,\mu_1),\\
d_F^{part}(\lambda_1,\lambda_2,\lambda_4,\mu_2) &= d_{1,4}(\lambda_1,\lambda_2,\lambda_4,\mu_2) + d_{2,3}(\lambda_1,\lambda_2,\lambda_4,\mu_2),
\end{aligned} \tag{5.16}$$

$$\begin{aligned}
d_G^{part}(\lambda_3,\lambda_4,\mu_2) &= d_{1,2,3}(\lambda_3,\lambda_4,\mu_2) + d_4(\lambda_3,\lambda_4,\mu_2),\\
d_H^{part}(\lambda_2,\lambda_4,\mu_1,\mu_2) &= d_{1,2,4}(\lambda_2,\lambda_4,\mu_1,\mu_2) + d_3(\lambda_2,\lambda_4,\mu_1,\mu_2),\\
d_I^{part}(\lambda_1,\lambda_2,\mu_1) &= d_{1,3,4}(\lambda_1,\lambda_2,\mu_1) + d_2(\lambda_1,\lambda_2,\mu_1),\\
d_J^{part}(\lambda_1,\lambda_2,\lambda_3) &= d_1(\lambda_1,\lambda_2,\lambda_3) + d_{2,3,4}(\lambda_1,\lambda_2,\lambda_3),
\end{aligned} \tag{5.17}$$

where each optimal value function d_i of an aggregated subproblem i is defined analogously to those of example 5.2.1.
The decompositions in (5.15) and (5.17) are obtained by net-cuts of H that separate single nodes/subproblems. These net-cuts are not obtained by vertex-cuts of H's net intersection graph G_{NIG} shown in figure 5.3. In contrast, each subproblem of the two decompositions in (5.16) arose from an aggregation of two single subproblems in $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$, or $\boldsymbol{x}_4$, and includes at least one former global constraint that changed into a local one. These two decompositions are more balanced than those of (5.15) and (5.17), and they are induced by the two non-trivial vertex-cuts of G_{NIG}, given by $\{\lambda\_2, \lambda\_3, \mu\_1\}$ and $\{\lambda\_1, \lambda\_2, [\lambda\_4, \mu\_2]\}$.
Next, we consider the *primal decompositions*. The only subsets of primal variables whose fixing decomposes the optimization problem is $\{\boldsymbol{x}_1, \boldsymbol{x}_3\}$. By fixing $\boldsymbol{x}_1 = t$ and $\boldsymbol{x}_3 = u$ in the subproblems, we can formulate problem (5.3) with constraints (5.9) by

$$\begin{aligned}
&\min_{t,u} f_1(t) + f_3(u) + \Big(\inf_{\substack{\boldsymbol{x}_2\\ g_1(t,\boldsymbol{x}_2)\le 0\\ g_2(t,\boldsymbol{x}_2,u)\le 0\\ h_1(\boldsymbol{x}_2,u)=0}} f_2(\boldsymbol{x}_2) + \inf_{\substack{\boldsymbol{x}_4\\ g_3(t,\boldsymbol{x}_4)\le 0\\ g_4(u,\boldsymbol{x}_4)\le 0\\ h_2(u,\boldsymbol{x}_4)=0}} f_4(\boldsymbol{x}_4) \Big)\\
=&\min_{t,u} f_1(t) + f_3(u) + \phi_2(t,u) + \phi_4(t,u),
\end{aligned} \tag{5.18}$$

where $\phi_2(t,u)$ and $\phi_4(t,u)$ are the optimal value functions of the related subproblems. Fixing different primal variables does not result in any further decomposition of problem (5.18).

In addition to primal and dual decomposition, *additional decomposition* schemes can be obtained by *simultaneously fixing suitable primal and dual variables*. In the treated problem, for example, fixing $\lambda_3 = t$ and $\boldsymbol{x}_3 = u$ decomposes problem (5.3) with constraints (5.9) in the following way:

$$\begin{aligned}
&\max_{t\geq 0} \inf_{u} f_3(u) + \Big(\inf_{\substack{\boldsymbol{x}_1,\boldsymbol{x}_2 \\ g_1(\boldsymbol{x}_1,\boldsymbol{x}_2)\leq 0 \\ g_2(\boldsymbol{x}_1,\boldsymbol{x}_2,u)\leq 0 \\ h_1(\boldsymbol{x}_2,u)=0}} f_1(\boldsymbol{x}_1) + f_2(\boldsymbol{x}_2) + t g_{1,3}(\boldsymbol{x}_1) + \inf_{\substack{\boldsymbol{x}_4 \\ g_4(u,\boldsymbol{x}_4)\leq 0 \\ h_2(u,\boldsymbol{x}_4)=0}} f_4(\boldsymbol{x}_4) + t g_{4,3}(\boldsymbol{x}_4) \Big) \\
&= \max_{t\geq 0} \inf_{u} f_3(u) + \phi_{1,2}^{g3}(t,u) + \phi_4^{g3}(t,u),
\end{aligned} \tag{5.19}$$

where $\phi_{1,2}^{g3}$ and ϕ_4^{g3} are the optimal value functions of the related partially relaxed subproblems. Such "mixed primal-dual decompositions" are obtained by vertex-cuts of the bipartite graph G_b that include elements of both vertex sets, V_s^{red} and V_d^{red}, as will be briefly explained in remark 5.2.10.

Regarding decompositions of a block-separable optimization problem, as stated in (5.1), and partitions of the related hypergraph H, defined in section 5.1, we obtain several general results.

Theorem 5.2.3. *If the aggregation of two subproblems i_1 and i_2 of problem* (5.1) *changes a coupling constraint to a local constraint, i_1 and i_2 are represented in H by the same merged vertex, or the vertices corresponding to i_1 and i_2 are neighbors in the hypergraph H and in its induced graph G_{ind}.*

Proof To change a coupling constraint $g_j(\boldsymbol{x}) = \sum_{i=1}^{N} g_{i,j}(\boldsymbol{x}_i) \leq 0$ (or analogously $h_j(\boldsymbol{x}) = 0$) to a local constraint, both problems must be affected by this constraint, i.e., $g_{i_1 j} \not\equiv 0$ and $g_{i_2 j} \not\equiv 0$. Either both subproblems are affected by exactly the same constraints, i.e., they are represented in G_b, and, therefore, in H, by the same vertex, or both subproblems are connected in G_b by the dual variable λ_j related to the constraint $g_j(\boldsymbol{x}) \leq 0$, which implies that they are neighbors in H. Furthermore, neighbors in H are also neighbors in G_{ind}.

Similar to definition 4.5.2 of a narrow separator, we introduce the following definitions:

Definition 5.2.4. *Consider a set $I \subset \{1, ..., N\}$ of subproblems for which the fixing of the related variables $\boldsymbol{x}_i, i \in I$, decomposes the overall optimization problem into $k \leq N - |I|$ independently solvable parts. We call this set I* ***narrow*** *if there is no subset $I' \subset I$ for which the fixing of variables $\boldsymbol{x}_i, i \in I'$, decomposes the overall optimization problem into k independently solvable parts.*

Definition 5.2.5. *Consider a set $J \subseteq \{1, ..., m + p\}$ of coupling constraints whose relaxation decomposes the overall optimization problem into $k \leq N$ independently solvable*

parts. We call this set J ***narrow*** *if there is no subset $J' \subset J$ of coupling constraints whose relaxation decomposes the overall optimization problem into k independently solvable parts.*

To prove the major results of theorems 5.2.8 and 5.2.9, we will use the following lemmata.

Lemma 5.2.6. *If two subproblems $i_1, i_2 \in \{1, ..., N\}$ are represented by the same node of H, a narrow set of subproblems, for which the fixing of related variables decomposes the overall problem, contains either none or both of i_1, i_2. If the narrow set does not contain them, one of the resulting independently solvable parts contains both, i_1 and i_2.*

Proof Let I be a narrow set of subproblems for which variable fixing decomposes the overall optimization problem into parts $\{P_1, ..., P_k\}$, with $P_i \subset \{1, ..., N\} \setminus I \ \forall \ i \in \{1, ..., k\}$, $P_i \bigcap P_j = \emptyset \ \forall \ i \neq j \in \{1, ..., k\}$, $\bigcup_{i=1}^{k} P_i = \{1, ..., N\} \setminus I$. If two subproblems $i_1, i_2 \in \{1, ..., N\}$ are represented by the same node of H, both problems are affected by exactly the same set of constraints. Let us assume that I contains only subproblem i_1, but not i_2. Since I is narrow there exists a constraint $g_j(\boldsymbol{x}) = \sum_{i=1}^{N} g_{i,j}(\boldsymbol{x}_i) \leq 0$ (or $h_j(\boldsymbol{x}) = 0$) with $g_{i_1,j} \not\equiv 0, g_{p,j} \not\equiv 0$, and $g_{q,j} \not\equiv 0$ for $p \in P_i$, $q \in P_j, i \neq j \in \{1, ..., k\}$. However, since $i_1 \in I$ and $i_2 \in \{1, ..., N\} \setminus I$ are assumed to be represented by the same node, we must also have $g_{i_2,j} \not\equiv 0$, which means that the parts P_i and P_j cannot be solved independently, and which contradicts that I is a narrow set of subproblems for which variable fixing decomposes the overall optimization problem into k parts.
The same argument shows that a part $P_i, i \in \{1, ..., k\}$, contains either none or both of i_1, i_2.

Lemma 5.2.7. *If two coupling constraints $j_1, j_2 \in \{1, ..., m + p\}$ are represented by the same net of H, a narrow set of coupling constraints, whose relaxation decomposes the overall problem, contains either none or both of j_1, j_2. If the narrow set does not contain them, one of the resulting independently solvable parts contains both, j_1 and j_2.*

Proof Let J be a narrow set of coupling constraints whose relaxation decomposes the overall optimization problem into parts $\{P_1, ..., P_k\}, P_i \subset \{1, ..., N\} \ \forall \ i \in \{1, ..., k\}$, $P_i \bigcap P_j = \emptyset \ \forall \ i \neq j \in \{1, ..., k\}$, $\bigcup_{i=1}^{k} P_i = \{1, ..., N\}$. If two coupling constraints j_1, j_2 are represented by the same net of H, both coupling constraints affect exactly the same set of subproblems. Let us assume that J contains only the coupling constraints j_1, but not j_2. Since J is narrow, the constraint $g_{j_1}(\boldsymbol{x}) = \sum_{i=1}^{N} g_{i,j_1}(\boldsymbol{x}_i) \leq 0$ (or $h_{j_1}(\boldsymbol{x}) = 0$) couples two subproblems $p \in P_i$, $q \in P_j, i \neq j \in \{1, ..., k\}$, i.e., $g_{p,j_1} \not\equiv 0$, and $g_{q,j_1} \not\equiv 0$. However, since j_1 and j_2 are assumed to be represented by the same net, we must also have $g_{p,j_2} \not\equiv 0$ and $g_{q,j_2} \not\equiv 0$, which means that the parts P_i and P_j cannot be solved independently, and which contradicts that J is a narrow set of coupling constraints whose relaxation decomposes the overall optimization problem into k parts.
The same argument shows that a part $P_i, i \in \{1, ..., k\}$, contains either none or both of j_1, j_2.

Theorem 5.2.8. *A narrow set $I \subset \{1, ..., N\}$ of subproblems for which the fixing of variables $\boldsymbol{x}_i, i \in I$, decomposes the overall optimization problem into $k \leq N - |I|$ independently solvable parts uniquely corresponds to a narrow k-way vertex-separator $V \subset V_s^{\text{red}}$ of H.*
The weight of the vertex-separator V, $\sum_{v \in V} w(v)$, equals $\sum_{i \in I} n_i$, where n_i is the dimension of $\boldsymbol{x}_i$. The dimension of each independently solvable subproblem is equal to the sum of vertex-weights in the corresponding component of $H - V$.

Proof a) Let I be such a narrow set of subproblems, and let $\{P_1, ..., P_k\}$ be the related partition of subproblems, i.e., $P_i \subset \{1, ..., N\} \setminus I \; \forall \; i \in \{1, ..., k\}, P_i \bigcap P_j = \emptyset \; \forall \; i \neq j \in \{1, ..., k\}, \bigcup_{i=1}^{k} P_i = \{1, ..., N\} \setminus I$. If two subproblems $i_1, i_2 \in \{1, ..., N\}$ are represented by the same node of H, either I, or one part $P_i, i \in \{1, ..., k\}$ contains both subproblems i_1, i_2, cf. lemma 5.2.6. Therefore, the partition $\{P_1, ..., P_k, I\}$ of the index set $\{1, ..., N\}$ implies a partition $\{V_1, ..., V_k, V_I\}$ of V_s^{red}. $H - V_I$ must decompose into k components with vertex sets $V_1, ..., V_k$ since a hyperedge connecting $v_i \in V_i$ and $v_j \in V_j, i \neq j \in \{1, ..., k\}$ would imply a coupling constraint between two parts P_i and P_j. If V_I were not narrow, this would contradict that I is narrow.
b) Let $V \subset V_s^{\text{red}}$ be a narrow vertex-separator of H that induces a partition $\{V_1, ..., V_k, V\}$ of V_s^{red}. By H's definition, it naturally implies a partition $\{P_1, ..., P_k, P_V\}$ of the index set $\{1, ..., N\}$ of subproblems. If there were a coupling constraint connecting two parts P_i and $P_j, i \neq j \in \{1, ..., k\}$, the corresponding vertex-sets V_i and V_j would be connected by a net because of the correspondence between nets of H and coupling constraints. Furthermore, P_V must be a narrow set of subproblems for which the fixing of variables $\boldsymbol{x}_i, i \in P_V$, decomposes the overall optimization problem, since otherwise V would not be a narrow separator.
The equality of weights and dimensions is due to the definition of vertex-weights in H.

Theorem 5.2.9. *A narrow set $J \subseteq \{1, ..., m+p\}$ of coupling constraints whose relaxation decomposes the overall optimization problem into $k \leq N$ independently solvable parts uniquely corresponds to a narrow k-way net-separator $N_J \subseteq N_d$ of H.*
The weight of the net-separator N_J is equal to the cardinality of J: $\sum_{n \in N_J} w(n) = |J|$. The dimension of each independently solvable subproblem is equal to the sum of vertex-weights in the corresponding component of $H' := (V_s^{\text{red}}, N_d \setminus N_J)$.

Proof a) Let J be a narrow set of coupling constraints whose relaxation decomposes the optimization problem into parts $\{P_1, ..., P_k\}$, with $P_i \subset \{1, ..., N\} \; \forall \; i \in \{1, ..., k\}$, $P_i \bigcap P_j = \emptyset \; \forall \; i \neq j \in \{1, ..., k\}, \bigcup_{i=1}^{k} P_i = \{1, ..., N\}$. By lemma 5.2.7, two coupling constraints which are represented by the same net of H are either contained in J or in one of the independently solvable parts. Therefore, J implies a set of nets $N_J \subseteq N_d$. N_J is a k-way net-separator of H, since $H' := (V_s^{\text{red}}, N_d \setminus N_J)$ decomposes into k components naturally corresponding to $\{P_1, ..., P_k\}$. If N_J were not narrow, J would not be narrow.
b) Let $N_J \subseteq N_d$ be a narrow net-separator of H inducing a partition $\{V_1, ..., V_k\}$ of V_s^{red}. By H's definition, N_J and $\{V_1, ..., V_k\}$ naturally imply a set $J \subseteq \{1, ..., m+p\}$ of coupling

constraints and a partition $\{P_1, ..., P_k\}$ of the index set $\{1, ..., N\}$ of subproblems, respectively. If there were a coupling constraint $l \in \{1, ..., m+p\} \setminus J$ connecting two parts P_i and $P_j, i \neq j \in \{1, ..., k\}$, the corresponding vertex-sets V_i and V_j would be connected by a net $n_l \in N_d \setminus N_J$ because of the correspondence between nets of H and coupling constraints. Furthermore, the set J must be a narrow set of coupling constraints whose relaxation decomposes the overall optimization problem, since otherwise N_J would not be a narrow separator.
The equality of weights and dimensions is due to the definitions of vertex-weights and net-weights in H.

Theorems 5.2.8 and 5.2.9 show that primal decompositions of an optimization problem uniquely correspond to vertex-cuts of the hypergraph H, and dual decompositions uniquely correspond to net-cuts of H. Thus the task is to detect suitable vertex-cuts and net-cuts of H.
Since by theorem 4.5.19, vertex-cuts of the hypergraph H are identical to those of its induced graph G_{ind}, one can determine them by methods and algorithms presented in sections 4.4 and 4.5. As originally shown by [KPcA12] and outlined in section 4.5, every vertex-cut of H's net intersection graph G_{NIG} implies a net-cut of H. However, as seen in examples 5.2.1 and 5.2.2, not every net-cut of H corresponds to a vertex-cut of G_{NIG}. Namely, if a net-cut of H separates single nodes of H, i.e., if at least one of the resulting components includes only one vertex and, therefore, no net, this net-cut is not represented by a vertex-cut of G_{NIG}. Especially for large problems, such decompositions with a very small component are obviously less favorable than a decomposition of which each component includes at least one net. Thus, it is reasonable to search for vertex-cuts of G_{NIG} to partition H.
Possible quality criteria for graph partitions are a small cut-size, a predetermined number of components, balanced size of components, or low intra-component connectivity, cf. section 4.5. Which of them are most important for a decomposition, depends on the respective application and on the purpose of the decomposition. According to specific aims, distinct algorithms for graph partitioning should be applied. For example, to approximate a subproblem, as described in the following section, a small cut-size is very important.
Since the quality of an optimization result does not depend on the chosen decomposition in a well-known way, we do not necessarily benefit from the, in some predefined sense, optimal decomposition of a large problem. Therefore, it is reasonable to use fast heuristics to compute good partitions of the hypergraph H.

Remark 5.2.10. Besides primal and dual decomposition, further decomposition schemes can be obtained by simultaneously fixing suitable primal and dual variables. We illustrated this at the end of example 5.2.2. Such decompositions correspond to vertex-cuts of the bipartite graph G_b that include elements of both vertex sets, V_s^{red} and V_d^{red}. In the hypergraph H this would correspond to a "mixed vertex-net-cut" which is not considered in graph theory. This fact justifies the definition of the bipartite graph G_b in addition

to the hypergraph H, even if both graphs display quite similar information.
The described decompositions can become useful if the objective is to find a cut of as few links as possible, since a "mixed vertex-net-cut" could have a smaller weight than a pure vertex-cut or a pure net-cut.

5.3 Approximating Subproblems

As mentioned above, subproblems of an overall problem (5.1) can be regarded as parametric problems in the links. If the optimal value functions of the subproblems were known, the overall problem could be solved just by solving the master problem. If one aims to solve a decomposed problem multiple times for varying parameters in certain subproblems, while the parameters in others are fixed, the approximation of subproblems with fixed parameters could be timesaving. Apart from this, in applications, the establishment of a parametric model for the optimal value function of a subproblem can be interesting with regard to modeling and analysis. In this section, we discuss the approximation of subproblems' optimal value functions, which are in general non-differentiable. We mainly focus on primal decomposition and dual decomposition. Approximating optimal value functions of the subproblems in case of a mixed primal-dual decomposition is mentioned in remark 5.3.2.
In this section, $\phi_i(\boldsymbol{u}^i)$ denotes the optimal value function of a subproblem i in case of primal decomposition, and $d_i(\boldsymbol{s}^i, \boldsymbol{t}^i)$ denotes the optimal value function of a subproblem i in case of dual decomposition. $\boldsymbol{u}^i$ and $(\boldsymbol{s}^i, \boldsymbol{t}^i)$ denote the links that connect subproblem i with the master problem. They correspond to certain components of the primal variable $\boldsymbol{x}$ and of the dual variable $(\boldsymbol{\lambda}, \boldsymbol{\mu})$, respectively. Let $I^i \subset \{1, ..., N\}$ (primal decomposition) and $J^i \subseteq \{1, ..., m+p\}$ (dual decomposition) denote the index sets of the corresponding links.

Definition 5.3.1. *We call the number of links of a subproblem i, which is the dimension of $\boldsymbol{u}^i$ or of $(\boldsymbol{s}^i, \boldsymbol{t}^i)$, the **link degree** d_l^i of subproblem i. For dual decomposition, the link degree splits into a part corresponding to inequality relaxation and a part corresponding to equality relaxation: $d_l^i = d_{li}^i + d_{le}^i$.*

Furthermore, in this section, we denote the variables of a subproblem i by $\widetilde{\boldsymbol{x}}^i$. $\widetilde{\boldsymbol{x}}^i$ comprises several components $\boldsymbol{x}_k, k \in P_i \subset \{1, ..., N\}$, which have dimension n_k. The subproblem dimension, which is the dimension of $\widetilde{\boldsymbol{x}}^i$, is given by $\widetilde{N}_i := \sum_{k \in P_i} n_k$.
For the general case of an overall problem (5.1) with coupling constraints, the optimal value functions of the subproblems, $\phi_i(\boldsymbol{u}^i) : \mathbb{R}^{d_l^i} \to \mathbb{R}$ or $d_i(\boldsymbol{s}^i, \boldsymbol{t}^i) : (\mathbb{R}_0^+)^{d_{li}^i} \times \mathbb{R}^{d_{le}^i} \to \mathbb{R}$, have the form

$$\phi_i(\boldsymbol{u}^i) = \inf_{\widetilde{\boldsymbol{x}}^i \in \widetilde{\mathcal{F}}^i(\boldsymbol{u}^i)} f^i(\widetilde{\boldsymbol{x}}^i), \tag{5.20}$$

$$d_i(\boldsymbol{s}^i, \boldsymbol{t}^i) = \inf_{\widetilde{\boldsymbol{x}}^i \in \widetilde{\mathcal{F}}^i} f^i(\widetilde{\boldsymbol{x}}^i) + \sum_{k \in J^i \cap \{1,...,m\}} s_k^i g_k^i(\widetilde{\boldsymbol{x}}^i) + \sum_{k \in J^i \cap \{m+1,...,m+p\}} t_{k-m}^i h_{k-m}^i(\widetilde{\boldsymbol{x}}^i). \tag{5.21}$$

$\widetilde{\mathcal{F}}^i$ denotes the local feasible set of subproblem i in variable $\widetilde{\boldsymbol{x}}^i$ and $\widetilde{\mathcal{F}}^i(\boldsymbol{u}^i)$ has the form

$$\widetilde{\mathcal{F}}^i(\boldsymbol{u}^i) := \{\ \widetilde{\boldsymbol{x}}^i \in \widetilde{\mathcal{F}}^i \mid g_k(\boldsymbol{u}^i, \widetilde{\boldsymbol{x}}^i) \leq 0\ \forall\ k \in \widetilde{J}_i^i \text{ and } h_k(\boldsymbol{u}^i, \widetilde{\boldsymbol{x}}^i) = 0\ \forall\ k \in \widetilde{J}_e^i\ \}, \quad (5.22)$$

where the index set $\widetilde{J}_i^i$ corresponds to those inequality constraints k with $g_{l,k}(\boldsymbol{x}_l) \equiv 0$ $\forall\ l \in \{1, ..., N\} \setminus (I^i \cup P_i)$ and $\widetilde{J}_e^i$ corresponds to those equality constraints k with $h_{l,k}(\boldsymbol{x}_l) \equiv 0\ \forall\ l \in \{1, ..., N\} \setminus (I^i \cup P_i)$. Furthermore,

$$\begin{aligned} f^i(\widetilde{\boldsymbol{x}}^i) &:= \sum_{l \in P_i} f_l(\boldsymbol{x}_l), \qquad g_k^i(\widetilde{\boldsymbol{x}}^i) := \sum_{l \in P_i} \boldsymbol{g}_{l,k}^i(\boldsymbol{x}_l), \quad \forall\ k \in J^i \cap \{1, ..., m\}, \quad \text{and} \\ h_k^i(\widetilde{\boldsymbol{x}}^i) &:= \sum_{l \in P_i} \boldsymbol{h}_{l,k-m}^i(\boldsymbol{x}_l), \quad \forall\ k \in J^i \cap \{m+1, ..., m+p\}. \end{aligned} \quad (5.23)$$

Explicit examples for the optimal value functions (5.20) and (5.21) are given in (5.13) and (5.12). Continuity properties of optimal value functions can be derived from theorem 3.2.4. Theorem 3.2.7 provides conditions under which they are convex. Conditions for local differentiability of the optimal value functions are stated in theorems 3.2.10 and 3.2.11.

Let us first consider *primal decomposition*. Under appropriate continuity and convexity assumptions on the overall problem (5.1), the optimal value function ϕ_i of a subproblem i, given by (5.20), is continuous and convex. Regarding a decomposition of the overall problem into k subproblems, these properties of the subproblems' optimal value functions ensure convexity of the master problem

$$\min_{\boldsymbol{u} \in \mathcal{F}_u} \sum_{i=1}^{k} \phi_i(\boldsymbol{u}^i), \quad (5.24)$$

where $\boldsymbol{u}$ is the union of all subproblem links, corresponding to the index set $\bigcup_{i=1}^k I^i$, and $\mathcal{F}_u$ consists of the inequality and equality constraints that do not affect variables $\boldsymbol{x}_l, l \in \{1, ..., N\} \setminus \bigcup_{i=1}^k I^i$. However, since the functions ϕ_i need not be differentiable, in general, problem (5.24) is nonsmooth.
To analyze the problem, it may be useful to approximate the optimal value functions ϕ_i of some subproblems. The same can be profitable if variants of the overall problem have to be solved many times while certain subproblems remain the same. To approximate a submodel function ϕ_i, we must realize several points:

1. Choose K values of $\boldsymbol{u}^i$ within a domain over which ϕ_i should be approximated and generate K data points $(\boldsymbol{u}_k^i, \phi_i(\boldsymbol{u}_k^i))_{(k=1,...,K)}$ by solving the subproblem for each value $\boldsymbol{u}_k^i,\ k = 1, ..., K$.

2. Select a parametric model ϕ_i^θ for ϕ_i, where the dimension of ϕ_i^θ's domain equals the link degree d_l^i.

3. Fit ϕ_i^θ to the data $(\boldsymbol{u}_k^i, \phi_i(\boldsymbol{u}_k^i))_{(k=1,...,K)}$ by determining an estimator $\hat{\theta}$ of the parameter θ. The resulting function $\phi_i^{\hat{\theta}}$ serves as an approximation $\widehat{\phi}_i : \mathcal{F}_u \to \mathbb{R}$ of ϕ_i.

We will shortly discuss these three steps after considering *dual decomposition*. Since the optimal value function d_i of a subproblem i, given by (5.21), is the pointwise infimum of a family of linear functions in $(\boldsymbol{s}^i, \boldsymbol{t}^i)$, it is concave and continuous, even if the overall problem (5.1) is not convex. Therefore, a dual master problem of the form

$$\min_{\boldsymbol{s}^i \geq 0, \boldsymbol{t}^i} -\sum_{i=1}^{k} d_i(\boldsymbol{s}^i, \boldsymbol{t}^i), \tag{5.25}$$

is a convex problem. However, since $d_i, i = 1, ..., k$, need not be differentiable, the master problem is in general nonsmooth. Approximating a submodel function d_i is similar to approximating ϕ_i:

1. Choose K values of $(\boldsymbol{s}^i, \boldsymbol{t}^i)$ within a domain over which d_i should be approximated and generate K data points $((\boldsymbol{s}_k^i, \boldsymbol{t}_k^i), d_i(\boldsymbol{s}_k^i, \boldsymbol{t}_k^i))_{(k=1,...,K)}$ by solving the subproblem for each value $(\boldsymbol{s}_k^i, \boldsymbol{t}_k^i),\ k \in \{1, ..., K\}$.

2. Select a parametric model d_i^θ for d_i, where the dimension of d_i^θ's domain equals the link degree $d_l^i = d_{li}^i + d_{le}^i$.

3. Fit d_i^θ to the data $((\boldsymbol{s}_k^i, \boldsymbol{t}_k^i), d_i(\boldsymbol{s}_k^i, \boldsymbol{t}_k^i))_{(k=1,...,K)}$ by determining an estimator $\hat{\theta}$ of the parameter θ. The resulting function $d_i^{\hat{\theta}}$ serves as an approximation $\widehat{d}_i : (\mathbb{R}_0^+)^{d_{li}^i} \times \mathbb{R}^{d_{le}^i} \to \mathbb{R}$ of d_i.

Generating data points:
The choice of the data points $\boldsymbol{u}_k^i$ and $(\boldsymbol{s}_k^i, \boldsymbol{t}_k^i)$, in particular the choice of the domain that they cover, is usually crucial for a suitable approximation of the optimal value functions ϕ_i and d_i. Conventionally, these values are chosen as multidimensional grid points.
Since the links $\boldsymbol{u}^i$ of a primal decomposition correspond to some primal variables of the overall model, they are often restricted by constraints of the overall problem. Basically, one should choose $\boldsymbol{u}_k^i \in \mathcal{F}_u$, where $\mathcal{F}_u$ is the feasible set of master problem (5.24). Further restrictions on the data points could be implied by constraints $g_k(\boldsymbol{u}^i, \widetilde{\boldsymbol{x}}^i) \leq 0, k \in \widetilde{J}_i^i$, and $h_k(\boldsymbol{u}^i, \widetilde{\boldsymbol{x}}^i) = 0, k \in \widetilde{J}_e^i$, that determine subproblem i's local feasible set $\widetilde{\mathcal{F}}^i(\boldsymbol{u}^i)$, cf. (5.22). Only values $\boldsymbol{u}_k^i$ for which $\widetilde{\mathcal{F}}^i(\boldsymbol{u}^i) \neq \emptyset$ should be taken into account. Since a wide approximation domain can lead to a locally bad approximation of ϕ_i, it can be worthwhile to further truncate the domain based on individual information or a reasonable idea about the order of magnitude, in which the value of $\boldsymbol{u}^i$ should lie in an optimal solution of the overall problem.
In case of dual decomposition, in general, every point in $(\mathbb{R}_0^+)^{d_{li}^i} \times \mathbb{R}^{d_{le}^i}$ can be chosen as data point $(\boldsymbol{s}_k^i, \boldsymbol{t}_k^i)$. However, due to the interpretability of dual variables as shadow

prices, there are sometimes natural bounds on the links by the model context, or the values of dual variables in an optimal solution can at least be restricted by reasonability. Regarding the numerical solution of a subproblem i for each parameter $(\boldsymbol{u}_k^i),\ k \in \{1, ..., K\}$, or $(\boldsymbol{s}_k^i, \boldsymbol{t}_k^i),\ k \in \{1, ..., K\}$, it can be efficient to solve these parametric problems at least partly sequentially and to use already computed optimal solutions of problems as start values for subsequent optimizations of problems in adjacent parameters. If the optimal value sensitivities of a subproblem are available, they might be used for an adaptive grid refinement.

Selecting a parametric model:
Since ϕ_i is convex and d_i is concave, a natural way, which requires only few parameters, is to model these functions by a multivariate quadratic polynomial. Principally, also a linear model, a polynomial of higher degree or a different model can be used. A suitable choice strongly depends on the application. For the purpose of an efficient numerical optimization, a tabulation of the calculated optimal values of a subproblem and local interpolations during the optimization procedure of the master problem might be preferable to the estimation of a global model with only a few number of parameters.

Determining the parameters:
An estimator $\hat{\theta}$ of the parameter θ can be obtained for example by a least squares estimation, cf. section 3.7. If one aims to compute a lower or an upper bound on the optimal solution of the master problems (5.24) and (5.25), the functions $\widehat{\phi}_i$ and $-\widehat{d}_i$ should underestimate or overestimate ϕ_i and $-d_i$, respectively. The former may be achieved by constraining the corresponding least squares estimation by the conditions $\phi_i^\theta(\boldsymbol{u}_k^i) \leq \phi_i(\boldsymbol{u}_k^i)\ \forall\ k \in \{1, ..., K\}$ and $d_i^\theta(\boldsymbol{s}_k^i, \boldsymbol{t}_k^i) \geq d_i(\boldsymbol{s}_k^i, \boldsymbol{t}_k^i)\ \forall\ k \in \{1, ..., K\}$, respectively.[9] Overestimators can be obtained analogously.

Remark 5.3.2. Approximating optimal value functions of subproblems is in principle also possible in the case of a mixed primal-dual decomposition, cf. 5.2.10. However, due to the two-level structure of the resulting master problem, it is in general not possible to compute lower or upper bounds on the optimal solution of the overall problem by underestimating or overestimating submodel functions.

Remark 5.3.3. (Sensitivity analysis within subproblems) Solving a subproblem for varying values $\boldsymbol{u}_k^i$, or $(\boldsymbol{s}_k^i, \boldsymbol{t}_k^i)$, $k \in \{1, ..., K\}$, which cover the relevant domain of the subproblems' optimal value function, generates, besides the optimal values $\phi_i(\boldsymbol{u}_k^i)$, or $d_i(\boldsymbol{s}_k^i, \boldsymbol{t}_k^i)$, further interesting data. These are the optimal solutions and, usually, also their sensitivities with respect to model parameters. Under the appropriate regularity assumptions, sensitivities are given through theorem 3.2.11. The matrices that are necessary to compute sensitivities are calculated by many numerical optimization algorithms while iterating. Firstly, considering the optimal solutions of a subproblem dependent

[9] [EFS12] obtains promising results in view of global optimal solutions of MINLPs by using underestimators for the optimal value functions of selected subsystems.

on $\boldsymbol{u}^i$, or $(\boldsymbol{s}^i, \boldsymbol{t}^i)$, we obtain an overview of possible subproblem solutions. Secondly, let us consider an optimization problem that includes a large number of parameters. The availability of subproblem sensitivities with respect to these parameters for all relevant values of $\boldsymbol{u}^i$, or $(\boldsymbol{s}^i, \boldsymbol{t}^i)$, makes an extended sensitivity analysis possible, which facilitates the differentiation between influential parameters and parameters that have only little effect on the overall solution.

6 Decomposing a Multi-Commodity Supply-Demand Network

In chapter 2, we introduced a constrained nonlinear optimization model for the simulation of price formation in a multi-commodity market, at which products are connected by production processes. Depending on the market that one simulates, this model can become large and include many parameters. Besides data for the modeling of the demand-price relationship of products with external demand, prices of external products and data about production and processing facilities are required. Since real market data is often hardly available or costly, data procurement can be difficult. To facilitate the task of data procurement, a classification of parameters into more and less important ones is useful. Furthermore, the interpretation of optimization results is complicated in a large model since major dependencies are not self-evident. Therefore, this chapter aims at developing a suitable method for reducing the complexity of a large multi-commodity market model. Specifically, we intend to

- structure the overall problem by determining weakly connected market components,
- aggregate components that are of less interest than other ones regarding a certain analysis,
- differentiate between parameters that strongly influence simulation results and those that have little to no effect.

The reduction of network models through the aggregation of components is accomplished in several scientific fields. In the context of operations research, for example, [Stu02] proposes a graph-theoretical approach to reduce a model for the technological oriented simulation of energy systems. Furthermore, [SRWD$^+$11] presents a method for a structure-preserving reduction of large-scale logistic networks based on the *Log-Rank* as measure for the importance of nodes. Reduction methods that are specifically tailored to the optimization of natural gas transmission networks were published by [RMWSB02, MHH$^+$04, BSH11]. They operate to some extent automatically. Gas transmission networks differ from our production network mainly by the crucial fact that, in our model, commodities are transformed while "flowing" through the network. Moreover, network reduction is accomplished in the scientific area of molecular biology: for example, [BCFK08, CSRS$^+$06] deal with an automated domain-oriented reduction approach for networks of signaling proteins, which provides a macroscopic description of network dynamics. However, despite the variety of existing reduction methods, there is,

as far as we know, no reduction method that is applicable to a constrained nonlinear network optimization model as ours, presented in chapter 2. Therefore, in chapter 5, we developed a general approach to the decomposition of constrained nonlinear programs with a block-separable structure. This approach reveals model components that are particularly suited for aggregation.
In this chapter, we apply a tailored version of our reduction approach of chapter 5 to the network optimization model of chapter 2. The block-separable structure of this model is due to its separable objective function and the products connected by chemical reaction chains. We detach branches of the production network that are linked to the remaining part of the network by only a few products called links. Within these branches, we maximize profit (or minimize costs) given chosen input (or output) quantities of the linking products. This enables us to approximate subproblems' optimal value functions, which represent the input/output-profit/cost relationship for a whole subsystem so that we can treat it like a single consumption sector or the aggregated costs of some preceding production processes, respectively.
This approach is advantageous because it divides the overall optimization problem into subproblems with less variables and constraints. Only few variables are necessary to describe the connection between subproblems and the main problem. Thus, the subproblems can be solved nearly independently from the remaining network. Once solved and aggregated, the subproblems' results can be used for various simulations of the overall problem under the sole assumption that the subsystem parameters do not vary dramatically. Accomplishing sensitivity analyses within subsystems over the whole range of possible input/output quantities yields global sensitivity results with respect to parameters outside the subsystem. The approach facilitates the interpretation of the simulation results, decreases the simulation time, and enables to differentiate between more and less influencing parameter data.
Section 6.1 defines the version of the multi-commodity model on which we focus in this chapter. Section 6.2 adapts the decomposition approach of the previous chapter to the model of section 6.1 and formulates resulting subproblems. In section 6.3, we present first numerical results of aggregations within a small consistent production system. Furthermore, we extend this production system to a larger model. Section 6.4 deals with the automatic identification of subsystems that are suited for an aggregation. Our developed algorithm is applied to the extended commodity market. In section 6.5, we discuss the aggregation of subsystems in terms of approximating the subproblems' optimal value functions. This discussion includes the definition of the relevant approximation domain, the properties of the optimal value functions, the choice of an appropriate parametric model, and an alternative formulation of the subproblem, which can be used to generate subsystem data. Section 6.6 addresses sensitivity analysis within the overall model and within subsystems. We explain why sensitivity analysis may justify model simplifications and how it supports the task of data procurement. Section 6.7 includes numerical results of subsystems' aggregations within the extended market model. Sensitivity analyses within the subsystems reveal several possibilities for model simplifications.

6.1 The Model

Given the definitions of section 2.1, we consider a special case of the optimization problem defined in section 2.2 by making the following assumptions/simplifications (in parts they were already stated for the example in section 2.5):

- We assume that all consumers have the same demand behavior. By remark 2.1.3, this means that we must consider only one consumer. Therefore, we omit the consumer index c.
- We consider only one time period and one region, i.e., we omit also the indices t and r. Thus, there are no storage facilities.
- Fixed costs for running the plants are neglected so that we do not need to consider plants, but we can attribute the aggregated capacities of plants that run a certain process to this process, i.e., we define $\forall\ s \in \mathcal{S}:\ a_s^{\text{cap}} := \sum_{i \in \mathcal{I}} a_{i,s}^{\text{cap}}$ and respective decision variables x_s^{q}.
- Assuming that the supplier offers his whole production output at the market, prices are determined by these quantities through the demand-price relationship (6.2). This means, we can leave the price variables out and end up only with the variables for production quantities.

The Price-Demand Relationship

By the above simplifications, the demand function (2.3) is given by

$$\phi_{p_{out}}(x^{\pi}_{p_{out}}) = \max\left[a^1_{p_{out}} \cdot \tanh\Big(\frac{a^2_{p_{out}} + w_{p_{out}}(a^{GDP}, a^{IndPro}) - x^{\pi}_{p_{out}}}{a^3_{p_{out}}}\Big), 0\right], \tag{6.1}$$

where $a^1_{p_{out}}, a^2_{p_{out}}$, and $a^3_{p_{out}}$ are defined by (2.2) and $w_{p_{out}}(a^{GDP}, a^{IndPro})$ by equation (2.4). Inverting (6.1), we can construct the corresponding demand-price relation

$$P_{p_{out}}(x^{\text{sales}}_{p_{out}}) = \max\left[a^2_{p_{out}} + w_{p_{out}}(a^{GDP}, a^{IndPro}) - a^3_{p_{out}} \cdot \operatorname{artanh}\Big(\frac{x^{\text{sales}}_{p_{out}}}{a^1_{p_{out}}}\Big), 0\right]. \tag{6.2}$$

In the domain of positive function values, i.e., for $0 \le x^{\text{sales}}_{p_{out}} < \phi_{p_{out}}(0)$, where $\phi_{p_{out}}(0) < a^1_{p_{out}}$, we can neglect the maximum-operator and the first and second derivatives of $P_{p_{out}}$ are given by

$$P'_{p_{out}}(x^{\text{sales}}_{p_{out}}) = \frac{-a^3_{p_{out}} \cdot a^1_{p_{out}}}{{a^1_{p_{out}}}^2 - {x^{\text{sales}}_{p_{out}}}^2} \tag{6.3}$$

and

$$P''_{p_{out}}(x^{\text{sales}}_{p_{out}}) = \frac{-2 \cdot a^3_{p_{out}} \cdot a^1_{p_{out}} \cdot x^{\text{sales}}_{p_{out}}}{({a^1_{p_{out}}}^2 - {x^{\text{sales}}_{p_{out}}}^2)^2}. \tag{6.4}$$

Remark 6.1.1. The following decomposition approach is applicable in the same manner using any other price-demand relationsship that leads to a convex optimization problem (6.5). We propose to use a function $P_{p_{out}} : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ with the following properties:

- $P_{p_{out}}$ is continuous, decreasing, and bounded,
- $P_{p_{out}}$ has a minimal zero, say y_0, and $P_{p_{out}}(y) = 0$ for $y \in [y_0, \infty]$,
- $P_{p_{out}}$ is twice continuously differentiable on $]0, y_0[$ with $P'_{p_{out}}(y) < 0$ and $P''_{p_{out}}(y) < 0$.

The Optimization Problem

Under the above assumptions, we consider the following nonlinear optimization problem with capacity bounds on the variables and non-negative gross production for products with external demand and all intermediates:

$$\max_{\boldsymbol{x}^{\mathrm{q}}} F(\boldsymbol{x}^{\mathrm{q}}) = \max_{\boldsymbol{x}^{\mathrm{q}}} \sum_{p_{out} \in \mathcal{P}_{out}} \Big(\sum_{s \in \mathcal{S}} x_s^{\mathrm{q}} \cdot a_{s,p_{out}}^{\mathrm{f}} \Big) \cdot P_{p_{out}} \Big(\sum_{s \in \mathcal{S}} x_s^{\mathrm{q}} \cdot a_{s,p_{out}}^{\mathrm{f}} \Big) + \sum_{\substack{p_{ex} \in \mathcal{P}_{ex}, \\ s \in \mathcal{S}}} x_s^{\mathrm{q}} \cdot a_{s,p_{ex}}^{\mathrm{f}} \cdot a_{p_{ex}}^{\pi} \tag{6.5a}$$

such that

$$\forall\ s \in \mathcal{S}: \qquad 0 \le x_s^{\mathrm{q}} \le a_s^{\mathrm{cap}} \tag{6.5b}$$

$$\forall\ p \in \mathcal{P}_{out} \dot{\bigcup} \mathcal{P}_{mid}: \qquad 0 \le \sum_{s \in \mathcal{S}} x_s^{\mathrm{q}} \cdot a_{s,p}^{\mathrm{f}}. \tag{6.5c}$$

Remark 6.1.2. (Differentiability and convexity) To assure differentiability and convexity of this problem, one can add as further constraints

$$\sum_{s \in \mathcal{S}} x_s^{\mathrm{q}} \cdot a_{s,p_{out}}^{\mathrm{f}} \le \phi_{p_{out}}(0) \ \ (< a_{p_{out}}^{1}) \qquad \forall\ p_{out} \in \mathcal{P}_{out}. \tag{6.6}$$

Compare also remark 2.2.1 and 2.5.1 concerning the maximum function in the demand function and the price-demand relationship, respectively. However, in most applications that only include one consumer, such constraints are not active since producing a product whose price is zero is unprofitable. It would be economically reasonable only if this product was a byproduct of a process whose primary output is in strong demand and, therefore, compensates for the profitless byproduct.
To show that the above optimization problem is convex, by remark 3.1.4, definition 3.1.5 and example 3.1.7, we have to prove that $\big(\sum_{s \in \mathcal{S}} x_s^{\mathrm{q}} \cdot a_{s,p_{out}}^{\mathrm{f}} \big) \cdot P_{p_{out}} \big(\sum_{s \in \mathcal{S}} x_s^{\mathrm{q}} \cdot a_{s,p_{out}}^{\mathrm{f}} \big)$ is concave $\forall\ p_{out} \in \mathcal{P}_{out}$. Then, convexity follows since the second part of the objective function as well as the constraints of problem (6.5) are linear, and since sums of convex (concave) functions are again convex (concave).
We have for $G_{p_{out}}(y) := y \cdot P_{p_{out}}(y)$:

$$G'_{p_{out}}(y) = P_{p_{out}}(y) + y \cdot P'_{p_{out}}(y) \tag{6.7}$$

$$G''_{p_{out}}(y) = 2P'_{p_{out}}(y) + y \cdot P''_{p_{out}}(y) \tag{6.8}$$

$P'_{p_{out}}(y) < 0$ and $P''_{p_{out}}(y) < 0 \; \forall \; y \in]0, \phi_{p_{out}}(0)[, \; p_{out} \in \mathcal{P}_{out}$, gives $G''_{p_{out}}(y) < 0 \; \forall \; y \in]0, \phi_{p_{out}}(0)[, \; p_{out} \in \mathcal{P}_{out}$, which means that $G_{p_{out}}(y)$ is strictly concave. Furthermore, with $\boldsymbol{x}^\theta := (1-\theta)\boldsymbol{x}^a + \theta \boldsymbol{x}^b$, $\theta \in [0,1]$, we have

$$\sum_{s=1}^{n} x_s^\theta \cdot a_s^{\mathrm{f}} = (1-\theta) \sum_{s=1}^{n} x_s^a \cdot a_s^{\mathrm{f}} + \theta \sum_{s=1}^{n} x_s^b \cdot a_s^{\mathrm{f}}. \tag{6.9}$$

In summary, this yields with $y^\theta := (1-\theta)y^a + \theta y^b := (1-\theta)\sum_{s=1}^{n} x_s^a \cdot a_s^{\mathrm{f}} + \theta \sum_{s=1}^{n} x_s^b \cdot a_s^{\mathrm{f}}$

$$\begin{aligned}
&\big(\sum_{s\in\mathcal{S}} x_s^\theta \cdot a_s^{\mathrm{f}}\big) \cdot P_{p_{out}}\big(\sum_{s\in\mathcal{S}} x_s^\theta \cdot a_s^{\mathrm{f}}\big) = G_{p_{out}}(y^\theta) \\
&\geq (1-\theta) G_{p_{out}}(y^a) + \theta G_{p_{out}}(y^b) \\
&= (1-\theta)\big(\sum_{s\in\mathcal{S}} x_s^a \cdot a_s^{\mathrm{f}}\big) \cdot P_{p_{out}}\big(\sum_{s\in\mathcal{S}} x_s^a \cdot a_s^{\mathrm{f}}\big) + \theta\big(\sum_{s\in\mathcal{S}} x_s^b \cdot a_s^{\mathrm{f}}\big) \cdot P_{p_{out}}\big(\sum_{s\in\mathcal{S}} x_s^b \cdot a_s^{\mathrm{f}}\big).
\end{aligned} \tag{6.10}$$

However, since $\boldsymbol{x}^a \neq \boldsymbol{x}^b$ does not imply $y^a \neq y^b$, our optimization problem is in general not strictly convex.

To apply the reduction approach proposed in chapter 5 to problem (6.5), we introduce additional variables and constraints so that we obtain a separable optimization problem with coupling constraints:

Production optimization problem in a separable formulation

To make the objective function of problem (6.5) separable, we must introduce additional variables $x_p^{\text{net\_prod}}$, describing the aggregated production, at least for all products $p \in \mathcal{P}_{out}$ because of the nonlinear price-demand functions $P_{p_{out}}, p_{out} \in \mathcal{P}_{out}$. This leads to a model in the variables x_s^{q} $(s \in \mathcal{S})$ and $x_p^{\text{net\_prod}}$ $(p \in \mathcal{P}_{out})$:

$$\max_{\boldsymbol{x}^{\mathrm{q}}, \boldsymbol{x}^{\text{net\_prod}}} \sum_{p_{out}\in\mathcal{P}_{out}} x_{p_{out}}^{\text{net\_prod}} \cdot P_{p_{out}}\big(x_{p_{out}}^{\text{net\_prod}}\big) + \sum_{p_{ex}\in\mathcal{P}_{ex}, s\in\mathcal{S}} x_s^{\mathrm{q}} \cdot a_{s,p_{ex}}^{\mathrm{f}} \cdot a_{p_{ex}}^{\pi} \tag{6.11a}$$

such that

$$\begin{aligned}
&\forall \; s \in \mathcal{S}: && 0 \leq x_s^{\mathrm{q}} \leq a_s^{\mathrm{cap}} \\
&\forall \; p \in \mathcal{P}_{mid}: && 0 \leq \sum_{s\in\mathcal{S}} x_s^{\mathrm{q}} \cdot a_{s,p}^{\mathrm{f}} \;\; \rightarrow \text{coupling constraints (dual variables } \lambda_{p_{mid}}) \\
&\forall \; p \in \mathcal{P}_{out}: && 0 \leq x_p^{\text{net_prod}} \\
&\forall \; p \in \mathcal{P}_{out}: && 0 = x_p^{\text{net_prod}} - \sum_{s\in\mathcal{S}} x_s^{\mathrm{q}} \cdot a_{s,p}^{\mathrm{f}} \\
& && \quad \rightarrow \text{coupling constraints (dual variables } \lambda_{p_{out}})
\end{aligned} \tag{6.11b}$$

Relaxing the coupling constraints, we obtain a partial dual function of this problem:

$$
\begin{aligned}
d(\boldsymbol{\lambda}_{\mathcal{P}_{out}}, \boldsymbol{\lambda}_{\mathcal{P}_{mid}}) = \inf_{\substack{\boldsymbol{x}^{\mathrm{q}}, \boldsymbol{x}^{\mathrm{net\_prod}}, \\ 0 \le x_s^{\mathrm{q}} \le a_s^{\mathrm{cap}}, \\ 0 \le x_p^{\mathrm{net\_prod}}}} & - \sum_{p_{out} \in \mathcal{P}_{out}} x_{p_{out}}^{\mathrm{net\_prod}} \cdot P_{p_{out}}\left(x_{p_{out}}^{\mathrm{net\_prod}}\right) - \sum_{\substack{p_{ex} \in \mathcal{P}_{ex}, \\ s \in \mathcal{S}}} x_s^{\mathrm{q}} \cdot a_{s,p_{ex}}^{\mathrm{f}} \cdot a_{p_{ex}}^{\pi} \\
& + \sum_{p_{out} \in \mathcal{P}_{out}} \lambda_{p_{out}} \cdot \left(x_{p_{out}}^{\mathrm{net\_prod}} - \sum_{s \in \mathcal{S}} x_s^{\mathrm{q}} \cdot a_{s,p}^{\mathrm{f}}\right) \\
& - \sum_{p_{mid} \in \mathcal{P}_{mid}, s \in \mathcal{S}} \lambda_{p_{mid}} \cdot x_s^{\mathrm{q}} \cdot a_{s,p_{mid}}^{\mathrm{f}} \\
= \inf_{\substack{\boldsymbol{x}^{\mathrm{q}}, \boldsymbol{x}^{\mathrm{net\_prod}}, \\ 0 \le x_s^{\mathrm{q}} \le a_s^{\mathrm{cap}}, \\ 0 \le x_p^{\mathrm{net\_prod}}}} & - \sum_{p_{out} \in \mathcal{P}_{out}} x_{p_{out}}^{\mathrm{net\_prod}} \cdot \left(P_{p_{out}}\left(x_{p_{out}}^{\mathrm{net\_prod}}\right) - \lambda_{p_{out}}\right) \\
& - \sum_{s \in \mathcal{S}} x_s^{\mathrm{q}} \cdot \Big(\sum_{p \in \mathcal{P}_{out} \bigcup \mathcal{P}_{mid}} a_{s,p}^{\mathrm{f}} \cdot \lambda_p + \sum_{p_{ex} \in \mathcal{P}_{ex}} a_{s,p_{ex}}^{\mathrm{f}} \cdot a_{p_{ex}}^{\pi} \Big)
\end{aligned}
\tag{6.12}
$$

Here, primal variables are blue-colored and dual variables are magenta-colored. This partial dual function d decomposes into $|\mathcal{P}_{out}| + |\mathcal{S}|$ subproblems, of which each is in one primal variable. The dimension of d's domain is $|\mathcal{P}_{out}| + |\mathcal{P}_{mid}|$.

6.2 The Approach

Aiming to analyze the structure of problem (6.5) by the approach of chapter 5, we start with a graphical representation of connections between primal variables $x_s^{\mathrm{q}}, s \in \mathcal{S}$, and $x_p^{\mathrm{net\_prod}}, p \in \mathcal{P}_{out}$, and dual variables $\lambda_{p_{out}}, p_{out} \in \mathcal{P}_{out}$, and $\lambda_{p_{mid}}, p_{mid} \in \mathcal{P}_{mid}$. The required information for the graphical representation, described in section 5.1, can, for example, be extracted from the partial dual function (6.12). Basically, we have one node for each primal variable and one for each dual variable. However, there are processes $s_i, s_j \in \mathcal{S}$ including exactly the same products $p \in \mathcal{P}_{ij} \subset \mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$, but with possibly different input/output factors, i.e., $a_{s_i,p}^{\mathrm{f}} \neq 0, a_{s_j,p}^{\mathrm{f}} \neq 0 \ \forall \ p \in \mathcal{P}_{ij}$ and $a_{s,p}^{\mathrm{f}} = 0 \ \forall \ p \in \left(\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}\right) \setminus \mathcal{P}_{ij}$. Such processes are represented by the same node of the graph. In principle, there could also be products $p \in \mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$ appearing in exactly the same processes, but this case does not occur in our commodity market data, and we do not take it into account. Therefore, the bipartite graph G_b has a structure as shown in figure 6.1.

Let us first consider the connections between the primal process variables $x_s^{\mathrm{q}}, s \in \mathcal{S}$, and the dual variables $\lambda_p, p \in \mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$. Assuming that the input/output factors $a_{s,p}^{\mathrm{f}}$ of processes, which include exactly the same products, i.e., which are represented by the same node of G_b, are of the same sign with respect to a certain product, we equip the corresponding edges of G_b with an arrow according to the sign of $a_{s,p}^{\mathrm{f}}$. If $a_{s,p}^{\mathrm{f}} > 0$, i.e., if p is an output of process s, the arrow points at the product (dual variable λ_p), and

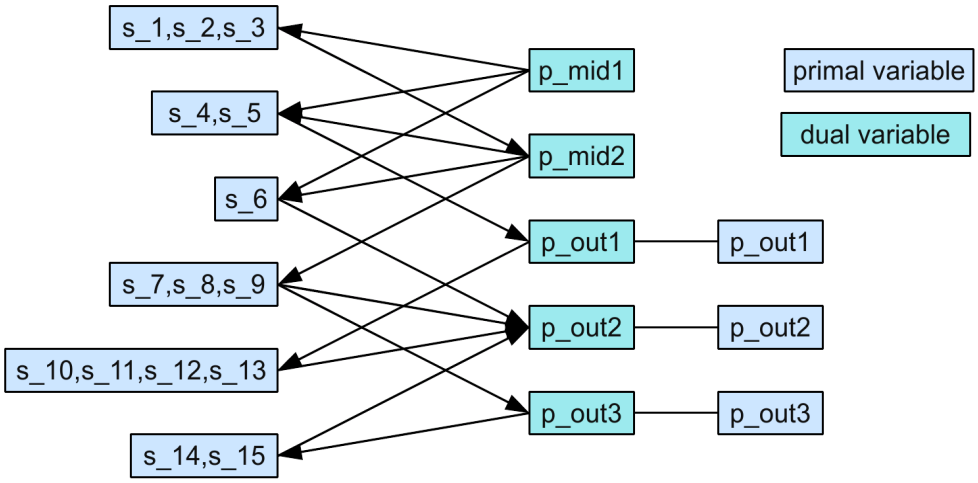

Figure 6.1: *Basic structure of the production optimization problem. All variables are represented by their indices. A blue node with multiple process indices represents the production variables of processes that include the same products, but possibly with distinct input/output factors.*

if $a^{\mathrm{f}}_{s,p} < 0$, i.e., if p is an input of process s, the arrow points at the process (primal variable x^{q}_s). Now, let us consider the connections of primal net production variables $x_p^{\mathrm{net\_prod}}, p \in \mathcal{P}_{out}$, and the dual variables $\lambda_p, p \in \mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$. Each of these primal variables $x_p^{\mathrm{net\_prod}}, p \in \mathcal{P}_{out}$, is connected to its respective dual variable $\lambda_p, p \in \mathcal{P}_{out}$. This is why, in figure 6.1, we choose a graphical representation of G_b that slightly differs from the usual bipartite graph representation. Because of the described unique relation, we do not need to regard the second set of primal variables, $x_p^{\mathrm{net\_prod}}, p \in \mathcal{P}_{out}$, in our following connectivity analysis.
In commodity markets, usually, the number of different processes, $|\mathcal{S}|$, is much larger than the number of relevant products, $|\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}|$, and there are mostly several processes including similar products. Thus, aiming for a small link degree, cf. definition 5.3.1, let us consider the dual variables $\lambda_p, p \in \mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$, corresponding to coupling constraints as possible links of a decomposition. We search for small subsets $\mathcal{P}_{\mathrm{link}} \subset \mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$ such that the relaxation of the respective coupling constraints cuts off a subproblem from the overall optimization problem (6.5). In the following tailored approach, however, we will not relax the chosen coupling constraints and use the respective dual variables as links, but we will include additional variables in the chosen coupling constraints such that they decompose. By this, we are able to formulate the subproblems and interprete the results without requiring duality theory. Considering the respective dual solutions optionally is advantageous with respect to the interpretation of the overall results because of the economical meaning of a dual variable λ_p as shadow price of the related product p. We will discuss this in section 6.5.

The Restatement of the Optimization Problem

Suppose that we determined a small, nonempty set of links $\mathcal{P}_{\text{link}} \subset \mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$ and a related nonempty set of submodel products $\mathcal{P}_{\text{b}} \subset \mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$, where $\mathcal{P}_{\text{link}} \bigcap \mathcal{P}_{\text{b}} = \emptyset$, $\mathcal{P}_{\text{b}} \bigcup \mathcal{P}_{\text{link}} \subset \mathcal{P}_{mid} \bigcup \mathcal{P}_{out}$, and no process of $\mathcal{S}$ includes both, a product of $\mathcal{P}_{\text{b}}$ and a product of $(\mathcal{P}_{mid} \bigcup \mathcal{P}_{out}) \setminus (\mathcal{P}_{\text{b}} \bigcup \mathcal{P}_{\text{link}})$, i.e., $\nexists\, s \in \mathcal{S}, p_1 \in \mathcal{P}_{\text{b}}, p_2 \in (\mathcal{P}_{mid} \bigcup \mathcal{P}_{out}) \setminus (\mathcal{P}_{\text{b}} \bigcup \mathcal{P}_{\text{link}}) : a^{\text{f}}_{s,p_1} \neq 0 \vee a^{\text{f}}_{s,p_2} \neq 0$. This situation is sketched in figure 6.2.[1]
Let $\mathcal{S}_{\text{b}} \subset \mathcal{S}$ be a set of processes that defines together with the products $\mathcal{P}_{\text{b}}$ a submodel with links $\mathcal{P}_{\text{link}}$. In general, $\mathcal{S}_{\text{b}}$ is not uniquely determined by $\mathcal{P}_{\text{b}}$. It must contain at least all processes that include products of the set $\mathcal{P}_{\text{b}}$ and must not contain processes including products of $\mathcal{P} \setminus (\mathcal{P}_{\text{b}} \bigcup \mathcal{P}_{\text{link}})$. A method for an automated identification of subproblems that are coupled to the remaining network by only few links is detailed in section 6.4. There, we also explain how the set $\mathcal{S}_{\text{b}}$ can be determined. In accordance with definition 5.3.1, we call the number of linking products, $|\mathcal{P}_{\text{link}}|$, the link degree of a submodel of products $\mathcal{P}_{\text{b}}$ and processes $\mathcal{S}_{\text{b}}$.[2]
To reduce complexity, we separate the subproblem of optimizing the production corresponding to the processes $s_{\text{b}} \in \mathcal{S}_{\text{b}}$ from the overall optimization. To preserve the connection of the problems, we augment the subproblem by additional constraints, namely certain fixed input or output quantities of the linking products $p_{\text{link}} \in \mathcal{P}_{\text{link}}$. The subproblem is solved n-times for different input/output bounds $\boldsymbol{y}^{\text{q}}_1, ..., \boldsymbol{y}^{\text{q}}_n \in D \subset \mathbb{R}^{|\mathcal{P}_{\text{link}}|}$. The results of these subsystem optimizations are later included in the optimization of the

[1] The shown graph corresponds to a projection of a bipartite graph, similar to the one in figure 6.1, onto its set of dual variables $\lambda_p, p \in \mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$.

[2] In general, one can also allow $\mathcal{P}_{\text{b}} = \emptyset$. However, this case is only useful if there is a reasonable number of processes including only linking products and external products. An example of $\mathcal{P}_{\text{b}} = \emptyset$ is given by example 1.3 of section 6.3. A subsystem with $\mathcal{P}_{\text{b}} = \emptyset$ leads to a linear subproblem. Similarly, in principle, a subsystem aggregation is possible if $\mathcal{P}_{\text{b}} \bigcup \mathcal{P}_{\text{link}} = \mathcal{P}_{mid} \bigcup \mathcal{P}_{out}$. However, we focus on more balanced decompositions with $\mathcal{P}_{\text{b}} \neq \emptyset$ and $\mathcal{P}_{\text{b}} \bigcup \mathcal{P}_{\text{link}} \subset \mathcal{P}_{mid} \bigcup \mathcal{P}_{out}$.

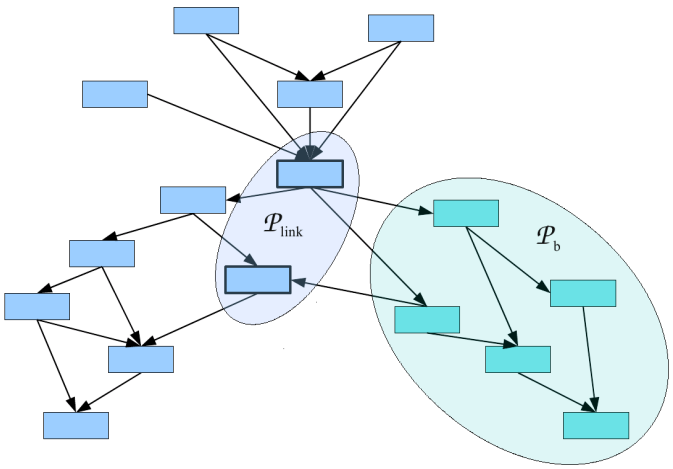

Figure 6.2: *System before reduction*

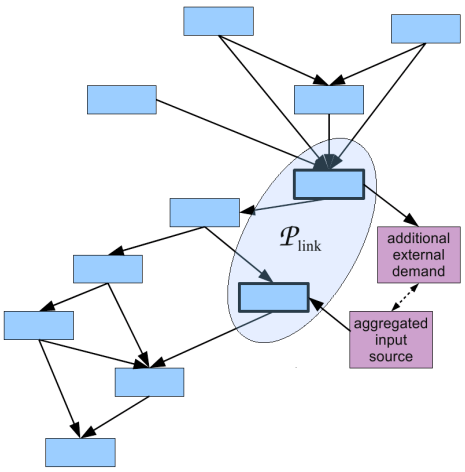

Figure 6.3: *System after reduction*

remaining part of the network through a function that approximates the input/output-profit/cost relationship of the subsystem, cf. figure 6.3.
In summary, a solution of the overall problem (6.5) can be approximated in three steps:

1. Solve the subproblem of processes $\mathcal{S}_\mathrm{b}$ for varying constraints $\boldsymbol{y}_1^\mathrm{q}, ..., \boldsymbol{y}_n^\mathrm{q}$.

2. With the results of step 1, estimate a joint input/output-profit/cost function $c : D \to \mathbb{R},\ \boldsymbol{y}^\mathrm{q} \mapsto c(\boldsymbol{y}^\mathrm{q})$ for the linking products $p_\mathrm{link} \in \mathcal{P}_\mathrm{link}$, where $D \subset \mathbb{R}^{|\mathcal{P}_\mathrm{link}|}$.

3. Solve the optimization problem of the remaining processes $s \in \mathcal{S} \setminus \mathcal{S}_\mathrm{b}$ taking into account the approximating input/output-profit/cost function c of the subsystem.

Step 1 is accomplished by solving for each parameter vector $\boldsymbol{y}_i^\mathrm{q} = (y_{p_1,i}^\mathrm{q}, ..., y_{p_{|\mathcal{P}_\mathrm{link}|},i}^\mathrm{q})^\mathrm{T}$, $i = 1, ..., n$,

$$\max_{\boldsymbol{x}_{\mathcal{S}_\mathrm{b}}^\mathrm{q}} F_1(\boldsymbol{x}_{\mathcal{S}_\mathrm{b}}^\mathrm{q}) = \max_{\boldsymbol{x}_{\mathcal{S}_\mathrm{b}}^\mathrm{q}} \sum_{\substack{p_{out} \in \\ \mathcal{P}_{out} \bigcap \mathcal{P}_\mathrm{b}}} \Big(\sum_{s \in \mathcal{S}_\mathrm{b}} x_s^\mathrm{q} \cdot a_{s,p_{out}}^\mathrm{f} \Big) \cdot P_{p_{out}} \Big(\sum_{s \in \mathcal{S}_\mathrm{b}} x_s^\mathrm{q} \cdot a_{s,p_{out}}^\mathrm{f} \Big) + \sum_{\substack{p_{ex} \in \mathcal{P}_{ex}, \\ s \in \mathcal{S}_\mathrm{b}}} x_s^\mathrm{q} \cdot a_{s,p_{ex}}^\mathrm{f} \cdot a_{p_{ex}}^\pi \tag{6.13a}$$

such that

$$\forall\ s \in \mathcal{S}_\mathrm{b} : \qquad 0 \le x_s^\mathrm{q} \le a_s^\mathrm{cap} \tag{6.13b}$$

$$\forall\ p \in (\mathcal{P}_{out} \dot{\bigcup} \mathcal{P}_{mid}) \bigcap \mathcal{P}_\mathrm{b} : \quad 0 \le \sum_{s \in \mathcal{S}_\mathrm{b}} x_s^\mathrm{q} \cdot a_{s,p}^\mathrm{f} \tag{6.13c}$$

$$\forall\ p_\mathrm{link} \in \mathcal{P}_\mathrm{link} : \qquad 0 \le y_{p_\mathrm{link},i}^\mathrm{q} + \sum_{s \in \mathcal{S}_\mathrm{b}} x_s^\mathrm{q} \cdot a_{s,p_\mathrm{link}}^\mathrm{f}. \tag{6.13d}$$

Step 2 consists in fitting a function $c : D \to \mathbb{R},\ \boldsymbol{y}^\mathrm{q} \mapsto c(\boldsymbol{y}^\mathrm{q})$, $D \subset \mathbb{R}^{|\mathcal{P}_\mathrm{link}|}$, to the data $(\boldsymbol{y}_i^\mathrm{q}, \pi_i)_{i=1,...,n}$, where π_i is the optimal objective value of problem (6.13) given the parameter vector $\boldsymbol{y}_i^\mathrm{q}$. Then, by definition 3.2.1, c is an approximation of the optimal value function F_1^* of the parametric problem (6.13). In the small example of section 6.3, we will use different parametric models for this curve fitting and estimate them by the method of least squares, cf. section 3.7. In a large model, it is not obvious how $D \subset \mathbb{R}^{|\mathcal{P}_\mathrm{link}|}$ and the quantities $\boldsymbol{y}_1^\mathrm{q}, ..., \boldsymbol{y}_n^\mathrm{q} \in D$ are chosen advantageously. We discuss the whole task of a submodel approximation in a detailed way in section 6.5.

Step 3 comprises the solution of the following optimization problem, given an estimated input/output-profit/cost function c. This problem approximates the original optimiza-

tion problem (6.5):

$$\begin{aligned}
&\max_{\substack{\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}},\\ \boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_{\mathrm{link}}}\in D}} F_2(\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}}, \boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_{\mathrm{link}}}) \\
&\quad = \max_{\substack{\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}},\\ \boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_{\mathrm{link}}}\in D}} \sum_{\substack{p_{out}\in\\ \mathcal{P}_{out}\setminus(\mathcal{P}_{\mathrm{b}}\bigcup\mathcal{P}_{\mathrm{link}})}} \Big(\sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}} x^{\mathrm{q}}_s\cdot a^{\mathrm{f}}_{s,p_{out}}\Big)\cdot P_{p_{out}}\Big(\sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}} x^{\mathrm{q}}_s\cdot a^{\mathrm{f}}_{s,p_{out}}\Big) \\
&\qquad + \sum_{\substack{p_{ex}\in\mathcal{P}_{ex},\\ s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}}} x^{\mathrm{q}}_s\cdot a^{\mathrm{f}}_{s,p_{ex}}\cdot a^{\pi}_{p_{ex}} + \sum_{p_{\mathrm{link}}\in\mathcal{P}_{\mathrm{link}}\bigcap\mathcal{P}_{out}} x^{\mathrm{sales}}_{p_{\mathrm{link}}}\cdot P_{p_{\mathrm{link}}}(x^{\mathrm{sales}}_{p_{\mathrm{link}}}) + c(\boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_{\mathrm{link}}})
\end{aligned} \tag{6.14a}$$

such that

$$\forall\ s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}: \qquad 0\le x^{\mathrm{q}}_s\le a^{\mathrm{cap}}_s \tag{6.14b}$$

$$\forall\ p\in(\mathcal{P}_{out}\dot{\bigcup}\mathcal{P}_{mid})\setminus(\mathcal{P}_{\mathrm{b}}\dot{\bigcup}\mathcal{P}_{\mathrm{link}}): \qquad 0\le\sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}} x^{\mathrm{q}}_s\cdot a^{\mathrm{f}}_{s,p} \tag{6.14c}$$

$$\forall\ p_{\mathrm{link}}\in\mathcal{P}_{\mathrm{link}}\bigcap\mathcal{P}_{out}: \qquad 0\le x^{\mathrm{sales}}_{p_{\mathrm{link}}}=\sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}} x^{\mathrm{q}}_s\cdot a^{\mathrm{f}}_{s,p_{\mathrm{link}}}-x^{\mathrm{sub}}_{p_{\mathrm{link}}} \tag{6.14d}$$

$$\forall\ p_{\mathrm{link}}\in\mathcal{P}_{\mathrm{link}}\setminus\mathcal{P}_{out}: \qquad 0\le\sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}} x^{\mathrm{q}}_s\cdot a^{\mathrm{f}}_{s,p_{\mathrm{link}}}-x^{\mathrm{sub}}_{p_{\mathrm{link}}}, \tag{6.14e}$$

where $\boldsymbol{x}^{\mathrm{sales}}_{\mathcal{P}_{\mathrm{link}}\bigcap\mathcal{P}_{out}}$ are slack variables.

Remark 6.2.1. Problem (6.14) can be generalized to the case of multiple submodels. For this purpose, we introduce the set $\mathcal{M}$ of submodels and variables $x^{\mathrm{sub}}_{p,m}$ for input/output quantities of linking products $p\in\mathcal{P}_{\mathrm{link}}$ related to a submodel $m\in\mathcal{M}$. We denote the input/output-profit/cost functions of the submodels by c_m for $m\in\mathcal{M}$. Furthermore, we define $D_{\mathcal{M}}:=D_1\times\ldots\times D_{|\mathcal{M}|}$. The result is the following constrained problem

$$\begin{aligned}
&\max_{\substack{\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}},\\ \boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_{\mathrm{link}},\mathcal{M}}\in D_{\mathcal{M}}}} F_2(\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}}, \boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_{\mathrm{link}},\mathcal{M}}) \\
&\quad = \max_{\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}},\boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_{\mathrm{link}},\mathcal{M}}} \sum_{\substack{p_{out}\in\\ \mathcal{P}_{out}\setminus(\mathcal{P}_{\mathrm{b}}\bigcup\mathcal{P}_{\mathrm{link}})}} \Big(\sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}} x^{\mathrm{q}}_s\cdot a^{\mathrm{f}}_{s,p_{out}}\Big)\cdot P_{p_{out}}\Big(\sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}} x^{\mathrm{q}}_s\cdot a^{\mathrm{f}}_{s,p_{out}}\Big) \\
&\qquad + \sum_{\substack{p_{ex}\in\mathcal{P}_{ex},\\ s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}}} x^{\mathrm{q}}_s\cdot a^{\mathrm{f}}_{s,p_{ex}}\cdot a^{\pi}_{p_{ex}} + \sum_{p_{\mathrm{link}}\in\mathcal{P}_{\mathrm{link}}\bigcap\mathcal{P}_{out}} x^{\mathrm{sales}}_{p_{\mathrm{link}}}\cdot P_{p_{\mathrm{link}}}(x^{\mathrm{sales}}_{p_{\mathrm{link}}}) + \sum_{m\in\mathcal{M}} c_m(\boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_{\mathrm{link}},m})
\end{aligned} \tag{6.15a}$$

such that

$$\forall\ s \in \mathcal{S}\backslash\mathcal{S}_\mathrm{b}: \qquad 0 \leq x_s^\mathrm{q} \leq a_s^\mathrm{cap} \tag{6.15b}$$

$$\forall\ p \in (\mathcal{P}_{out} \dot{\bigcup} \mathcal{P}_{mid})\backslash(\mathcal{P}_\mathrm{b} \dot{\bigcup} \mathcal{P}_\mathrm{link}): \quad 0 \leq \sum_{s\in\mathcal{S}\backslash\mathcal{S}_\mathrm{b}} x_s^\mathrm{q} \cdot a_{s,p}^\mathrm{f} \tag{6.15c}$$

$$\forall\ p_\mathrm{link} \in \mathcal{P}_\mathrm{link} \bigcap \mathcal{P}_{out}: \qquad 0 \leq x_{p_\mathrm{link}}^\mathrm{sales} = \sum_{s\in\mathcal{S}\backslash\mathcal{S}_\mathrm{b}} x_s^\mathrm{q} \cdot a_{s,p_\mathrm{link}}^\mathrm{f} - \sum_{m\in\mathcal{M}} x_{p_\mathrm{link},m}^\mathrm{sub} \tag{6.15d}$$

$$\forall\ p_\mathrm{link} \in \mathcal{P}_\mathrm{link} \setminus \mathcal{P}_{out}: \qquad 0 \leq \sum_{s\in\mathcal{S}\backslash\mathcal{S}_\mathrm{b}} x_s^\mathrm{q} \cdot a_{s,p_\mathrm{link}}^\mathrm{f} - \sum_{m\in\mathcal{M}} x_{p_\mathrm{link},m}^\mathrm{sub}, \tag{6.15e}$$

where $\boldsymbol{x}^\mathrm{sales}_{\mathcal{P}_\mathrm{link}\bigcap\mathcal{P}_{out}}$ are slack variables.

Remark 6.2.2. Considering the approximation of one subsystem, a reasonable alternative is to include the optimization of sales quantities $\boldsymbol{x}^\mathrm{sales}_{\mathcal{P}_\mathrm{link}\bigcap\mathcal{P}_{out}}$ of linking products with external demand in the optimization problem (6.13) instead of problem (6.14). This obviously influences the choice of input/output values $\boldsymbol{y}_1^\mathrm{q}, ..., \boldsymbol{y}_n^\mathrm{q} \in D \subset \mathbb{R}^{|\mathcal{P}_\mathrm{link}|}$, and also the interpretation of results. In case of multiple subsystems with overlapping sets of links, one must ensure that each linking product is only sold in one part of the overall problem. Which part should include the selling of linking products can depend on the application and on the size of the subsystems. In our applications, we always include it, as described above, in the main problem part, represented by problem (6.14) or (6.15), respectively.

In a next step we discuss the relationship between optimal solutions of problem (6.5) and solutions of problem (6.14) if we choose $c(\boldsymbol{y}^\mathrm{q}) = F_1^*(\boldsymbol{y}^\mathrm{q})$, where F_1^* is the optimal value function of problem (6.13) with parameter $\boldsymbol{y}^\mathrm{q}$. As we will show in the proof of theorem 6.5.2, F_1^* is concave on the relevant domain $\tilde{D}$ that will be defined by equation (6.32). This makes problem (6.14) to a convex optimization problem, cf. also remark 6.1.2.
The following theorem serves as a justification and basis for our model reduction approach.[3]

Theorem 6.2.3. *Let $F_1^*(\boldsymbol{y}^\mathrm{q})$ be the optimal value function and $S(\boldsymbol{y}^\mathrm{q})$ be the optimal solution map of the parametric problem* (6.13) *and set in optimization problem* (6.14) *$c(\boldsymbol{y}^\mathrm{q}) = F_1^*(\boldsymbol{y}^\mathrm{q})$. Then,*

a) each (global) maximum $(\boldsymbol{x}^{\mathrm{q}*}_{\mathcal{S}\backslash\mathcal{S}_\mathrm{b}}, \boldsymbol{x}^{\mathrm{q}*}_{\mathcal{S}_\mathrm{b}})$ of the original optimization problem (6.5) *includes, with $x^{\mathrm{sub}*}_{p_\mathrm{link}} := -\sum_{s\in\mathcal{S}_\mathrm{b}} x_s^{\mathrm{q}*}\cdot a^\mathrm{f}_{s,p_\mathrm{link}}$, a solution $(\boldsymbol{x}^{\mathrm{q}*}_{\mathcal{S}\backslash\mathcal{S}_\mathrm{b}}, \boldsymbol{x}^{\mathrm{sub}*}_{\mathcal{P}_\mathrm{link}})$ of problem* (6.14) *and*

[3] The result is strongly related to *Bellman's Principle of Optimality*, which is the basis of dynamic programming.

b) each (global) maximum $(\boldsymbol{x}^{\mathrm{q}*}_{\mathcal{S}\setminus\mathcal{S}_\mathrm{b}}, \boldsymbol{x}^{\mathrm{sub}*}_{\mathcal{P}_\mathrm{link}})$ *of problem* (6.14) *leads to a (global) maximum* $(\boldsymbol{x}^{\mathrm{q}*}_{\mathcal{S}\setminus\mathcal{S}_\mathrm{b}}, \boldsymbol{x}^{\mathrm{q}*}_{\mathcal{S}_\mathrm{b}})$ *of the original optimization problem* (6.5) *by choosing* $\boldsymbol{x}^{\mathrm{q}*}_{\mathcal{S}_\mathrm{b}} \in S(\boldsymbol{x}^{\mathrm{sub}*}_{\mathcal{P}_\mathrm{link}})$.

Proof Let $\mathcal{F}$ and $\mathcal{F}_2$ be the feasible sets of problem (6.5) and problem (6.14), respectively. Then, we have

$$(\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}\setminus\mathcal{S}_\mathrm{b}}, \boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}_\mathrm{b}}) \in \mathcal{F},\ x^{\mathrm{sub}}_{p_\mathrm{link}} = -\sum_{s\in\mathcal{S}_\mathrm{b}} x^{\mathrm{q}}_s \cdot a^{\mathrm{f}}_{s,p_\mathrm{link}}\ \forall\, p_\mathrm{link} \in \mathcal{P}_\mathrm{link} \ \Rightarrow\ (\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}\setminus\mathcal{S}_\mathrm{b}}, \boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_\mathrm{link}}) \in \mathcal{F}_2 \quad (6.16)$$

and

$$(\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}\setminus\mathcal{S}_\mathrm{b}}, \boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_\mathrm{link}}) \in \mathcal{F}_2,\ \boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}_\mathrm{b}} \in S(\boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_\mathrm{link}}) \quad \Rightarrow \quad (\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}\setminus\mathcal{S}_\mathrm{b}}, \boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}_\mathrm{b}}) \in \mathcal{F}. \quad (6.17)$$

Furthermore, it is clear that each solution $(\boldsymbol{x}^{\mathrm{q}*}_{\mathcal{S}\setminus\mathcal{S}_\mathrm{b}}, \boldsymbol{x}^{\mathrm{q}*}_{\mathcal{S}_\mathrm{b}})$ of problem (6.5) fulfills $\boldsymbol{x}^{\mathrm{q}*}_{\mathcal{S}_\mathrm{b}} \in S(\boldsymbol{x}^{\mathrm{sub}*}_{\mathcal{P}_\mathrm{link}})$, where $x^{\mathrm{sub}*}_{p_\mathrm{link}} := -\sum_{s\in\mathcal{S}_\mathrm{b}} x^{\mathrm{q}*}_s \cdot a^{\mathrm{f}}_{s,p_\mathrm{link}}\ \forall\, p_\mathrm{link} \in \mathcal{P}_\mathrm{link}$. Therefore, with $c = F_1^*$ and the above relationship between the variables $\boldsymbol{x}^{\mathrm{q}*}_{\mathcal{S}_\mathrm{b}}$ and $\boldsymbol{x}^{\mathrm{sub}*}_{\mathcal{P}_\mathrm{link}}$, the objective function values of both problems are equal. This means,

a) if there was a feasible point $(\tilde{\boldsymbol{x}}^{\mathrm{q}}_{\mathcal{S}\setminus\mathcal{S}_\mathrm{b}}, \tilde{\boldsymbol{x}}^{\mathrm{sub}}_{\mathcal{P}_\mathrm{link}})$ of (6.14) with a higher objective value, $(\boldsymbol{x}^{\mathrm{q}*}_{\mathcal{S}\setminus\mathcal{S}_\mathrm{b}}, \boldsymbol{x}^{\mathrm{q}*}_{\mathcal{S}_\mathrm{b}})$ would not maximize (6.5), since we could construct a feasible point $(\tilde{\boldsymbol{x}}^{\mathrm{q}}_{\mathcal{S}\setminus\mathcal{S}_\mathrm{b}}, \tilde{\boldsymbol{x}}^{\mathrm{q}}_{\mathcal{S}_\mathrm{b}})$ of (6.5) with a higher objective function value by choosing $\tilde{\boldsymbol{x}}^{\mathrm{q}}_{\mathcal{S}_\mathrm{b}} \in S(\tilde{\boldsymbol{x}}^{\mathrm{sub}}_{\mathcal{P}_\mathrm{link}})$ and

b) if there was a feasible point $(\tilde{\boldsymbol{x}}^{\mathrm{q}}_{\mathcal{S}\setminus\mathcal{S}_\mathrm{b}}, \tilde{\boldsymbol{x}}^{\mathrm{q}}_{\mathcal{S}_\mathrm{b}})$ of (6.5) with a higher objective function value, $(\boldsymbol{x}^{\mathrm{q}*}_{\mathcal{S}\setminus\mathcal{S}_\mathrm{b}}, \boldsymbol{x}^{\mathrm{sub}*}_{\mathcal{P}_\mathrm{link}})$ would not maximize (6.14), since we could find a feasible point $(\tilde{\boldsymbol{x}}^{\mathrm{q}}_{\mathcal{S}\setminus\mathcal{S}_\mathrm{b}}, \tilde{\boldsymbol{x}}^{\mathrm{sub}}_{\mathcal{P}_\mathrm{link}})$ of (6.14) with a higher objective function value by defining $\tilde{x}^{\mathrm{sub}}_{p_\mathrm{link}} := -\sum_{s\in\mathcal{S}_\mathrm{b}} \tilde{x}^{\mathrm{q}}_s \cdot a^{\mathrm{f}}_{s,p_\mathrm{link}}\ \forall\, p_\mathrm{link} \in \mathcal{P}_\mathrm{link}$.

Let us discuss these results in the context of decomposition methods, cf. section 3.5. Our approach is closely related to primal decomposition of block-separable problems with coupling constraints, which we presented in section 3.5. This can be seen if we include an additional variable $\boldsymbol{y}^{\mathrm{q}}_{\mathcal{P}_\mathrm{link}} \in D$, where $D \subset \mathbb{R}^{|\mathcal{P}_\mathrm{link}|}$ is suitably chosen, and the partitions of the sets $\mathcal{S}$ and $\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$ in the production optimization problem in a separable formulation, which is given by (6.11). We get

$$\begin{aligned}\max_{\boldsymbol{y}^{\mathrm{q}}_{\mathcal{P}_\mathrm{link}}} \Bigg(&\max_{\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}\setminus\mathcal{S}_b}, \boldsymbol{x}^{\mathrm{net\_prod}}_{\mathcal{P}_{out}\setminus\mathcal{P}_b}} \sum_{p_{out}\in\mathcal{P}_{out}\setminus\mathcal{P}_b} x^{\mathrm{net\_prod}}_{p_{out}} \cdot P_{p_{out}}\big(x^{\mathrm{net\_prod}}_{p_{out}}\big) + \sum_{p_{ex}\in\mathcal{P}_{ex}, s\in\mathcal{S}\setminus\mathcal{S}_b} x^{\mathrm{q}}_s \cdot a^{\mathrm{f}}_{s,p_{ex}} \cdot a^{\pi}_{p_{ex}} \\ &+ \max_{\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}_b}, \boldsymbol{x}^{\mathrm{net\_prod}}_{\mathcal{P}_{out}\cap\mathcal{P}_b}} \sum_{p_{out}\in\mathcal{P}_{out}\bigcap\mathcal{P}_b} x^{\mathrm{net\_prod}}_{p_{out}} \cdot P_{p_{out}}\big(x^{\mathrm{net\_prod}}_{p_{out}}\big) + \sum_{p_{ex}\in\mathcal{P}_{ex}, s\in\mathcal{S}_b} x^{\mathrm{q}}_s \cdot a^{\mathrm{f}}_{s,p_{ex}} \cdot a^{\pi}_{p_{ex}} \Bigg)\end{aligned} \quad (6.18a)$$

such that

$$\forall\ s \in \mathcal{S}: \qquad 0 \le x^{\mathrm{q}}_s \le a^{\mathrm{cap}}_s \quad (6.18b)$$

$$\forall\ p \in \mathcal{P}_{out}: \qquad 0 \leq x_p^{\text{net\_prod}} \tag{6.18c}$$

$$\forall\ p \in \mathcal{P}_{mid} \setminus (\mathcal{P}_{\text{link}} \bigcup \mathcal{P}_{\text{b}}): \qquad 0 \leq \sum_{s \in \mathcal{S} \setminus \mathcal{S}_{\text{b}}} x_s^{\text{q}} \cdot a_{s,p}^{\text{f}} \tag{6.18d}$$

$$\forall\ p \in \mathcal{P}_{out} \setminus (\mathcal{P}_{\text{link}} \bigcup \mathcal{P}_{\text{b}}): \qquad 0 = x_p^{\text{net\_prod}} - \sum_{s \in \mathcal{S} \setminus \mathcal{S}_{\text{b}}} x_s^{\text{q}} \cdot a_{s,p}^{\text{f}} \tag{6.18e}$$

$$\forall\ p_{\text{link}} \in \mathcal{P}_{mid} \bigcap \mathcal{P}_{\text{link}}: \qquad 0 \leq \sum_{s \in \mathcal{S} \setminus \mathcal{S}_{\text{b}}} x_s^{\text{q}} \cdot a_{s,p_{\text{link}}}^{\text{f}} - y_{p_{\text{link}}}^{\text{q}} \tag{6.18f}$$

$$\forall\ p_{\text{link}} \in \mathcal{P}_{out} \bigcap \mathcal{P}_{\text{link}}: \qquad 0 = x_{p_{\text{link}}}^{\text{net\_prod}} - \sum_{s \in \mathcal{S} \setminus \mathcal{S}_{\text{b}}} x_s^{\text{q}} \cdot a_{s,p_{\text{link}}}^{\text{f}} + y_{p_{\text{link}}}^{\text{q}} \tag{6.18g}$$

$$\forall\ p_{\text{link}} \in \mathcal{P}_{mid} \bigcap \mathcal{P}_{\text{link}}: \qquad 0 \leq y_{p_{\text{link}}}^{\text{q}} + \sum_{s \in \mathcal{S}_{\text{b}}} x_s^{\text{q}} \cdot a_{s,p_{\text{link}}}^{\text{f}} \tag{6.18h}$$

$$\forall\ p_{\text{link}} \in \mathcal{P}_{out} \bigcap \mathcal{P}_{\text{link}}: \qquad 0 = y_{p_{\text{link}}}^{\text{q}} + \sum_{s \in \mathcal{S}_{\text{b}}} x_s^{\text{q}} \cdot a_{s,p_{\text{link}}}^{\text{f}} \tag{6.18i}$$

$$\forall\ p \in \mathcal{P}_{mid} \bigcap \mathcal{P}_{\text{b}}: \qquad 0 \leq \sum_{s \in \mathcal{S}_{\text{b}}} x_s^{\text{q}} \cdot a_{s,p}^{\text{f}} \tag{6.18j}$$

$$\forall\ p \in \mathcal{P}_{out} \bigcap \mathcal{P}_{\text{b}}: \qquad 0 = x_p^{\text{net\_prod}} - \sum_{s \in \mathcal{S}_{\text{b}}} x_s^{\text{q}} \cdot a_{s,p}^{\text{f}} \tag{6.18k}$$

This problem decomposes into two subproblems. The first one is in the variables $\boldsymbol{x}^{\text{q}}_{\mathcal{S}\setminus\mathcal{S}_b}$ and $\boldsymbol{x}^{\text{net\_prod}}_{\mathcal{P}_{out}\setminus\mathcal{P}_b}$ and constrained by (6.18d) to (6.18g), the second one is in the variables $\boldsymbol{x}^{\text{q}}_{\mathcal{S}_b}$ and $\boldsymbol{x}^{\text{net\_prod}}_{\mathcal{P}_b}$ and constrained by (6.18h) to (6.18k). The master problem entails maximizing the overall profit by adjusting the variable $\boldsymbol{y}^{\text{q}}_{\mathcal{P}_{\text{link}}}$.
In our above approach, we consider the second problem as a parametric problem in the variable $\boldsymbol{y}^{\text{q}}_{\mathcal{P}_{\text{link}}}$, and approximate its optimal value function by the input/output-profit/cost function c.[4] Then, we combine the subproblem in the remaining variables with the master problem, which results in problem (6.14).

Before we apply our approach to the first small example, we summarize its advantages:

- The connection between the main part of the production system and subsystems is described by a few linking variables. Each optimal solution of the overall problem

[4]Note that, in contrast to the required equality in constraint (6.18i) for linking products with external demand, the corresponding constraint (6.13d) of problem (6.13) allows inequality. However, a certain inequality constraint of (6.13d) is inactive only if a further freely available quantity of the linking product does not raise profit in the subsystem. This happens, e.g., if the respective subsystem capacities are exploited. Assuming the same optimal objective value of problem (6.13) for two different parameters $\tilde{y}^{\text{q}}_{p_{\text{link}}}$ and $\bar{y}^{\text{q}}_{p_{\text{link}}}$, an optimal solution of problem (6.14) usually includes the lower value, i.e., $x^{\text{sub}\,*}_{p_{\text{link}}} = \min(\tilde{y}^{\text{q}}_{p_{\text{link}}}, \bar{y}^{\text{q}}_{p_{\text{link}}})$, since producing the product usually raises costs and selling the product at the market usually increases profit. In this situation, providing the substem with the product does not make any sense. Therefore, this difference in the problem formulations does not affect the overall solution.

includes a solution of each subproblem as partial solution. Therefore, varying the linking variables of a subproblem gives complete information about the range of its primal and dual optimal solutions. This range depends on the subsystem's parameters but not on the parameters of the remaining part of the system.

- Sensitivities within the subsystem can be computed together with each subsystem optimization. Therefore, not merely local sensitivities are available, but rather the whole range of sensitivity values, given fixed subsystem parameters, can be analyzed. This is because the variation of the linking quantities reflects all possible variations of parameters outside the subsystem.

- Detailed subsystem analyses and simulations of diverse subsystem scenarios become possible through the decoupling.

- In our network optimization model, changes in parameter values often cause active-set-changes in the optimal solutions. Thus, it is difficult to approximate perturbed optimal solutions using only sensitivity information of one optimization. Using aggregations of subsystems with fixed parameters and calculating the exact solution of the approximating optimization problem with perturbed parameters is a reasonable alternative.

- Our decomposition approach preserves the structure of the model so that the results remain well interpretable.

- Sensitivity analysis within small subsystems facilitates the classification of parameters into more and less important ones, which reduces the effort of data procurement.

6.3 Numerical Results 1

We apply our approach to the model of section 6.1 with sets and parameter values shown in tables 6.1 to 6.4. It is the same small part of the petrochemical network that we simulated in the example of section 2.5. In the present example, however, we consider prices of one region in one time period. Hence, there are less processes related to the products of interest. The chosen sets of products and processes build a consistent model to simulate prices of the products 54, 55, 84, 102, and 179 under the assumption that there is no incentive for trade and transport between the region under consideration and other regions. The used parameters are real historical data as far as available and reasonable estimates if real data is missing. In each of the following three examples, we present aggregations of one selected subsystem of the production network.

$\mathcal{P}_{ex}$	25, 31, 49, 66, 68, 76, 89, 98, 111, 120, 143, 260, 273, 284
$\mathcal{P}_{out}$	54, 55, 84, 102, 179
$\mathcal{S}$	47, 48, 49, 58, 97, 120, 121, 135, 136, 139, 165, 169, 244, 278, 291

Table 6.1: *Sets*

	25	**31**	**49**	**66**	**68**	**76**	**89**	**98**	**111**	**120**	**143**	**260**	**273**	**284**
47	0	0	0	0	-0.78	0	0	0	-0.28	0	0	0	0	0
48	0	0	0	0	-0.81	0	0	0	-0.29	0	0	0	0	0
49	0	0	0	0	0	0	0	-0.01	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0	0	0	-0.07	0	0	0
97	0	1	0	0	0	0	-0.48	0	0	0	0	0	0	0
120	0	0	0	0	0	0	-0.7	0	0	0	0	0	0	0
121	0	0	0	0	0	0	-0.7	0	0	0	0	0	0	0
135	-0.24	0	0	0	0	0	-0.17	0	0	0	0	0	0	0
136	-0.26	0	0	0	0	0	-0.17	0	0	0	0	0	0	0
139	0	1	0	0	0	0	-0.48	0	0	0	0	0	0	0
165	0	0	0	0	0	1	-0.65	0	0	0	0	0	0	0
169	0	0	0.43	0	-0.86	0	0	0	-0.31	-0.34	0	0	0	0
244	0	0	0	0	0	0	0	0	0	0	0	1	0	0
278	-0.39	0	0	0	0	0	-0.15	0	0	0	0	0	0.25	0
291	0	0	0	-0.4	0	0	-0.3	0	0	0	0	0	0	1
$a^{\pi}_{p_{ex}}$	1808	2630	1500	2700	1016	2500	1734	1195	1276	1412	890	4100	1900	2700

Table 6.2: *Production factors $a^{\mathrm{f}}_{s,p}$ $(p \in \mathcal{P}_{ex})$ together with external prices a^{π}_{p} $(p \in \mathcal{P}_{ex})$ in the last row*

	54	**55**	**84**	**102**	**179**	**Capacity**
47	0	0	1	0	0	3215
48	0	0	1	0	0	2617
49	0	1	-1	0	0	3460
58	0	0	-0.95	0	1	1883
97	0	0	-0.57	0	0	1388
120	0	0	-0.26	1	0	295
121	0	0	-0.24	1	0	1247
135	1	0	-0.52	0	0	926
136	1	0	-0.56	0	0	205
139	0	0	-0.4	0	0	248
165	0	0	-0.3	0	0	425
169	0	0	1	0	0	2445
244	0	0	-1	0	0	36
278	1	0	-0.67	0	0	210
291	0	0	-0.3	0	0	95

Table 6.3: *Production factors $a^{\mathrm{f}}_{s,p}$ $(p \in \mathcal{P}_{out})$ and process capacities a^{cap}_{s}*

	54	**55**	**84**	**102**	**179**	
$a^{\mathrm{con\_last}}_{p_{out}}$	1153	3266	1646	1304	1761	
$a^{\pi\_\mathrm{last}}_{p_{out}}$	2250	1674	1284	3010	1920	
$\alpha^{GDP}_{p_{out}}$	0.1046	0.0919	0.1156	0.7014	0.07	
$\beta^{IndPro}_{p_{out}}$	13.07	6.23	3.51	-149.15	47.40	
a^{GDP}						21323 - 20129
a^{IndPro}						110.46 - 109.56

Table 6.4: *Parameters related to $\mathcal{P}_{out}$*

Example 1.1

Our first partition of the network is visualized in figure 6.4.[5] The only linking product in the example is product 84, so that the function c that we will fit is univariate.
Solving problem (6.13) for varying values $y^{\mathrm{q}}_{84,i}$ leads to the optimal solutions π_i that are shown in table 6.5 together with the Lagrange multipliers $\lambda^{\mathrm{sub}}_{84,i} := \lambda^{\mathrm{sub}}_{84}(y^{\mathrm{q}}_{84,i})$ corresponding to constraint (6.13d). These values present the derivative of the optimal value function $F_1^*(y^{\mathrm{q}}_{84,i})$ of the parametric problem (6.13) at its differentiable points. To fit the data $(y^{\mathrm{q}}_{84,i}, \pi_i), i \in \{1, ..., 16\}$, we choose a scaled hyperbolic tangent (tanh) function as well as a quadratic polynomial. Results of the least squares estimations are

$$c_{\tanh}(y) = 9049840 \cdot \tanh(y/1108.14) \tag{6.19}$$

$$c_{\mathrm{poly}}(y) = 762033 + 6889y - 1.4298y^2. \tag{6.20}$$

[5] The network in this figure can be considered as a hypergraph if we regard all edges with same label, i.e., edges representing the same processes, as one hyperedge. Note that most products of this network are external products, which are not represented in graphs like those of figures 6.1 and 6.2.

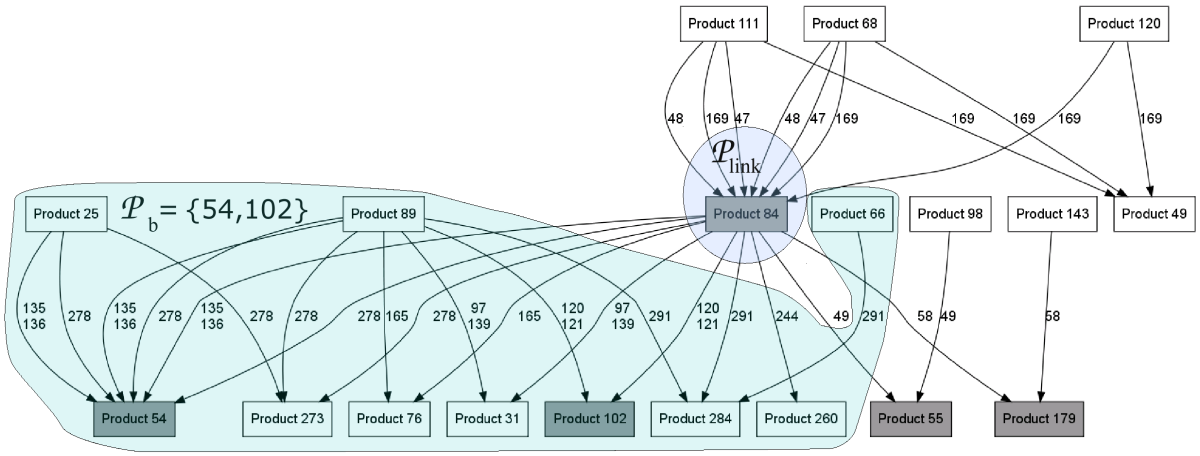

Figure 6.4: *Partition of the network (example 1.1)*

$y^{\mathrm{q}}_{84,i}$	0	160	320	480	640	800	960	1120
π_i	0	2114960	3316520	4052990	4771340	5420570	5970140	6474780
$\lambda^{\mathrm{sub}}_{84,i}$	15432	10815	4771	4577	4346	3723	3154	3154

$y^{\mathrm{q}}_{84,i}$	1280	1440	1600	1760	1920	2080	2240	2400
π_i	6979420	7484060	7988700	8489410	8867370	9048600	9049840	9049840
$\lambda^{\mathrm{sub}}_{84,i}$	3154	3154	3154	2934	1747	181	0	0

Table 6.5: *Optimal values π_i of problem (6.13) for varying input parameters $y^{\mathrm{q}}_{84,i}$ together with sensitivities $\lambda^{\mathrm{sub}}_{84,i}$ (example 1.1)*

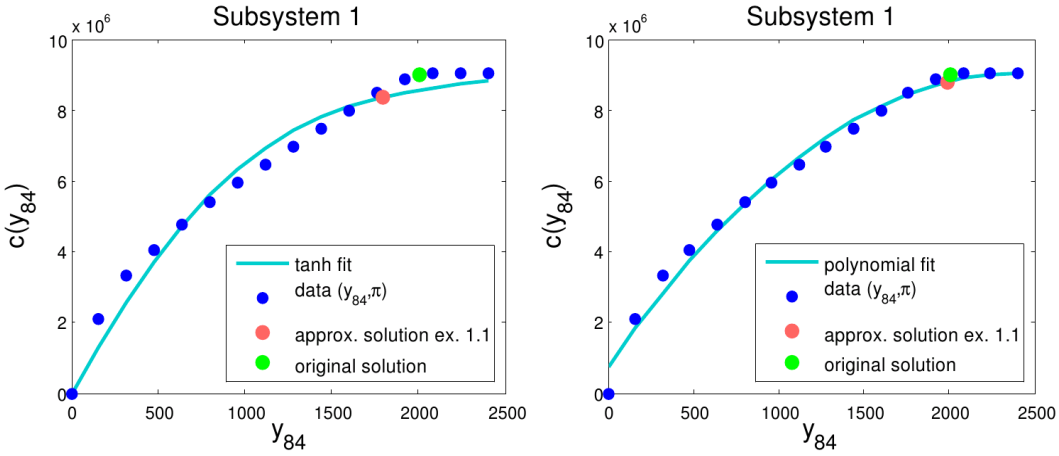

Figure 6.5: *Two fits of the input/output-profit/cost function c (example 1.1)*

	47	**48**	**49**	**58**	**97**	**120**	**121**	**135**	**136**	**139**	**165**	**169**
orig. results	3215	1893	2810	1619	1388	129	1247	863	0	248	425	2445
tanh approx.	3215	1680	2810	1619	1388	56	1247	760	0	248	425	2445
poly. approx.	3215	1881	2810	1619	1388	123	1247	842	0	248	425	2445

	244	**278**	**291**	**54**	**55**	**84**	**102**	**179**	z_{84}^{q*}	x_{84}^{sub*}	**obj.value**
orig. results	36	210	95	2476	1916	1599	3609	2133	-	2004	10964684
tanh approx.	36	0	95	2759	1916	1599	3715	2133	2992	1791	10585857
poly. approx.	36	210	95	2497	1916	1599	3618	2133	3193	1992	10794734

Table 6.6: *Solutions of (6.14) with $c(y) = c_{\tanh}(y)$ and $c(y) = c_{\text{poly}}(y)$ (example 1.1), where $z_{84}^{q*} := \sum_{s \in \mathcal{S} \setminus \mathcal{S}_b} x_s^{q*} a_{s,84}^{f}$. The numbers 47,...,291 denote the processes, 54,...,179 denote the products with external demand.*

Figure 6.5 shows these curve fits. Regarding the scaled hyperbolic tangent fit, $c_{\tanh}(y)$, we set the first parameter equal to the maximum value of the π-data, 9049840, which cannot be exceeded because of the capacity limits of the subnetwork. Thus, only the second parameter is estimated.
Next, we solve the optimization problem (6.14) with both, $c(y) = c_{\tanh}(y)$ and $c(y) = c_{\text{poly}}(y)$. Table 6.6 includes corresponding results along with the solutions of the original problem (6.5). The gray numbers in the table result from solving the subproblem (6.13) with $\boldsymbol{y}^{q} = y_{84}^{q} = x_{84,\tanh}^{sub*} = 1791$ and $\boldsymbol{y}^{q} = y_{84}^{q} = x_{84,\text{poly}}^{sub*} = 1992$, respectively. The results of splitting the overall problem (6.5) into the two subproblems (6.13) and (6.14) show the following: even if we do not correctly reproduce the original solutions for the submodel, which consists of processes 97, 120, 121, 135, 136, 139, 165, 244, 278, and 291 and products 54 and 102, the prices of products 55, 84, and 179, in which we are mainly interested in this example, are properly simulated.

Example 1.2

In this second example, we focus on the prices of products 54, 84, and 102. Therefore, we choose another subnetwork to be aggregated, cf. figure 6.6. Fitting the data of table 6.7, which are the optimal solutions of the corresponding subproblem (6.13) under predetermined varying values of y^{q}_{84}, leads to

$$c_{\tanh}(y) = 13286500 \cdot \tanh(y/3746.5) \tag{6.21}$$

$$c_{\mathrm{poly}}(y) = 278148 + 3371y - 0.212y^2. \tag{6.22}$$

The least squares estimates are visualized in figure 6.7. As in (6.19), we set the first parameter in (6.21) equal to the maximum value of the π-data, here 13286500.
The results of solving problem (6.14) with $c(y) = c_{\tanh}(y)$ and $c(y) = c_{\mathrm{poly}}(y)$ can be found in table 6.8 along with the solutions for the full system. The gray numbers in the table derive from solving the subproblem (6.13) with $y^{\mathrm{q}}_{84} = x^{\mathrm{sub}*}_{84,\tanh} = 4273$ and $y^{\mathrm{q}}_{84} = x^{\mathrm{sub}*}_{84,\mathrm{poly}} = 5137$, respectively. The results resemble those of example 1.1: the prices of the "relevant" products 54, 84, and 102 do not change through the reduction, while

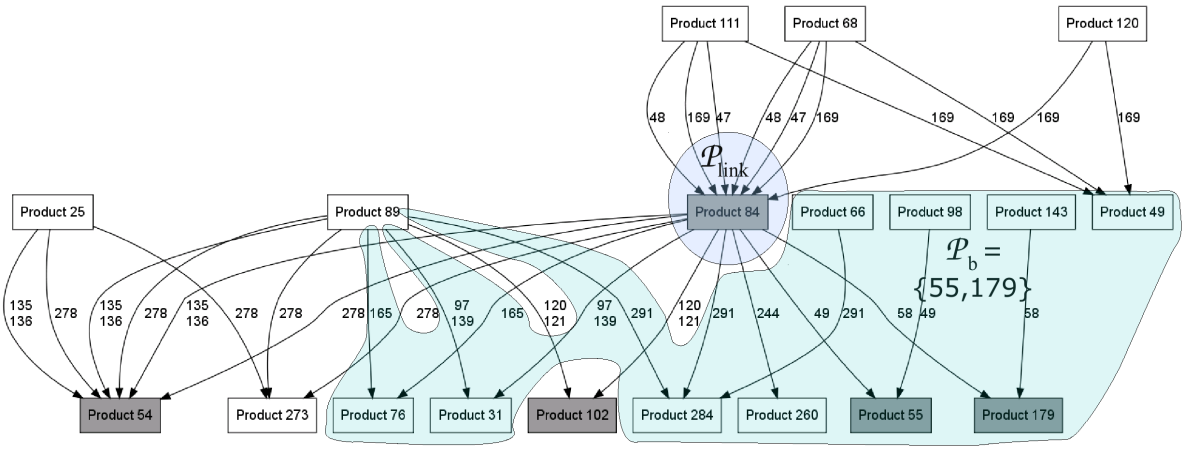

Figure 6.6: *Partition of the network (example 1.2)*

$y^{\mathrm{q}}_{84,i}$	0	400	800	1200	1600	2000	2400	2800	3200
π_i	0	1624280	2885880	4111540	5166810	6128740	7050570	7931300	8769180
$\lambda^{\mathrm{sub}}_{84,i}$	4577	3154	3154	2802	2470	2355	2254	2149	2039

$y^{\mathrm{q}}_{84,i}$	3600	4000	4400	4800	5200	5600	6000	6400
π_i	9561700	10305500	10995800	11626700	12189500	12672500	13058500	13286500
$\lambda^{\mathrm{sub}}_{84,i}$	1922	1795	1655	1496	1313	1095	825	0

Table 6.7: *Optimal values π_i of problem* (6.13) *for varying input parameters $y^{\mathrm{q}}_{84,i}$ together with sensitivities $\lambda^{\mathrm{sub}}_{84,i}$ (example 1.2)*

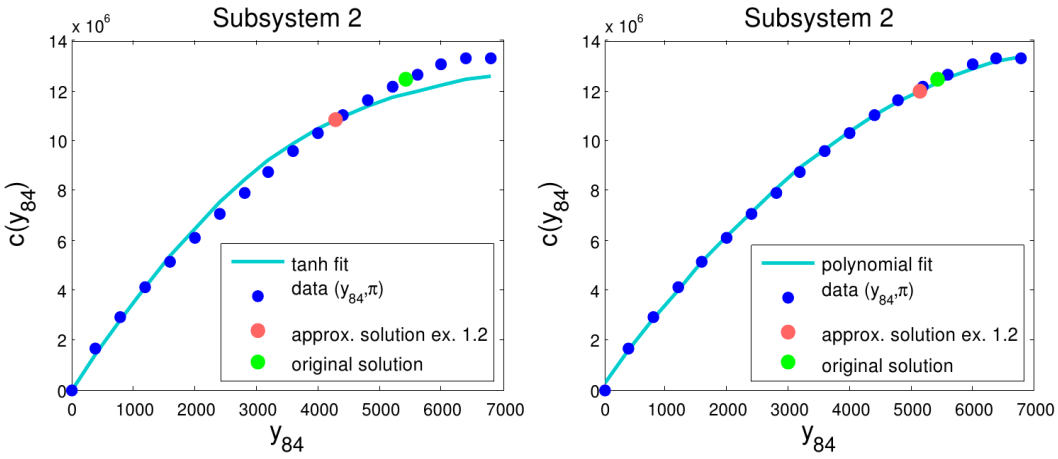

Figure 6.7: *Two fits of the input/output-profit/cost function c (example 1.2)*

	47	**48**	**49**	**58**	**97**	**120**	**121**	**135**	**136**	**139**	**165**	**169**
orig. results	3215	1893	2810	1619	1388	129	1247	863	0	248	425	2445
tanh approx.	3215	737	1955	1301	1388	129	1247	863	0	248	425	2445
poly. approx.	3215	1600	2595	1537	1388	129	1247	863	0	248	425	2445

	244	**278**	**291**	**54**	**55**	**84**	**102**	**179**	z_{84}^{q*}	x_{84}^{sub*}	**obj.value**
orig. results	36	210	95	2476	1916	1599	3609	2133	-	5430	10964684
tanh approx.	36	210	95	2476	2108	1599	3609	2299	5474	4273	10690084
poly. approx.	36	210	95	2476	1969	1599	3609	2179	6338	5137	10836847

Table 6.8: *Solutions of (6.14) with $c(y) = c_{\tanh}(y)$ and $c(y) = c_{\text{poly}}(y)$ (example 1.2), where $z_{84}^{q*} := \sum_{s \in \mathcal{S} \setminus \mathcal{S}_b} x_s^{q*} a_{s,84}^{f}$. The numbers 47,...,291 denote the processes, 54,...,179 denote the products with external demand.*

subsystem quantities are more or less influenced. Furthermore, in both examples, the polynomial fit of the input/output-profit/cost function c yields better results than the tanh fit.

Example 1.3

In this example, we do not aggregate a subnetwork that requires the linking product 84 for processing, but a subnetwork that produces the linking product, cf. figure 6.8. Therefore, we must fit a cost function $c(y)$ depending on the output from the subnetwork instead of a profit function depending on the input to the subnetwork. Since the submodel processes, $\mathcal{S}_b = \{47, 48, 169\}$, only contain the linking product 84 and external products, the set $\mathcal{P}_b$ is empty.
We fit the piecewise linear costs data of table 6.9, which we received by solving problem

(6.13) for varying negative values of y^{q}_{84}, by a linear function as well as by a polynomial of degree 2. This leads to

$$c_{\mathrm{lin}}(y) = \ 69528 + 1148y \tag{6.23}$$

$$c_{\mathrm{poly}}(y) = -2706 + 1090y - 0.007y^2. \tag{6.24}$$

The curve fits c_{lin} and c_{poly} are plotted in figure 6.9. Table 6.10 shows the results from solving problem (6.14) with $c(y) = c_{\mathrm{lin}}(y)$ and $c(y) = c_{\mathrm{poly}}(y)$, where the gray numbers arise from solving the subproblem (6.13) with $y^{\mathrm{q}}_{84} = x^{\mathrm{sub}*}_{84,\mathrm{lin}} = -7686$ and $y^{\mathrm{q}}_{84} = x^{\mathrm{sub}*}_{84,\mathrm{poly}} = -7545$, respectively.
In contrast to the previous examples, the prices of products 54, 55, 84, 102, and 179, in which we are mainly interested and which are highlighted in the gray columns of table 6.10, differ in the reduced models from the solution of the original problem (6.5). This deviation is due to the approximation error in the production costs $c(y)$ of product 84 that are the direct price drivers of the products 54, 55, 84, 102, and 179.

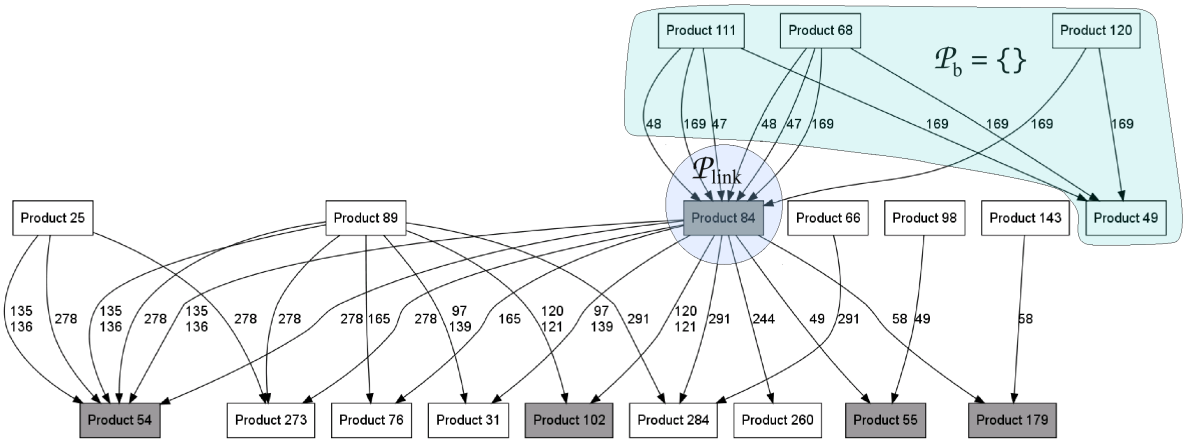

Figure 6.8: *Partition of the network (example 1.3)*

$y^{\mathrm{q}}_{84,i}$	0	-800	-1600	-2400	-3200	-4000
π_i	0	-883392	-1766780	-2650180	-3567890	-4487660
$\lambda^{\mathrm{sub}}_{84,i}$	0	1104	1104	1104	1150	1150

$y^{\mathrm{q}}_{84,i}$	-4800	-5600	-6400	-7200	-7760
π_i	-5407430	-6327190	-7278950	-8233310	-8901360
$\lambda^{\mathrm{sub}}_{84,i}$	1150	1150	1193	1193	1193

Table 6.9: *Optimal values π_i of problem (6.13) for varying output parameters $y^{\mathrm{q}}_{84,i}$ together with sensitivities $\lambda^{\mathrm{sub}}_{84,i}$ (example 1.3)*

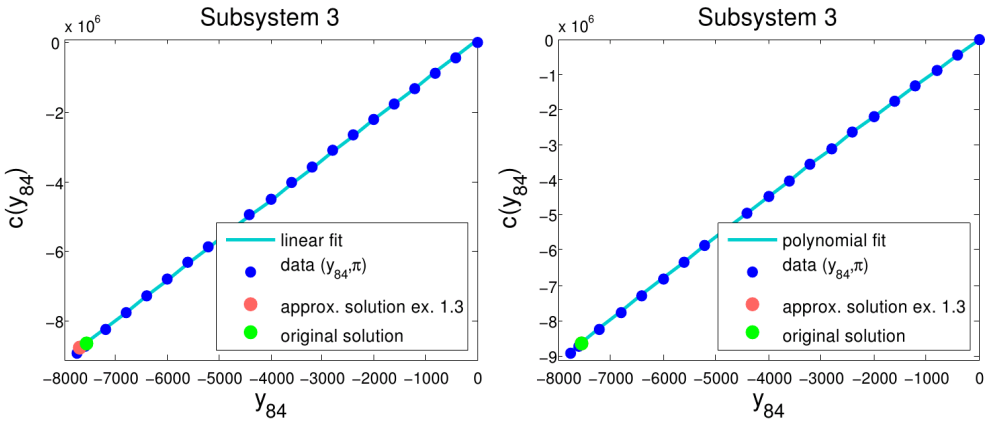

Figure 6.9: *Two fits of the input/output-profit/cost function c (example 1.3)*

	47	**48**	**49**	**58**	**97**	**120**	**121**	**135**	**136**	**139**	**165**	**169**
orig. results	3215	1893	2810	1619	1388	129	1247	863	0	248	425	2445
lin. approx.	3215	2026	2868	1642	1388	131	1247	869	0	248	425	2445
poly. approx.	3215	1885	2806	1618	1388	129	1247	862	0	248	425	2445

	244	**278**	**291**	**54**	**55**	**84**	**102**	**179**	z_{84}^{q*}	x_{84}^{sub*}	**obj.value**
orig. results	36	210	95	2476	1916	1599	3609	2133	-	-7553	10964684
lin. approx.	36	210	95	2469	1901	1582	3606	2120	-6436	-7686	11020793
poly. approx.	36	210	95	2476	1917	1600	3609	2134	-6347	-7545	10984303

Table 6.10: *Solutions of* (6.14) *with* $c(y) = c_{\text{lin}}(y)$ *and* $c(y) = c_{\text{poly}}(y)$ *(example 1.3), where* $z_{84}^{q*} := \sum_{s \in \mathcal{S} \setminus \mathcal{S}_b} x_s^{q*} a_{s,84}^{f}$. *The numbers 47,...,291 denote the processes, 54,...,179 denote the products with external demand.*

Extension of the Showcase Network

Algorithm 2.4.1 sets up a minimal consistent network model starting with at least one product with external demand for which one intends to simulate price formation, cf. section 2.4. Moreover, algorithm 2.4.2 is able to expand a given consistent network model using algorithm 2.4.1. Since we aim to apply our reduction approach to a larger network, we expand our above small showcase model by running algorithm 2.4.2.
Figure 6.10 shows the resulting extension of the small network shown in figure 6.4, where the highlighted part forms the original model. Being more general than the smaller production system, this one includes intermediates without external demand (e.g., product 32). The products with external demand are gray-colored. To restrict the size of the model extension, the intermediate product 128 is regarded as external product. Table 6.11 lists the product and process numbers of the extended model together with process capacities. Tables 6.12 to 6.13 show external prices as well as parameters concerning the products with external demand. Regarding this extended model, we desist from listing all production factors as we did for the smaller model in table 6.2.

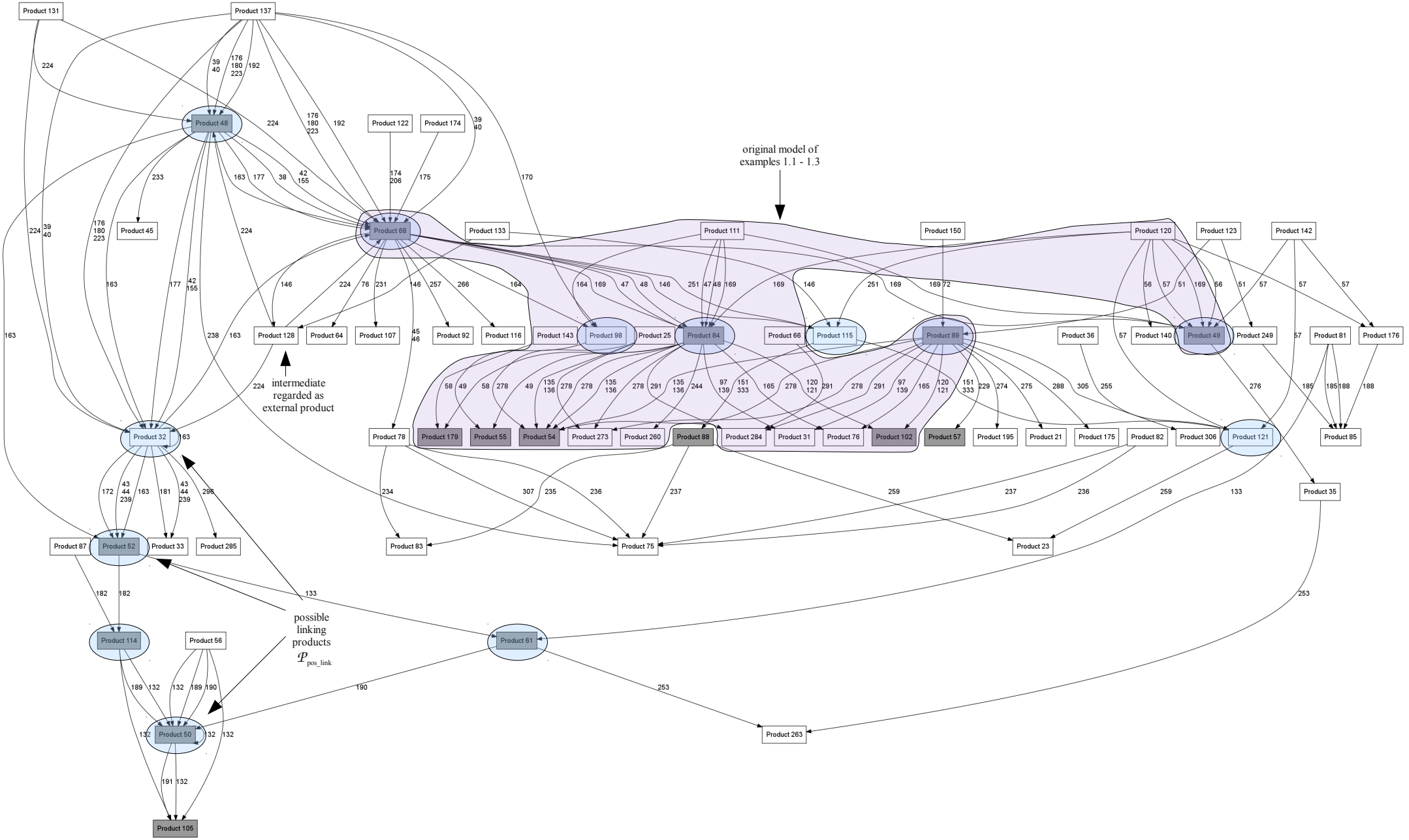

Figure 6.10: *Extended petrochemical network*

$\mathcal{P}_{ex}$	21, 23, 25, 31, 33, 36, 45, 56, 64, 66, 75, 76, 81, 82, 83, 85, 87, 92, 107, 111, 116, 120, 122, 123, 128, 131, 133, 137, 140, 142, 143, 150, 174, 175, 195, 260, 263, 273, 284, 285, 306
$\mathcal{P}_{out}$	48, 49, 50, 52, 54, 55, 57, 61, 68, 84, 88, 89, 102, 105, 114, 179
$\mathcal{P}_{mid}$	32, 35, 78, 98, 115, 121, 176, 249
$\mathcal{S}$ (with capacities)	38(1457), 39(590), 40(200), 42(200), 43(1570), 44(16), 45(1408), 46(1209), 47(3215), 48(2617), 49(3460), 51(3214), 56(1128), 57(470), 58(1883), 72(245), 76(82), 97(1388), 120(295), 121(1247), 132(1225), 133(1230), 135(926), 136(205), 139(248), 146(2100), 151(3092), 155(620), 163(40), 164(7911), 165(425), 169(2445), 170(157), 172(50), 174(7424), 175(3154), 176(70), 177(230), 180(2176), 181(682), 182(6144), 185(2500), 188(1080), 189(4561), 190(1361), 191(3740), 192(1015), 206(1310), 223(755), 224(258), 229(811), 231(2420), 233(440), 234(1053), 235(150), 236(1460), 237(350), 238(80), 239(1705), 244(36), 251(2191), 253(45), 255(155), 257(1225), 259(1847), 266(181), 274(215), 275(164), 276(351), 278(210), 288(28), 291(95), 296(100), 305(480), 307(84), 333(250)

Table 6.11: *Sets*

21	**23**	**25**	**31**	**33**	**36**	**45**	**56**	**64**	**66**	**75**	**76**	**81**	**82**
1200	2097	1808	2630	1168	1538	1205	1311	2500	2700	2665	2500	505	1210

83	**85**	**87**	**92**	**107**	**111**	**116**	**120**	**122**	**123**	**128**	**131**	**133**	**137**
2016	939	1308	1215	1220	1276	1225	1412	350	827	737	500	1092	450

140	**142**	**143**	**150**	**174**	**175**	**195**	**260**	**263**	**273**	**284**	**285**	**306**
1230	818	890	813	629	1235	1240	4100	1245	1900	2700	1250	2393

Table 6.12: *External prices $a^{\pi}_{p_{ex}}$ of $\mathcal{P}_{ex}$*

	48	**49**	**50**	**52**	**54**	**55**	**57**	**61**
$a^{\text{con\_last}}_{p_{out}}$	2306	2516	19301	38	1153	3266	786	220
$a^{\pi\_\text{last}}_{p_{out}}$	800	1925	1559	1061	2250	1674	1500	1046
a^{GDP}	-0.0203	0.0489	-0.0241	0.0597	0.1046	0.0919	0.0051	-0.1406
a^{IndPro}	32.18	25.70	6.3	22.61	13.07	6.23	5.58	118.58

	68	**84**	**88**	**89**	**102**	**105**	**114**	**179**
$a^{\text{con\_last}}_{p_{out}}$	4157	1646	2723	1176	1304	5241	122	1761
$a^{\pi\_\text{last}}_{p_{out}}$	1031	1284	1642	1185	3010	1893	1106	1920
a^{GDP}	0.0535	0.1156	0.3454	-0.0028	0.7014	0.0665	-0.0009	0.07
a^{IndPro}	42.94	3.51	-56.03	42.42	-149.15	30.98	22.07	47.40

Table 6.13: *Parameters related to $\mathcal{P}_{out}$*

6.4 Automated Identification of Network Components Suited for Aggregation

In view of applying the approach proposed in section 6.2 to a large production network, which shall be optimized, this section concerns the identification of network components

that are convenient for aggregation. This means, we aim to detect small, nonempty sets of links, $\mathcal{P}_{\text{link}} \subset \mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$, and related nonempty sets of submodel products, $\mathcal{P}_{\text{b}} \subset \mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$, where $\mathcal{P}_{\text{link}} \bigcap \mathcal{P}_{\text{b}} = \emptyset$, $\mathcal{P}_{\text{b}} \bigcup \mathcal{P}_{\text{link}} \subset \mathcal{P}_{mid} \bigcup \mathcal{P}_{out}$, and no process of $\mathcal{S}$ includes both, a product of $\mathcal{P}_{\text{b}}$ and a product of $(\mathcal{P}_{mid} \bigcup \mathcal{P}_{out}) \setminus (\mathcal{P}_{\text{b}} \bigcup \mathcal{P}_{\text{link}})$, i.e., $\nexists s \in \mathcal{S}, p_1 \in \mathcal{P}_{\text{b}}, p_2 \in (\mathcal{P}_{mid} \bigcup \mathcal{P}_{out}) \setminus (\mathcal{P}_{\text{b}} \bigcup \mathcal{P}_{\text{link}}) : a^{\text{f}}_{s,p_1} \neq 0 \vee a^{\text{f}}_{s,p_2} \neq 0$.
Following [SRWD+11], we distinguish between three basic types of subnetworks that clearly differ from each other.[6] Figures 6.11 to 6.13 sketch these three kinds of network components: sequentially connected nodes, nodes connected in parallel, and marginal subnetworks. In applications, networks also include hybrids of these basic types of subnetworks. In terms of graph theory, cf. chapter 4, our production network can be regarded as hypergraph with products as vertices and hyperedges representing processes, i.e., all products that appear in one process are connected by a hyperedge. Since hyperedges are much harder to visualize than edges, we mostly represent a hyperedge by multiple edges labeled with the same process numbers. Even if external products, $\mathcal{P}_{ext}$, do not influence the connectivity structure of variables, we sometimes represent them in a network illustration.[7]
By its definition, a production network of products $\mathcal{P}_{mid} \bigcup \mathcal{P}_{out}$ consists of several components if and only if its underlying optimization problem (6.5) decomposes into inde-

[6] Inspired by [MSOI+02], [SRWD+11] refers to such network components as motifs, where motifs are, in general, subgraphs that repeat themselves in a specific network or even among various networks.
[7] Without regard to external products, the hypergraph of products $\mathcal{P}_{mid} \bigcup \mathcal{P}_{out}$ is induced by the bipartite graph of figure 6.1. In contrast to the hypergraph of all products, $\mathcal{P} = \mathcal{P}_{out} \bigcup \mathcal{P}_{mid} \bigcup \mathcal{P}_{ext}$, which represents all processes of the set $\mathcal{S}$, this hypergraph does not include processes that contain only one product of the set $\mathcal{P}_{mid} \bigcup \mathcal{P}_{out}$.

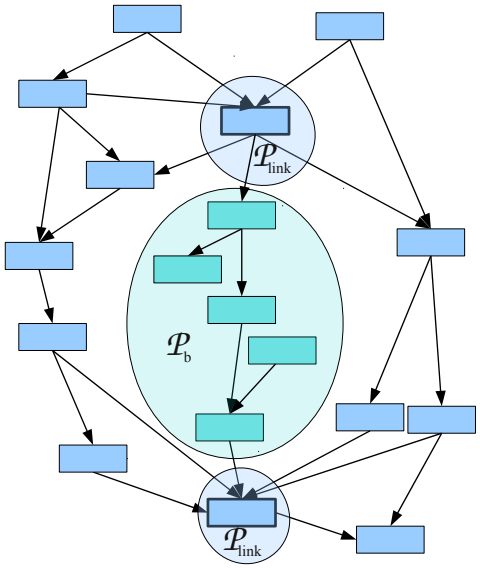

Figure 6.11: *Sequentially connected nodes*

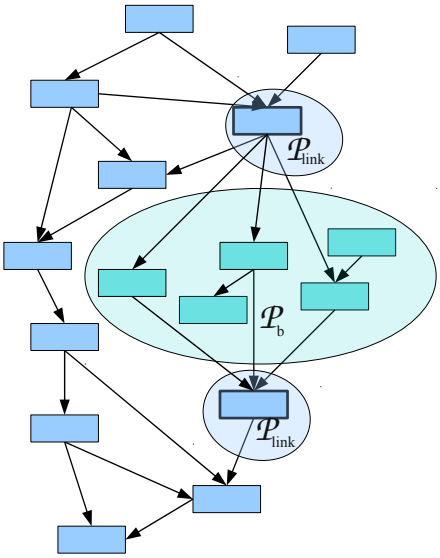

Figure 6.12: *Nodes connected in parallel*

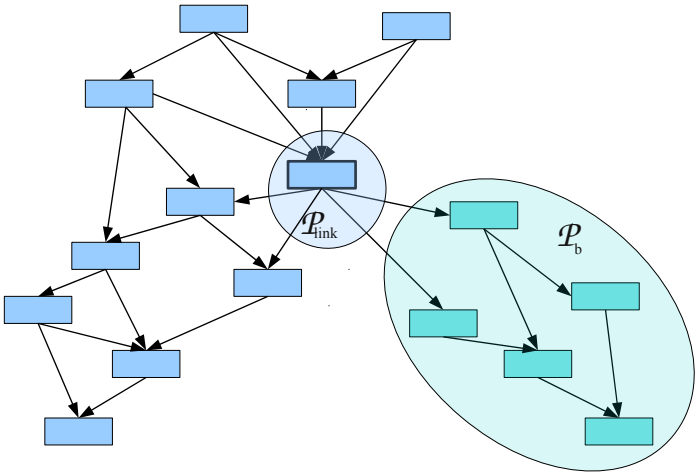

Figure 6.13: *Almost disconnected subgraph*

pendent parts. If so, we can divide the set of processes $\mathcal{S}$ into $k \geq 2$ disjoint, nonempty subsets $\mathcal{S}_i$ such that $\bigcup_{i=1}^{k} \mathcal{S}_i = \mathcal{S}$, where for $i \neq j$ the processes of $\mathcal{S}_i$ and the processes of $\mathcal{S}_j$ have no intermediates, $p \in \mathcal{P}_{mid}$, and no products with external demand, $p \in \mathcal{P}_{out}$, in common.

Assumption 6.4.1. During the rest of this section, we assume that the production network of products $\mathcal{P}_{mid} \bigcup \mathcal{P}_{out}$ is connected, i.e., it does not consist of several subsets of processes that have only external products in common.

If this assumption is violated, the respective optimization problem can be immediately decomposed into independent subproblems, and the following procedure can be applied to each component separately.

We divide the identification of network components that are suited for aggregation into three working steps:

a. Identification of all products that come into consideration as linking products as well as the determination of the relevant connections between them
   $\Rightarrow$ network of possible linking products, which is an undirected graph

b. Partitioning of this network of possible linking products

c. Transfer of the network partition to the original model

a. Determining the Network of Possible Linking Products

Regarding the reduction approach of section 6.2, in principle, all products with external demand, $\mathcal{P}_{out}$, and all intermediates that are not considered as external products, $\mathcal{P}_{mid}$, are candidates for linking products. In the most simple way, the connectivity of these products is represented by connecting each two products that appear in the same process

by an undirected edge. The resulting graph is the induced graph of the hypergraph with node set $\mathcal{P}_{mid} \bigcup \mathcal{P}_{out}$, which we described above, and also the projection of the bipartite graph of figure 6.1 onto the set of dual variables. Figure 6.14 exemplarily shows this simple, undirected graph for the extended optimization problem, which we presented in the end of section 6.3, cf. figure 6.10.
However, this network includes nodes that are not as much qualified to be a link as others. First, these are the products with only one neighbor. In general, there does not exist any narrow vertex-cut that contains such a node of degree one, since any vertex-cut including such a node, is also a vertex-cut without this node. Furthermore, we can disregard the products with only two neighbors that are both at least of degree three as possible links. To make this clear, let us consider a narrow vertex-cut that includes such a vertex v with only two neighbors and induces a bipartition with components of at least size two. Since the vertex-cut is narrow, it does not contain any of the two neighbors of v. Hence, at least one of v's neighbors can replace v in the vertex-cut, while leading to a higher graph fragmentation because of its higher degree. Thus, in a sufficiently large problem, we can exclude several products from the set of possible links: those with only one neighbor and those with two neighbors that are of at least degree three. We will see later on that these neglected products can be assigned to certain network components by means of the processes containing them, cf. paragraph c. Figure 6.15 shows the graph of the reduced set of possible linking products, $\mathcal{P}_{\text{pos\_link}}$, which results from the following procedure, given the sets $\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$ and $\mathcal{S}$:

1. Detect all products of the set $\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$ with only one neighbor $\Rightarrow$ set $\mathcal{P}_{\text{deg\_1}}$.

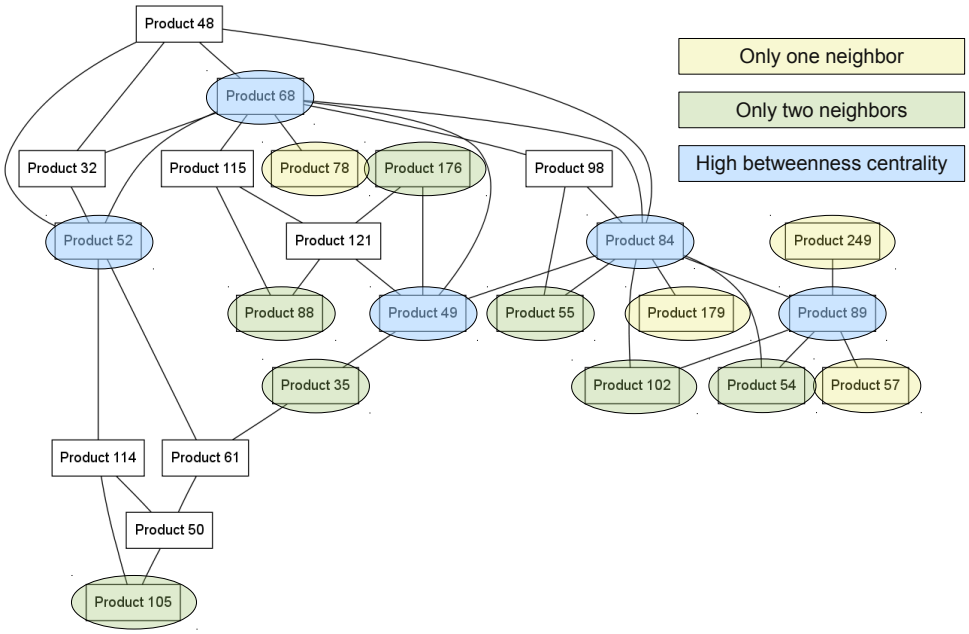

Figure 6.14: *Network of all principally possible linking products, $\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$*

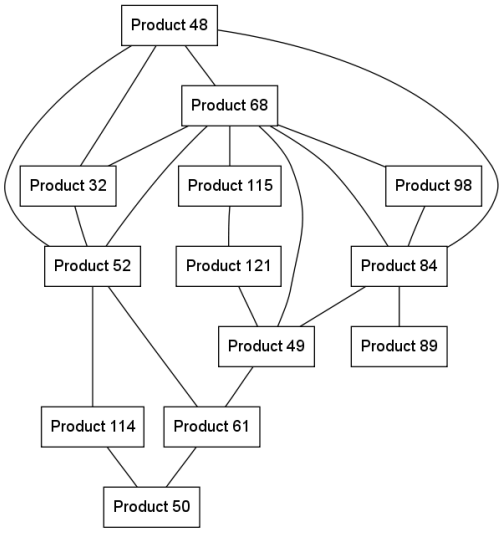

Figure 6.15: *Reduced network of possible linking products* $\mathcal{P}_{\text{pos\_link}}$

2. Detect all products of the set $\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$ with only two neighbors, which are of at least degree three $\Rightarrow$ set $\mathcal{P}_{\text{deg\_2}}$.

3. Define an edge between each two products of

$$\mathcal{P}_{\text{pos\_link}} := \left(\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}\right) \setminus \left(\mathcal{P}_{\text{deg\_1}} \bigcup \mathcal{P}_{\text{deg\_2}}\right) \tag{6.25}$$

   that are part of a common process as well as between each two products that are connected via two different processes to the same product $p \in \mathcal{P}_{\text{deg\_2}}$.

Algorithm 6.4.1 formalizes steps 1 to 3 given a connected, consistent network model. Its output is an appropriate network of possible linking products: each process $s \in \mathcal{S}$ can either be uniquely assigned to one product $p \in \mathcal{P}_{\text{pos\_link}}$ or the possible linking products to which it can be assigned are neighbors, i.e., connected by an edge, within the network of possible linking products. At the end of this section, we will show how a decomposition of this network, which is based on a selection of several linking products $\mathcal{P}_{\text{link}} \subset \mathcal{P}_{\text{pos\_link}}$, can be retransferred to the original model. By this, we finally achieve network components that can be aggregated by approximating an input/output-profit/cost function c as described in section 6.2.

b. Partitioning the Network of Possible Links

Next, we discuss the problem of partitioning the graph of possible linking products and propose, among others, a heuristic approach to solve it. This approach is based on betweenness centrality of vertices.
In general, our task consists in identifying subsets $\mathcal{P}_{\text{link}} \subset \mathcal{P}_{\text{pos\_link}}$ of products whose removal decomposes the hypergraph of relevant products $\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$, i.e., a set $\mathcal{P}_{\text{link}}$ of

Algorithm 6.4.1: Determining the Network of Possible Linking Products

Input: Disjoint product sets $\mathcal{P}_{out}$, $\mathcal{P}_{mid}$ and set of processes $\mathcal{S}$ that build a connected, consistent model.

Output: Set of possible linking products $\mathcal{P}_{\text{pos\_link}} \subseteq (\mathcal{P}_{out} \cup \mathcal{P}_{mid})$ and set $\mathcal{E}$ of edges $\{p_i, p_j\}_{p_i,p_j \in \mathcal{P}_{\text{pos\_link}}}$ between each two possible linking products p_i and p_j that are directly connected in the meaning of the above step 3.

$\mathcal{P}_{\text{pos\_link}} = \mathcal{E} = \mathcal{P}_{\text{deg\_1}} = \mathcal{P}_{\text{deg\_2}} = \emptyset$
for $p_1 \in \mathcal{P}_{out} \cup \mathcal{P}_{mid}$ **do**
  if $\left|\left\{p \in (\mathcal{P}_{out} \cup \mathcal{P}_{mid}) \setminus \{p_1\} \mid \exists s \in \mathcal{S} : a^{\text{f}}_{s,p}, a^{\text{f}}_{s,p_1} \neq 0\right\}\right| = 1$ **then**
    $\mathcal{P}_{\text{deg\_1}} := \mathcal{P}_{\text{deg\_1}} \cup p_1$
  else if $\left\{p \in (\mathcal{P}_{out} \cup \mathcal{P}_{mid}) \setminus \{p_1\} | \ \exists s \in \mathcal{S} : a^{\text{f}}_{s,p}, a^{\text{f}}_{s,p_1} \neq 0\right\} = \{p_2, p_3\} \wedge$
  $\left|\left\{p \in (\mathcal{P}_{out} \cup \mathcal{P}_{mid}) \setminus \{p_2\} \mid \exists s \in \mathcal{S} : a^{\text{f}}_{s,p}, a^{\text{f}}_{s,p_2} \neq 0\right\}\right| \geq 3 \wedge$
  $\left|\left\{p \in (\mathcal{P}_{out} \cup \mathcal{P}_{mid}) \setminus \{p_3\} \mid \exists s \in \mathcal{S} : a^{\text{f}}_{s,p}, a^{\text{f}}_{s,p_3} \neq 0\right\}\right| \geq 3$ **then**
    $\mathcal{P}_{\text{deg\_2}} := \mathcal{P}_{\text{deg\_2}} \cup p_1$
$\mathcal{P}_{\text{pos\_link}} := (\mathcal{P}_{out} \cup \mathcal{P}_{mid}) \setminus (\mathcal{P}_{\text{deg\_1}} \cup \mathcal{P}_{\text{deg\_2}})$
for $p_1 \neq p_2 \in \mathcal{P}_{\text{pos\_link}}$ **do**
  if $\exists s \in \mathcal{S} : a^{\text{f}}_{s,p_1} \neq 0 \wedge a^{\text{f}}_{s,p_2} \neq 0$ **then**
    $\mathcal{E} := \mathcal{E} \cup \{p_1, p_2\}$
  else if $\exists p_3 \in \mathcal{P}_{\text{deg\_2}}, s_1 \neq s_2 \in \mathcal{S} : (a^{\text{f}}_{s_1,p_1}, a^{\text{f}}_{s_1,p_3}, a^{\text{f}}_{s_2,p_2}, a^{\text{f}}_{s_2,p_3} \neq 0)$ **then**
    $\mathcal{E} := \mathcal{E} \cup \{p_1, p_2\}$

links must be a vertex-cut of this graph. The components into which a graph decomposes by the vertex-cut $\mathcal{P}_{\text{link}}$ will define parts of the network that can be aggregated, i.e., for which the input/output-profit/cost function c can be approximated. This means, the set $\mathcal{P}_{\text{b}}$ of a subproblem's products is given by the nodes of a hypergraph component. Since a vertex-cut of the reduced graph of possible links, $\mathcal{P}_{\text{pos\_link}}$, is also one of the simple undirected graph of products $\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$, it is, by theorem 4.5.19, likewise a vertex-cut of the hypergraph of products $\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$. Thus, depending on size and structure of the problem, and on the preferred size of components, one can either partition the simple undirected graph of nodes $\mathcal{P}_{\text{pos\_link}}$, or the simple undirected graph of nodes $\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$, which are both described in the preceding section. The acceptable size of a vertex-cut depends on the problem. Conceivable objectives of the partitioning problem are the detection of a minimum vertex-cut or of a cut of predetermined size that leads to highest graph fragmentation.
In our showcase model, we allow up to two vertices between the subsystem of products that shall be aggregated and the rest of the network, i.e., we request $|\mathcal{P}_{\text{link}}| \leq 2$. Since our example is relatively small, we can identify such subsets of nodes manually. They are visualized in figure 6.16.
In the following, we sketch algorithms that find such decompositions automatically.

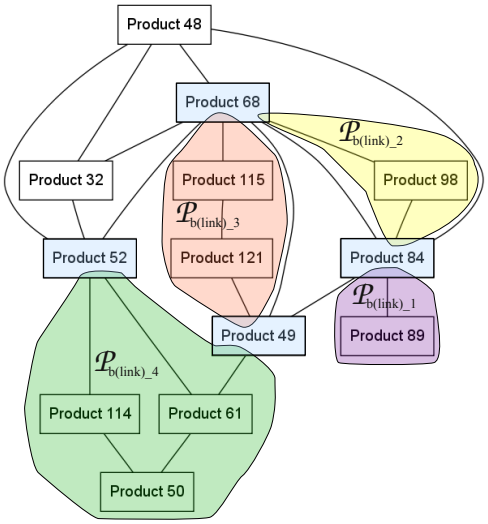

Figure 6.16: *Major network components of degree 1 and 2*

Many different methods from graph theory support an automated identification of vertex-cuts and associated components, cf. chapter 4. Regarding the partitioning of our network of linking products, the following ones are very convenient. Which of these methods should be chosen in an application strongly depends on the problem's size and structure as well as on the desired properties of a partition.

- In sparse networks, algorithms for the *identification of biconnected and triconnected components* of the graph can be useful, cf. section 4.4.
- Large networks can be *successively bipartitioned*, for example by multiply computing minimum or balanced vertex-separators. Such common graph partitioning problems are discussed in section 4.5.
- Given a predetermined link degree k, the problem of *identifying a graph's k key players*, presented in section 4.6, meets our requirements well.[8] A high fragmentation of the residual graph, obtained by removing the links, suggests relatively easily solvable subproblems.
- Vertices of high *centrality*, cf. section 4.3, represent reasonable candidates for linking nodes, which motivates heuristic methods.

Regarding the last point, betweenness centrality is obviously the most appropriate centrality measure to detect linking products. Therefore, we propose the following approach to an automated identification of network parts that are suited for aggregation:

1. Compute the betweenness centrality $c_B(v)$ of each vertex v of the graph.[9]

[8] k should at least equal the vertex-connectivity κ of the graph.

[9] To calculate betweenness centralities, you can use, e.g., the respective algorithm of the Boost Graph Library (BGL), cf. [SLL02].

2. Order the vertices by their centrality and search for a gap that separates around 20-30% of the vertices with highest betweenness centrality.

3. Check if removing subsets of the chosen vertices (and all their connecting edges) decomposes the graph into (at least two) components.[10]

If a vertex-cut is found, it can be used as set of links, and our aggregation procedure proposed previously can be applied to one or several components that the vertex-cut induces. First, however, the decomposition of the network of products $\mathcal{P}_{\text{pos\_link}}$, must be transferred to the original optimization problem. This means, given a set of links $\mathcal{P}_{\text{link}}$ that is a narrow vertex-cut separating a set of products $\mathcal{P}_{\text{b(link)}}$ from the network of products $\mathcal{P}_{\text{pos\_link}}$, we must reconstruct the sets $\mathcal{P}_{\text{b}}$ and $\mathcal{S}_{\text{b}}$, which define a subproblem as described in section 6.2. We will discuss this in subsection c., but first let us consider the decomposition of our showcase problem in the following example.

Example 6.4.2. Computing and ordering the betweenness centralities for the network of products $\mathcal{P}_{\text{pos\_link}}$, shown in figure 6.15, leads to the following values:

Linking product	68:	23.17	(0.35)	Linking product	115:	2.25	(0.03)
Linking product	52:	15.75	(0.24)	Linking product	50:	1.08	(0.02)
Linking product	49:	14.92	(0.23)	Linking product	121:	1.08	(0.02)
Linking product	84:	13.75	(0.21)	Linking product	32:	0	(0)
Linking product	61:	9.67	(0.15)	Linking product	98:	0	(0)
Linking product	48:	3	(0.05)	Linking product	89:	0	(0)
Linking product	114:	2.33	(0.04)				

Removing either one or two of the blue colored network products, leads to exactly the network decomposition shown in figure 6.16.
Let us compare these results with the vertex betweenness centralities of the network of products $\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$, shown in figure 6.14:

Linking product	84:	122.8	(0.49)	Linking product	98:	3.70	(0.01)
Linking product	68:	95.04	(0.38)	Linking product	32:	0	(0)
Linking product	52:	61.54	(0.24)	Linking product	54:	0	(0)
Linking product	49:	57.46	(0.23)	Linking product	55:	0	(0)
Linking product	89:	43.50	(0.17)	Linking product	57:	0	(0)
Linking product	114:	24.73	(0.10)	Linking product	78:	0	(0)
Linking product	48:	21.00	(0.08)	Linking product	88:	0	(0)
Linking product	61:	18.11	(0.07)	Linking product	102:	0	(0)
Linking product	115:	16.83	(0.07)	Linking product	105:	0	(0)
Linking product	35:	12.29	(0.05)	Linking product	176:	0	(0)
Linking product	121:	9.67	(0.04)	Linking product	179:	0	(0)
Linking product	50:	4.33	(0.02)	Linking product	249:	0	(0)

[10]The BGL [SLL02] also provides an algorithm that determines the connected components of a graph.

Except for product 35, the centrality index of each product $p \in \mathcal{P}_{\text{deg\_1}} \bigcup \mathcal{P}_{\text{deg\_2}}$ is zero. This is because the two neighbors of each product $p \in \mathcal{P}_{\text{deg\_2}}$ are connected by an edge, except for those of product 35. Similarly, the adjacency of the three neighbors of product 32 is the reason for its zero betweenness centrality.
The most remarkable consequence of removing products $\mathcal{P}_{\text{deg\_1}} \bigcup \mathcal{P}_{\text{deg\_2}}$ from the network, is the decrease of product 89's centrality. Regarding the small size of components that result using either only product 89 or product 89 and product 84 as links, cf. figure 6.14, it is reasonable to disregard product 89 as linking product and to compute centralities in the network of products $\mathcal{P}_{\text{pos\_link}}$. However, if the objective is also to determine very small model components, one should compute centralities and search for vertex-cuts in the network of products $\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$, since not every vertex-cut of this network is one in the smaller network of products $\mathcal{P}_{\text{pos\_link}}$.

Remark 6.4.3. In general, the graphical representation of the network optimization problem facilitates many interesting analyses. Depending on the application and the questions that one aims to answer, it can be advantageous, for instance, to aggregate a roughly prespecified area of the network model. This can be a component including certain products. Therefore, one could search for as less linking products as possible whose removal separates a component that includes these predetermined products. Furthermore, it could be useful to restrict such a component to a predetermined maximum or minimum size.

c. Transferring the Decomposition of the Network of Linking Products back to the Original Model

The final step consists in transferring the decomposition of the network of products $\mathcal{P}_{\text{pos\_link}}$ to the original optimization problem. Given a set of links $\mathcal{P}_{\text{link}}$ that is a narrow vertex-cut separating a set of products $\mathcal{P}_{\text{b(link)}}$ from the network of products $\mathcal{P}_{\text{pos\_link}}$, we aim to reconstruct compatible sets $\mathcal{P}_{\text{b}}$ and $\mathcal{S}_{\text{b}}$, which define a subproblem as described in section 6.2.
We will see that, given $\mathcal{P}_{\text{link}}$ and $\mathcal{P}_{\text{b(link)}}$, the choice of $\mathcal{S}_{\text{b}}$ and also $\mathcal{P}_{\text{b}}$ is not necessarily unique, but, in general, there are different possibilities to construct the submodel.[11] These are bounded by a certain minimal set of submodel processes $\mathcal{S}_{\text{b\_min}}$ and a certain maximal set of submodel processes $\mathcal{S}_{\text{b\_max}}$.
To reconstruct a complete submodel, we pass through the following steps:

1. Let $\mathcal{P}^{\text{adj}}_{\text{b(link)}} \subseteq \mathcal{P}_{\text{deg\_1}} \bigcup \mathcal{P}_{\text{deg\_2}}$ denote the subset of products of $\mathcal{P}_{\text{deg\_1}} \bigcup \mathcal{P}_{\text{deg\_2}}$ that are adjacent to a product $p \in \mathcal{P}_{\text{b(link)}}$ within the graph of products $\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$. By this definition, each product $p \in \mathcal{P}^{\text{adj}}_{\text{b(link)}}$ appears in at least one process that also includes a product of $\mathcal{P}_{\text{b(link)}}$. We define $\mathcal{P}_{\text{b\_min}} := \mathcal{P}_{\text{b(link)}} \bigcup \mathcal{P}^{\text{adj}}_{\text{b(link)}}$.

[11] Regarding hypergraph partitioning, the different possibilities to choose sets $\mathcal{S}_{\text{b}}$ correspond to the assignment of free nodes to components, cf. section 4.5. We discuss this connection in more detail in remark 6.4.5.

2. Let $\mathcal{S}_p^{\text{inc}}$ denote the set of all processes $s \in \mathcal{S}$ that include product p. Then, a submodel compatible to $\mathcal{P}_{\text{link}}$ and $\mathcal{P}_{\text{b(link)}}$ must include at least the processes

$$\mathcal{S}_{\text{b\_min}} := \bigcup_{p \in \mathcal{P}_{\text{b\_min}}} \mathcal{S}_p^{\text{inc}}. \tag{6.26}$$

3. Let $\mathcal{P}_{\text{link}}^{\text{adj}} \subseteq (\mathcal{P}_{\text{deg\_1}} \bigcup \mathcal{P}_{\text{deg\_2}}) \setminus \mathcal{P}_{\text{b(link)}}^{\text{adj}}$ denote the subset of products of $\mathcal{P}_{\text{deg\_1}} \bigcup \mathcal{P}_{\text{deg\_2}}$ that are solely adjacent to products of $\mathcal{P}_{\text{link}}$ within the graph of products $\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$. By this definition, there does not exist any product $p \in \mathcal{P}_{\text{link}}^{\text{adj}}$ appearing in a process that also includes a product of $\mathcal{P}_{\text{pos\_link}} \setminus \mathcal{P}_{\text{link}}$. We define $\mathcal{P}_{\text{b\_max}} := \mathcal{P}_{\text{b\_min}} \bigcup \mathcal{P}_{\text{link}}^{\text{adj}}$.

4. The largest possible submodel compatible to $\mathcal{P}_{\text{link}}$ and $\mathcal{P}_{\text{b(link)}}$, called $\mathcal{S}_{\text{b\_max}}$, contains, in addition to the processes of $\mathcal{S}_{\text{b\_min}}$, those processes that include linking products of the submodel and products of the set $\mathcal{P}_{\text{link}}^{\text{adj}}$ but no further possible linking products:

$$\mathcal{S}_{\text{b\_max}} := \mathcal{S}_{\text{b\_min}} \cup \Big(\bigcup_{p \in \mathcal{P}_{\text{link}} \cup \mathcal{P}_{\text{link}}^{\text{adj}}} \mathcal{S}_p^{\text{inc}} \setminus (\bigcup_{p \in \mathcal{P}_{\text{pos\_link}} \setminus \mathcal{P}_{\text{link}}} \mathcal{S}_p^{\text{inc}}) \Big). \tag{6.27}$$

5. Having defined these sets, we can now choose S_{b} such that

$$\mathcal{S}_{\text{b\_min}} \subseteq \mathcal{S}_{\text{b}} \subseteq \mathcal{S}_{\text{b\_max}}, \tag{6.28}$$

   where $s_1, s_2 \in \mathcal{S}_{\text{b\_max}} \setminus \mathcal{S}_{\text{b\_min}}$ with $a_{s_1,p}^{\text{f}} \neq 0$ and $a_{s_2,p}^{\text{f}} \neq 0$ for a $p \in \mathcal{P}_{\text{link}}^{\text{adj}}$ must be either both included in the submodel ($s_1, s_2 \in \mathcal{S}_{\text{b}}$) or not ($s_1, s_2 \in \mathcal{S} \setminus \mathcal{S}_{\text{b}}$).

   Remark 6.4.4. If we would like to use multiple submodel aggregations with overlapping sets of linking products $\mathcal{P}_{\text{link\_1}}$,...,$\mathcal{P}_{\text{link\_n}}$ to simulate the whole network model, it must be ensured that the aggregated sets of submodel processes and corresponding product sets are disjoint: $\mathcal{S}_{\text{b\_1}} \bigcap ... \bigcap \mathcal{S}_{\text{b\_n}} = \emptyset$ and $\mathcal{P}_{\text{b\_1}} \bigcap ... \bigcap \mathcal{P}_{\text{b\_n}} = \emptyset$.

6. Given $\mathcal{S}_{\text{b}}$, $\mathcal{P}_{\text{b}}$ consists of all products $p \in (\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}) \setminus \mathcal{P}_{\text{link}}$ that are included in at least one of the processes $s \in \mathcal{S}_{\text{b}}$.
   For example, if $\mathcal{S}_{\text{b}} = \mathcal{S}_{\text{b\_min}}$, we have $\mathcal{P}_{\text{b}} = \mathcal{P}_{\text{b\_min}}$, and if $\mathcal{S}_{\text{b}} = \mathcal{S}_{\text{b\_max}}$, we have $\mathcal{P}_{\text{b}} = \mathcal{P}_{\text{b\_max}}$.

The above steps can be realized by algorithm 6.4.2 and 6.4.3. Algorithm 6.4.2 constructs the sets $\mathcal{S}_{\text{b\_min}}$ and $\mathcal{P}_{\text{b\_min}}$ as well as $\mathcal{S}_{\text{b\_max}}$ and $\mathcal{P}_{\text{b\_max}}$. Given any set $\mathcal{S}_{\text{b\_chosen}}$ that fulfills $\mathcal{S}_{\text{b\_min}} \subset \mathcal{S}_{\text{b\_chosen}} \subset \mathcal{S}_{\text{b\_max}}$, algorithm 6.4.3 sets up a minimal submodel, represented by $\mathcal{S}_{\text{b}}$ and $\mathcal{P}_{\text{b}}$, that includes $\mathcal{S}_{\text{b\_chosen}}$ and is suited for aggregation.

We will now verify that

a) the network definition by algorithm 6.4.1,

b) the choice of a narrow vertex-cut $\mathcal{P}_{\text{link}} \subset \mathcal{P}_{\text{pos\_link}}$ that separates a component $\mathcal{P}_{\text{b(link)}} \subset \mathcal{P}_{\text{pos\_link}}$ from the network of possible linking products, and

c) algorithms 6.4.2 and 6.4.3

jointly lead, as requested, to a set $\mathcal{S}_{\text{b}}$ of processes that is only connected to the remaining processes $\mathcal{S} \setminus \mathcal{S}_{\text{b}}$ by linking products, i.e., $\mathcal{S}_{\text{b}}$ and $\mathcal{S} \setminus \mathcal{S}_{\text{b}}$ do not have any product of the set $(\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}) \setminus \mathcal{P}_{\text{link}}$ in common:
Let $s_1, s_2 \in \mathcal{S}, p \in \mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$ with $a^{\text{f}}_{s_1,p} \neq 0$ and $a^{\text{f}}_{s_2,p} \neq 0$. We have to show that

Algorithm 6.4.2: Retransferring a Decomposition to the Original Model (Part 1)

Input: Disjoint product sets $\mathcal{P}_{out}$, $\mathcal{P}_{mid}$ and set of processes $\mathcal{S}$ that build a connected, consistent model. Nonempty set of possible linking products $\mathcal{P}_{\text{pos\_link}} \subseteq \mathcal{P}_{out} \bigcup \mathcal{P}_{mid}$ with nonempty subset $\mathcal{P}_{\text{link}} \subset \mathcal{P}_{\text{pos\_link}}$ of chosen linking products, which is a narrow separator with regard to the corresponding network component $\mathcal{P}_{\text{b(link)}} \subset \mathcal{P}_{\text{pos\_link}}$.

Output: Minimal sets $\mathcal{S}_{\text{b\_min}}$ and $\mathcal{P}_{\text{b\_min}}$ that build a submodel with respect to $\mathcal{P}_{\text{link}}$, which is suited for aggregation. Maximal sets $\mathcal{S}_{\text{b\_max}}$ and $\mathcal{P}_{\text{b\_max}}$ that build a submodel with respect to $\mathcal{P}_{\text{link}}$, which is suited for aggregation.

$\mathcal{S}_{\text{b\_min}} = \mathcal{P}^{\text{adj}}_{\text{b(link)}} = \mathcal{P}^{\text{adj}}_{\text{link}} = \emptyset$
for $p \in (\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}) \setminus \mathcal{P}_{\text{pos\_link}}$ **do**
  if $\exists\ s \in \mathcal{S}, p_1 \in \mathcal{P}_{\text{b(link)}} : a^{\text{f}}_{s,p}, a^{\text{f}}_{s,p_1} \neq 0$ **then**
    $\mathcal{P}^{\text{adj}}_{\text{b(link)}} := \mathcal{P}^{\text{adj}}_{\text{b(link)}} \cup p$
$\mathcal{P}_{\text{b\_min}} := \mathcal{P}_{\text{b(link)}} \bigcup \mathcal{P}^{\text{adj}}_{\text{b(link)}}$
for $s \in \mathcal{S}$ **do**
  if $\exists\ p \in \mathcal{P}_{\text{b\_min}} : a^{\text{f}}_{s,p} \neq 0$ **then**
    $\mathcal{S}_{\text{b\_min}} := \mathcal{S}_{\text{b\_min}} \cup s$
for $p \in (\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}) \setminus (\mathcal{P}_{\text{pos\_link}} \bigcup \mathcal{P}^{\text{adj}}_{\text{b(link)}})$ **do**
  if $\nexists\ s \in \mathcal{S}, p_1 \in \mathcal{P}_{\text{pos\_link}} \setminus \mathcal{P}_{\text{link}} : a^{\text{f}}_{s,p}, a^{\text{f}}_{s,p_1} \neq 0$ **then**
    $\mathcal{P}^{\text{adj}}_{\text{link}} := \mathcal{P}^{\text{adj}}_{\text{link}} \cup p$
$\mathcal{P}_{\text{b\_max}} := \mathcal{P}_{\text{b\_min}} \bigcup \mathcal{P}^{\text{adj}}_{\text{link}}$
$\mathcal{S}_{\text{b\_max}} := \mathcal{S}_{\text{b\_min}}$
for $s \in \mathcal{S} \setminus \mathcal{S}_{\text{b\_min}}$ **do**
  if $\forall\ p_{\text{pos\_link}} \in \mathcal{P}_{\text{pos\_link}} \setminus \mathcal{P}_{\text{link}} : a^{\text{f}}_{s,p_{\text{pos\_link}}} = 0 \wedge \exists\ p \in \mathcal{P}_{\text{link}} \bigcup \mathcal{P}^{\text{adj}}_{\text{b(link)}} : a^{\text{f}}_{s,p} \neq 0$ **then**
    $\mathcal{S}_{\text{b\_max}} := \mathcal{S}_{\text{b\_max}} \cup s$

Algorithm 6.4.3: Retransferring a Decomposition to the Original Model (Part 2)

Input: Minimal sets $\mathcal{S}_{\text{b\_min}}$ and $\mathcal{P}_{\text{b\_min}}$ as well as maximal sets $\mathcal{S}_{\text{b\_max}}$ and $\mathcal{P}_{\text{b\_max}}$ that build submodels suited for aggregation, cf. output of algorithm 6.4.2. Chosen set $\mathcal{S}_{\text{b\_chosen}}$ fulfilling $\mathcal{S}_{\text{b\_min}} \subset \mathcal{S}_{\text{b\_chosen}} \subset \mathcal{S}_{\text{b\_max}}$.

Output: Sets $\mathcal{S}_{\text{b}}$ and $\mathcal{P}_{\text{b}}$ that build a minimal submodel that includes $\mathcal{S}_{\text{b\_chosen}}$ and is suited for aggregation.

$\mathcal{S}_{\text{b}} := \mathcal{S}_{\text{b\_chosen}}, \mathcal{P}_{\text{b}} := \mathcal{P}_{\text{b\_min}}$
for $p \in \mathcal{P}_{\text{b\_max}} \setminus \mathcal{P}_{\text{b\_min}}$ **do**
  if $\exists\ s_1 \in \mathcal{S}_{\text{b\_chosen}} : a^{\text{f}}_{s_1,p} \neq 0$ **then**
    $\mathcal{P}_{\text{b}} := \mathcal{P}_{\text{b}} \cup p$
    for $s_2 \in \mathcal{S}_{\text{b\_max}} \setminus \mathcal{S}_{\text{b\_chosen}}$ **do**
      if $a^{\text{f}}_{s_2,p} \neq 0$ **then**
        $\mathcal{S}_{\text{b}} := \mathcal{S}_{\text{b}} \cup s_2$

$p \notin \mathcal{P}_{\text{link}} \Rightarrow s_1, s_2 \in \mathcal{S}_{\text{b}}$ or $s_1, s_2 \in \mathcal{S} \setminus \mathcal{S}_{\text{b}}$.
Case 1: $p \in \mathcal{P}_{\text{b(link)}} \Rightarrow s_1, s_2 \in \mathcal{S}_{\text{b}}$
Case 2: $p \in \mathcal{P}_{\text{pos\_link}} \setminus (\mathcal{P}_{\text{link}} \bigcup \mathcal{P}_{\text{b(link)}}) \Rightarrow s_1, s_2 \in \mathcal{S} \setminus \mathcal{S}_{\text{b}}$
Case 3: $p \in (\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}) \setminus \mathcal{P}_{\text{pos\_link}}$, i.e., $p \in \mathcal{P}_{\text{deg\_1}} \bigcup \mathcal{P}_{\text{deg\_2}}$
$\Rightarrow$ since the original model is connected, by algorithm 6.4.1, $\exists s_3 \in \mathcal{S}, p_1 \in \mathcal{P}_{\text{pos\_link}} : a^{\text{f}}_{s_3,p}, a^{\text{f}}_{s_3,p_1} \neq 0$
$\Rightarrow$ by the definition of $\mathcal{S}_{\text{b\_min}}$ and $\mathcal{S}_{\text{b\_max}}$ in algorithm 6.4.2 and the condition on the choice of $\mathcal{S}_{\text{b}}$ in algorithm 6.4.3, we have either $s_1, s_2, s_3 \in \mathcal{S}_{\text{b}}$ or $s_1, s_2, s_3 \in \mathcal{S} \setminus \mathcal{S}_{\text{b}}$ with the following case differentiation:
Case 3.1: $p_1 \in \mathcal{P}_{\text{b(link)}} \Rightarrow p \in \mathcal{P}^{\text{adj}}_{\text{b(link)}} \Rightarrow s_1, s_2, s_3 \in \mathcal{S}_{\text{b\_min}} \subseteq \mathcal{S}_{\text{b}}$
Case 3.2: $p_1 \in \mathcal{P}_{\text{pos\_link}} \setminus (\mathcal{P}_{\text{link}} \bigcup \mathcal{P}_{\text{b(link)}}) \Rightarrow s_1, s_2, s_3 \in \mathcal{S} \setminus \mathcal{S}_{\text{b\_max}}$
Case 3.3: $p_1 \in \mathcal{P}_{\text{link}} \Rightarrow s_1, s_2, s_3 \in \mathcal{S}_{\text{b}}$ or $s_1, s_2, s_3 \in \mathcal{S} \setminus \mathcal{S}_{\text{b}}$

Table 6.14 contains the sets $\mathcal{P}_{\text{b\_min}}$ and $\mathcal{P}_{\text{b\_max}}$ related to the two components of the network shown in figure 6.16 that are represented by the sets $\mathcal{P}_{\text{b(link)\_1}} = \{89\}$ and $\mathcal{P}_{\text{b(link)\_2}} = \{98\}$. Note that, if both submodels are simultaneously aggregated, only one of them may include product 179, cf. remark 6.4.4
Figure 6.17 shows a network partition of the whole petrochemical model, which corresponds to that of the possible linking products in figure 6.16. The products and

$\mathcal{P}_{\text{link}}$	$\mathcal{P}_{\text{b(link)}}$	$\mathcal{P}_{\text{b\_min}}$	$\mathcal{P}_{\text{b\_max}}$
$\{84\}$	$\{89\}$	$\{54, 57, 89, 102, 249\}$	$\{54, 57, 89, 102, 249, 179\}$
$\{68, 84\}$	$\{98\}$	$\{55, 98\}$	$\{55, 98, 78, 179\}$

Table 6.14: *Exemplary sets of submodel products regarding the networks shown in figures 6.14 and 6.16*

processes highlighted in violet match component $\mathcal{P}_{\text{b(link)\_1}}$, the yellow ones correspond to $\mathcal{P}_{\text{b(link)\_2}}$, the orange products and processes fit $\mathcal{P}_{\text{b(link)\_3}}$ and the green ones fit component $\mathcal{P}_{\text{b(link)\_4}}$. The blue colored linking products interconnect these components as well as the remaining part of the network. We decide to assign the products 78 and 179 as well as all processes only including the linking products 68 and 84 to the second submodel, highlighted in yellow. The single process only including product 49 is assigned to the fourth subproblem, which is highlighted in green. The remaining linking product 52 is not contained in any freely assignable process.

Besides the appropriate definition of model components, the choice of the varying constraints $\boldsymbol{y}_1^{\text{q}}, ..., \boldsymbol{y}_n^{\text{q}}$ is crucial in order to aggregate the subproblem of optimizing the production quantities related to processes $\mathcal{S}_{\text{b}}$. A convenient way to choose them will be discussed in the following section.

Remark 6.4.5. (Free node assignment) Let us consider the bipartite graph of figure 6.1, and its induced hypergraph H with primal variables (corresponding to processes) as nodes and dual variables (corresponding to products) as hyperedges, which are also called nets, cf. section 4.1. We call its net intersection graph G_{NIG}. Mostly, a vertex-cut $\mathcal{P}_{\text{link}}$ of the undirected graph of products $\mathcal{P}_{\text{pos\_link}}$, defined in paragraph a), is also a vertex-cut of G_{NIG} and, therefore, a net-cut of H. However, the node partition of H by this net-cut is not necessarily uniquely determined. Processes that only include products of the cut-set $\mathcal{P}_{\text{pos\_link}}$ represent "free" nodes. In some cases, they can be assigned to different parts of the graph, cf. hypergraph partitioning in section 4.5.

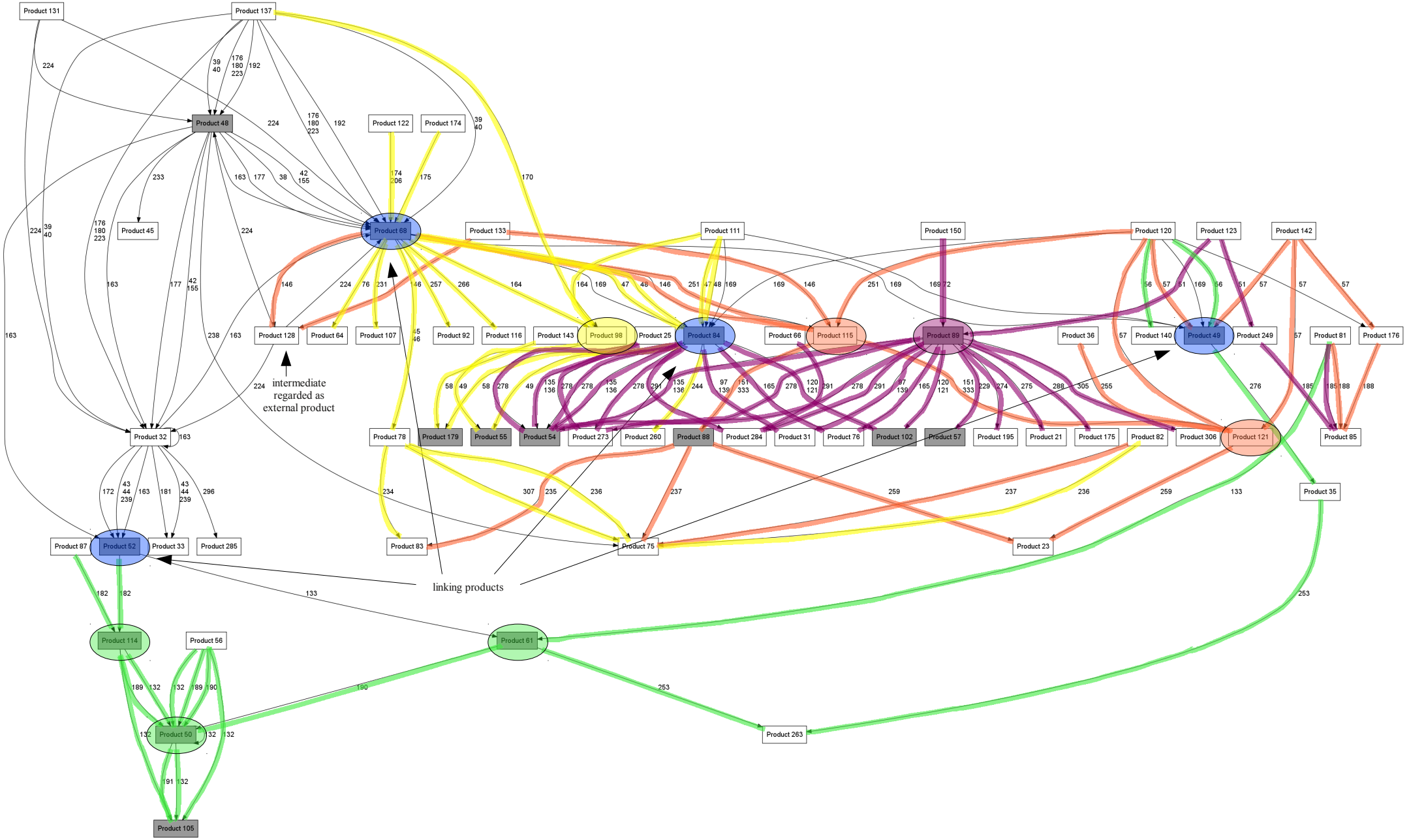

Figure 6.17: *Network partition into four submodel components*

6.5 Approximating the Optimal Value Function of a Subproblem

In this section, we discuss on which domain the optimal value function of a subproblem should be estimated and study its properties on this domain, cf. also the general approach presented in section 5.3. Afterwards, we propose a parametric model for the approximating input/output-profit/cost function c to fit the generated submodel data $(\boldsymbol{y}_i^{\mathrm{q}}, \pi_i)_{i=1,\dots,n}$. Finally, we present a different formulation of the subproblem, which can alternatively be used to calculate subproblem data. It is strongly related to the subproblem's Lagrangian dual problem.

a. Determining and Discretizing the Relevant Approximation Domain

The choice of the grid $\boldsymbol{y}_1^{\mathrm{q}}, ..., \boldsymbol{y}_n^{\mathrm{q}}$, with $\boldsymbol{y}_i^{\mathrm{q}} \in D \subset \mathbb{R}^{|\mathcal{P}_{\mathrm{link}}|}$, and, in particular, the choice of the domain that it covers, is crucial for a suitable approximation of a subproblem's optimal value function and, consequently, for an appropriate aggregation of the subproblem by means of an input/output-profit/cost function c. Since a wide approximation domain can lead to a locally bad approximation, we try to restrict the domain, over which we generate data by computing subproblem solutions, as much as possible. In the following, we discuss structural restrictions and plaubsible restrictions of this domain.

Structural restrictions through process capacities

Given a submodel with linking products $\mathcal{P}_{\mathrm{link}}$ and processes $\mathcal{S}_{\mathrm{b}}$, lower and upper bounds for input/output of $p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}$ with respect to the submodel are given by

$$\begin{aligned} y_{p_{\mathrm{link}}}^{\mathrm{q},l} &= \max\Big(-\sum_{s\in\mathcal{S}_{\mathrm{b}}} a_s^{\mathrm{cap}} \cdot \max(0, a_{s,p_{\mathrm{link}}}^{\mathrm{f}}),\ -a_{p_{out}}^{\mathrm{con\_max}} + \sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}} a_s^{\mathrm{cap}} \cdot \min(a_{s,p_{\mathrm{link}}}^{\mathrm{f}}, 0)\Big) \\ &\leq 0 \end{aligned} \tag{6.29}$$

and

$$\begin{aligned} y_{p_{\mathrm{link}}}^{\mathrm{q},u} &= \min\Big(-\sum_{s\in\mathcal{S}_{\mathrm{b}}} a_s^{\mathrm{cap}} \cdot \min(a_{s,p_{\mathrm{link}}}^{\mathrm{f}}, 0),\ \sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}} a_s^{\mathrm{cap}} \cdot \max(0, a_{s,p_{\mathrm{link}}}^{\mathrm{f}})\Big) \\ &\geq 0. \end{aligned} \tag{6.30}$$

The lower bound $y_{p_{\mathrm{link}}}^{\mathrm{q},l}$ is (the negative of) the minimum of two quantities: first, the quantity of product p_{link} that can be produced in the submodel and, second, the quantity of it that can be processed and sold in the remaining part of the model. Accordingly, $y_{p_{\mathrm{link}}}^{\mathrm{q},u}$ is the minimum of the quantity that can be processed in the submodel and the one that can be produced in the remaining network. We define a domain $D \subset \mathbb{R}^{|\mathcal{P}_{\mathrm{link}}|}$ corresponding to these bounds by

$$D := \{\boldsymbol{y}^{\mathrm{q}} = (y_{p_{\mathrm{link\_1}}}^{\mathrm{q}}, ..., y_{p_{\mathrm{link\_}|\mathcal{P}_{\mathrm{link}}|}}^{\mathrm{q}})^{\mathrm{T}} \in \mathbb{R}^{|\mathcal{P}_{\mathrm{link}}|} \mid y_{p_{\mathrm{link}}}^{\mathrm{q},l} \leq y_{p_{\mathrm{link}}}^{\mathrm{q}} \leq y_{p_{\mathrm{link}}}^{\mathrm{q},u} \,\forall\, p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}\}. \tag{6.31}$$

However, in general, the input/output quantities $y^{\mathrm{q}}_{p_{\mathrm{link}}}$, $p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}$, cannot be chosen independently: there can be input/output values $\boldsymbol{y} \in D$ that lead to an empty set of feasible points for the optimization problem (6.13). This means that, possibly, we have to restrict D to ensure feasibility. Therefore, we define the smaller set

$$\tilde{D} := \{\boldsymbol{y}^{\mathrm{q}} \in D \mid \mathcal{F}_1(\boldsymbol{y}^{\mathrm{q}}) \neq \emptyset\}, \tag{6.32}$$

where

$$\mathcal{F}_1(\boldsymbol{y}^{\mathrm{q}}) = \left\{ \boldsymbol{x}^{\mathrm{q}} \in \mathbb{R}^{|\mathcal{S}_{\mathrm{b}}|} \;\middle|\; \begin{array}{ll} \forall\, s \in \mathcal{S}_{\mathrm{b}} : & 0 \leq x^{\mathrm{q}}_s \leq a^{\mathrm{cap}}_s \\ \forall\, p \in (\mathcal{P}_{out} \dot{\bigcup} \mathcal{P}_{mid}) \bigcap \mathcal{P}_{\mathrm{b}} : & 0 \leq \sum_{s \in \mathcal{S}_{\mathrm{b}}} x^{\mathrm{q}}_s \cdot a^{\mathrm{f}}_{s,p} \\ \forall\, p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}} : & 0 \leq y^{\mathrm{q}}_{p_{\mathrm{link}}} + \sum_{s \in \mathcal{S}_{\mathrm{b}}} x^{\mathrm{q}}_s \cdot a^{\mathrm{f}}_{s,p_{\mathrm{link}}} \end{array} \right\} \tag{6.33}$$

is the feasible set of optimization problem (6.13). Note that, for $\bar{\boldsymbol{y}}^{\mathrm{q}} \geq \boldsymbol{y}^{\mathrm{q}}$, i.e., $\bar{y}^{\mathrm{q}}_{p_{\mathrm{link}}} \geq y^{\mathrm{q}}_{p_{\mathrm{link}}} \forall\, p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}$, we have

$$\mathcal{F}_1(\boldsymbol{y}^{\mathrm{q}}) \subseteq \mathcal{F}_1(\bar{\boldsymbol{y}}^{\mathrm{q}}), \tag{6.34}$$

and therefore

$$\mathcal{F}_1(\boldsymbol{y}^{\mathrm{q}}) \neq \emptyset \Rightarrow \mathcal{F}_1(\bar{\boldsymbol{y}}^{\mathrm{q}}) \neq \emptyset. \tag{6.35}$$

Remark 6.5.1. $\tilde{D}$ is convex: let $\boldsymbol{y}^1, \boldsymbol{y}^2 \in \tilde{D}$, i.e., $\exists\, \boldsymbol{x}^1 \in \mathcal{F}_1(\boldsymbol{y}^1), \boldsymbol{x}^2 \in \mathcal{F}_1(\boldsymbol{y}^2)$ with

$$-\sum_{s \in \mathcal{S}_{\mathrm{b}}} x^1_s \cdot a^{\mathrm{f}}_{s,p_{\mathrm{link}}} \leq y^1_{p_{\mathrm{link}}},\; -\sum_{s \in \mathcal{S}_{\mathrm{b}}} x^2_s \cdot a^{\mathrm{f}}_{s,p_{\mathrm{link}}} \leq y^2_{p_{\mathrm{link}}}, \quad \forall\, p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}. \tag{6.36}$$

Then it holds that $\forall\, t \in [0,1]$

$$-\sum_{s \in \mathcal{S}_{\mathrm{b}}} (t x^1_s + (1-t) x^2_s) \cdot a^{\mathrm{f}}_{s,p_{\mathrm{link}}} \leq t y^1_{p_{\mathrm{link}}} + (1-t) y^2_{p_{\mathrm{link}}}, \quad \forall\, p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}, \tag{6.37}$$

where $t\boldsymbol{x}^1 + (1-t)\boldsymbol{x}^2$ is also feasible regarding the first two conditions of (6.33). This means, we have

$$t\boldsymbol{x}^1 + (1-t)\boldsymbol{x}^2 \in \mathcal{F}_1(t\boldsymbol{y}^1 + (1-t)\boldsymbol{y}^2), \tag{6.38}$$

i.e., $t\boldsymbol{y}^1 + (1-t)\boldsymbol{y}^2 \in \tilde{D}$.

The shape of the domain $\tilde{D}$, if $\tilde{D} \subset D$, is sketched in figure 6.18 for the two-dimensional case. Note that we have

$$\{\boldsymbol{y}^{\mathrm{q}} \in \mathbb{R}^{|\mathcal{P}_{\mathrm{link}}|} \mid 0 \leq y^{\mathrm{q}}_{p_{\mathrm{link}}} \leq y^{\mathrm{q},u}_{p_{\mathrm{link}}}\} \subseteq \tilde{D}. \tag{6.39}$$

Similarly, we can further exclude input/output values $\boldsymbol{y}^{\mathrm{q}} \in \tilde{D}$ that lead to an empty set of feasible points for optimization problem (6.14), since they cannot lead to a feasible

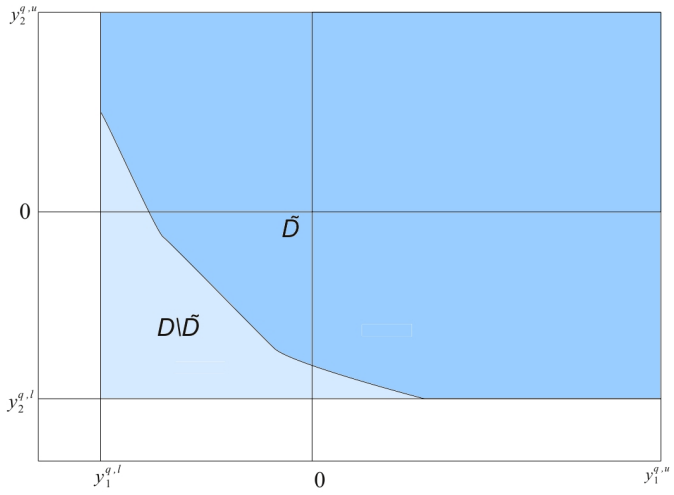

Figure 6.18: *The domain $\tilde{D}$ for $|\mathcal{P}_{\text{link}}| = 2$*

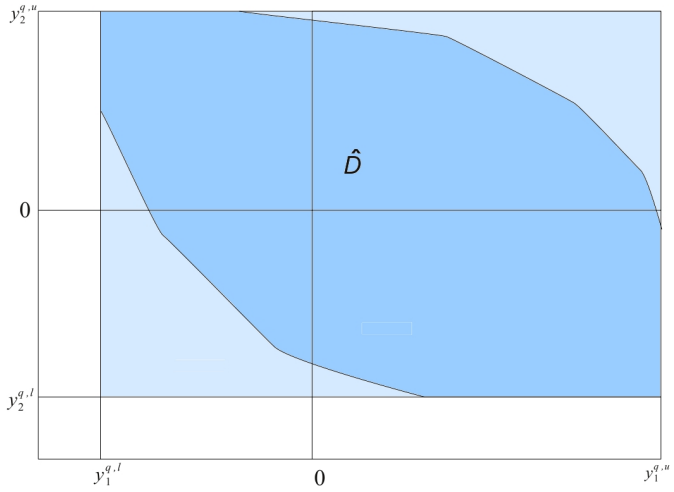

Figure 6.19: *The domain $\hat{D}$ for $|\mathcal{P}_{\text{link}}| = 2$*

point of the overall optimization problem. Given the feasible set of optimization problem (6.14) by

$$\mathcal{F}_2(\boldsymbol{y}^{\text{q}}) = \left\{ \boldsymbol{x}^{\text{q}} \in \mathbb{R}^{|\mathcal{S}\setminus\mathcal{S}_{\text{b}}|} \;\middle|\; \begin{array}{ll} \forall\ s \in \mathcal{S}\setminus\mathcal{S}_{\text{b}}: & 0 \le x_s^{\text{q}} \le a_s^{\text{cap}} \\ \forall\ p \in (\mathcal{P}_{out}\dot{\bigcup}\mathcal{P}_{mid})\setminus(\mathcal{P}_{\text{b}}\dot{\bigcup}\mathcal{P}_{\text{link}}): & 0 \le \sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\text{b}}} x_s^{\text{q}} \cdot a_{s,p}^{\text{f}} \\ \forall\ p_{\text{link}} \in \mathcal{P}_{\text{link}}: & y_{p_{\text{link}}}^{\text{q}} \le \sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\text{b}}} x_s^{\text{q}} \cdot a_{s,p_{\text{link}}}^{\text{f}} \end{array} \right\}, \tag{6.40}$$

we define,

$$\hat{D} := \{\boldsymbol{y}^{\text{q}} \in \tilde{D} \mid \mathcal{F}_2(\boldsymbol{y}^{\text{q}}) \neq \emptyset\}. \tag{6.41}$$

It is easy to show, that $\hat{D}$, as the intersection of convex sets, is also convex. Its shape is sketched in figure 6.19.

Depending on the size of the examined network in terms of the quantities of the linking products that can be produced and processed, the bounds (6.29) and (6.30), and even the restricted domains $\tilde{D}$ and $\hat{D}$, may admit a wide range on which the input/output-profit/cost function c can be estimated. Because of the prespecified parametric model for the function c, e.g., a polynomial of degree two, see below, such a wide domain can lead to a locally bad approximation of the data. For this reason, it can be worthwhile to further truncate the domain based on individual information about the order of magnitude of the optimal input or output of linking products with respect to the submodel. In the following paragraph, we sketch a possibility to further restrict the domain by considering the subproblem's Lagrange multipliers.

Plausible restrictions through shadow prices

Comparing optimal solutions of problem (6.13) with optimality conditions of problem (6.14), we can restrict the domain over which the function c should be estimated by some plausible conclusions. However, in contrary to the structural restrictions, in order to apply this approach, we must first generate submodel solutions $(\boldsymbol{y}_i^{\mathrm{q}}, F_1^*(\boldsymbol{y}_i^{\mathrm{q}}))$ by solving (6.13). Afterwards, we can exclude those data points that do not meet certain conditions regarding their Lagrange multipliers, which we will specify in the following.

Consider optimization problem (6.13)**:**
In an optimal solution of each subsystem optimization i, the vector $\boldsymbol{\lambda}_i^{\mathrm{sub}} \in \mathbb{R}^{|\mathcal{P}_{\mathrm{link}}|}$ of Lagrange multipliers corresponding to the constraints

$$\forall\ p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}} :\ 0 \leq y^{\mathrm{q}}_{p_{\mathrm{link}},i} + \sum_{s \in \mathcal{S}_{\mathrm{b}}} x_s^{\mathrm{q}} \cdot a^{\mathrm{f}}_{s,p_{\mathrm{link}}} \tag{6.42}$$

tells, if it is unique, i.e., under certain regularity conditions as the LICQ, cf. theorem 3.1.10, how much the optimal value of the objective F_1 would increase if the right hand side of the constraints was increased for one product $p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}$, respectively. This fact causes the interpretation of the Lagrange multipliers as shadow prices, cf. also corollary 3.2.12.
Under the assumptions of theorem 3.2.10 with respect to the parameter vectors $\boldsymbol{y}_i^{\mathrm{q}} = (y^{\mathrm{q}}_{p_{\mathrm{link\_1}},i}, ..., y^{\mathrm{q}}_{p_{\mathrm{link\_}|\mathcal{P}_{\mathrm{link}}|},i})^{\mathrm{T}}, i = 1, ..., n$, we have by corollary 3.2.12, in a neighborhood of $\boldsymbol{y}_i^{\mathrm{q}}$, for each $i = 1, ..., n$ and $p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}$

$$\lambda^{\mathrm{sub}}_{p_{\mathrm{link}},i} = \frac{\partial F_1^*}{\partial y^{\mathrm{q}}_{p_{\mathrm{link}},i}}(\boldsymbol{y}_i^{\mathrm{q}}), \tag{6.43}$$

where F_1^* is the optimal value function of problem (6.13), which we aim to fit by a function c, as described above. The differentiability of F_1^* will be further discussed in theorem 6.5.3.

Consider optimization problem (6.14)**:**
Formulating the objective function of problem (6.14) without the slack variables $\boldsymbol{x}^{\mathrm{sales}}_{\mathcal{P}_{\mathrm{link}}}$,

and with $c \equiv F_1^*$, we obtain

$$\begin{aligned}
&\max_{\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}},\boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_{\mathrm{link}}}} F_2(\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}},\boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_{\mathrm{link}}}) = \\
&\max_{\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}},\boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_{\mathrm{link}}}} \sum_{\substack{p_{out}\in\\ \mathcal{P}_{out}\setminus(\mathcal{P}_{\mathrm{b}}\bigcup\mathcal{P}_{\mathrm{link}})}} \Big(\sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}} x^{\mathrm{q}}_s\cdot a^{\mathrm{f}}_{s,p_{out}}\Big)\cdot P_{p_{out}}\Big(\sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}} x^{\mathrm{q}}_s\cdot a^{\mathrm{f}}_{s,p_{out}}\Big) + \sum_{\substack{p_{ex}\in\mathcal{P}_{ex},\\ s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}}} x^{\mathrm{q}}_s\cdot a^{\mathrm{f}}_{s,p_{ex}}\cdot a^{\pi}_{p_{ex}} \\
&\qquad + \sum_{p_{\mathrm{link}}\in\mathcal{P}_{\mathrm{link}}\bigcap\mathcal{P}_{out}} (z^{\mathrm{q}}_{p_{\mathrm{link}}} - x^{\mathrm{sub}}_{p_{\mathrm{link}}})\cdot P_{p_{\mathrm{link}}}(z^{\mathrm{q}}_{p_{\mathrm{link}}} - x^{\mathrm{sub}}_{p_{\mathrm{link}}}) + F_1^*(\boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_{\mathrm{link}}}),
\end{aligned} \tag{6.44}$$

where $z^{\mathrm{q}}_{p_{\mathrm{link}}} := \sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}} x^{\mathrm{q}}_s\cdot a^{\mathrm{f}}_{s,p_{\mathrm{link}}}$, and the variables $x^{\mathrm{sub}}_{p_{\mathrm{link}}},\ p_{\mathrm{link}}\in\mathcal{P}_{\mathrm{link}}$, only appear in the constraints

$$\forall\ p_{\mathrm{link}}\in\mathcal{P}_{\mathrm{link}}: \qquad 0 \le \sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}} x^{\mathrm{q}}_s\cdot a^{\mathrm{f}}_{s,p_{\mathrm{link}}} - x^{\mathrm{sub}}_{p_{\mathrm{link}}}. \tag{6.45}$$

Let $\lambda^{\mathrm{main}}_{p_{\mathrm{link}}} \ge 0, p_{\mathrm{link}}\in\mathcal{P}_{\mathrm{link}}$, denote the Lagrange multipliers corresponding to these constraints.
First, we consider linking products with external demand, i.e., $p_{\mathrm{link}}\in\mathcal{P}_{\mathrm{link}}\bigcap\mathcal{P}_{out}$. In a common optimal solution of problem (6.14) with $x^{\mathrm{sales}\,*}_{p_{\mathrm{link}}} = z^{\mathrm{q}\,*}_{p_{\mathrm{link}}} - x^{\mathrm{sub}\,*}_{p_{\mathrm{link}}} > 0$, the constraints (6.45) are inactive, which means that $\lambda^{\mathrm{main}}_{p_{\mathrm{link}}} = 0$.
Partial derivatives of F_2 with respect to the variables $x^{\mathrm{sub}}_{p_{\mathrm{link}}},\ p_{\mathrm{link}}\in\mathcal{P}_{\mathrm{link}}\bigcap\mathcal{P}_{out}$, are given by

$$\begin{aligned}
\frac{\partial F_2}{\partial x^{\mathrm{sub}}_{p_{\mathrm{link}}}}(\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}},\boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_{\mathrm{link}}}) = &- P_{p_{\mathrm{link}}}(z^{\mathrm{q}}_{p_{\mathrm{link}}} - x^{\mathrm{sub}}_{p_{\mathrm{link}}}) + \frac{\partial P_{p_{\mathrm{link}}}}{\partial x^{\mathrm{sub}}_{p_{\mathrm{link}}}}(z^{\mathrm{q}}_{p_{\mathrm{link}}} - x^{\mathrm{sub}}_{p_{\mathrm{link}}})\cdot(z^{\mathrm{q}}_{p_{\mathrm{link}}} - x^{\mathrm{sub}}_{p_{\mathrm{link}}}) \\
&+ \frac{\partial F_1^*}{\partial x^{\mathrm{sub}}_{p_{\mathrm{link}}}}(\boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_{\mathrm{link}}}).
\end{aligned} \tag{6.46}$$

By the Karush-Kuhn-Tucker (KKT) conditions of theorem 3.1.10, the expression

$$\frac{\partial F_2}{\partial x^{\mathrm{sub}}_{p_{\mathrm{link}}}}(\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}},\boldsymbol{x}^{\mathrm{sub}}_{\mathcal{P}_{\mathrm{link}}}) + \underbrace{\lambda^{\mathrm{main}}_{p_{\mathrm{link}}}}_{=0} \tag{6.47}$$

vanishes in an optimal solution $(\boldsymbol{x}^{\mathrm{q}\,*}_{\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}},\boldsymbol{x}^{\mathrm{sub}\,*}_{\mathcal{P}_{\mathrm{link}}})\ \forall\ p_{\mathrm{link}}\in\mathcal{P}_{\mathrm{link}}$, i.e., we have, in case of linking products with external demand,

$$P_{p_{\mathrm{link}}}(z^{\mathrm{q}\,*}_{p_{\mathrm{link}}} - x^{\mathrm{sub}\,*}_{p_{\mathrm{link}}}) - \underbrace{\frac{\partial P_{p_{\mathrm{link}}}}{\partial x^{\mathrm{sub}}_{p_{\mathrm{link}}}}(z^{\mathrm{q}\,*}_{p_{\mathrm{link}}} - x^{\mathrm{sub}\,*}_{p_{\mathrm{link}}})}_{=-P'_{p_{\mathrm{link}}}(z^{\mathrm{q}\,*}_{p_{\mathrm{link}}} - x^{\mathrm{sub}\,*}_{p_{\mathrm{link}}})>0}\cdot\underbrace{(z^{\mathrm{q}\,*}_{p_{\mathrm{link}}} - x^{\mathrm{sub}\,*}_{p_{\mathrm{link}}})}_{>0} = \frac{\partial F_1^*}{\partial x^{\mathrm{sub}}_{p_{\mathrm{link}}}}(\boldsymbol{x}^{\mathrm{sub}\,*}_{\mathcal{P}_{\mathrm{link}}}). \tag{6.48}$$

This equation can be interpreted as follows: In an optimum of problem (6.14), there are internal equilibrium prices between the subsystem and the main system for all linking products $p_{\text{link}} \in \mathcal{P}_{\text{link}}$. The main system optimizes the production of processes $\mathcal{S} \setminus \mathcal{S}_{\text{b}}$. At a production state $\boldsymbol{x}^{\text{q}}_{\mathcal{S}\setminus\mathcal{S}_{\text{b}}}$ and subsystem input/output $\boldsymbol{x}^{\text{sub}}_{\mathcal{P}_{\text{link}}}$, in the main system, the value of a product p_{link} with external demand is given by $P_{p_{\text{link}}}(z^{\text{q}}_{p_{\text{link}}} - x^{\text{sub}}_{p_{\text{link}}}) - P'(z^{\text{q}}_{p_{\text{link}}} - x^{\text{sub}}_{p_{\text{link}}}) \cdot (z^{\text{q}}_{p_{\text{link}}} - x^{\text{sub}}_{p_{\text{link}}})$. The subsystem optimizes the processes $\mathcal{S}_{\text{b}}$. Given a subsystem input/output $\boldsymbol{x}^{\text{sub}}_{\mathcal{P}_{\text{link}}}$, in the subsystem, the value of the product p_{link} is $\lambda^{\text{sub}}_{p_{\text{link}}} = \frac{\partial F_1^*}{\partial y^{\text{q}}_{p_{\text{link}},i}}(\boldsymbol{x}^{\text{sub}}_{\mathcal{P}_{\text{link}}})$. If the value of the linking product in the subsystem and in the main system differed, the overall profit could be increased by changing the overall production. This means, such a situation would not be optimal.
From equation (6.48), it follows that at the optimal solution $(\boldsymbol{x}^{\text{q}\,*}_{\mathcal{S}\setminus\mathcal{S}_{\text{b}}}, \boldsymbol{x}^{\text{sub}\,*}_{\mathcal{P}_{\text{link}}})$, we have

$$\frac{\partial F_1^*}{\partial x^{\text{sub}}_{p_{\text{link}}}}(\boldsymbol{x}^{\text{sub}\,*}_{\mathcal{P}_{\text{link}}}) < P_{p_{\text{link}}}(z^{\text{q}\,*}_{p_{\text{link}}} - x^{\text{sub}\,*}_{p_{\text{link}}}) \quad \forall\, p_{\text{link}} \in \mathcal{P}_{\text{link}} \bigcap \mathcal{P}_{out}, \tag{6.49}$$

where $z^{\text{q}\,*}_{p_{\text{link}}} = \sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\text{b}}} x^{\text{q}\,*}_s \cdot a^{\text{f}}_{s,p_{\text{link}}}$.

Next, let us consider linking products without external demand, i.e., $p_{\text{link}} \in \mathcal{P}_{\text{link}} \bigcap \mathcal{P}_{mid}$. For these products, constraint (6.45) is usually active. The KKT condition corresponding to (6.47) becomes

$$\lambda^{\text{main}}_{p_{\text{link}}} = \frac{\partial F_1^*}{\partial x^{\text{sub}}_{p_{\text{link}}}}(\boldsymbol{x}^{\text{sub}\,*}_{\mathcal{P}_{\text{link}}}) (= \lambda^{\text{sub}}_{p_{\text{link}}}). \tag{6.50}$$

As above, the condition can be interpreted as the existence of an equilibrium price. At this, in the main system, the value of a linking product without external demand is given by its shadow price regarding constraint (6.45).

Conclusion:
We summarize the presented results by combining (6.43) and (6.49). Let $(\boldsymbol{y}^{\text{q}}_i, \pi_i)$ be generated by solving (6.13), which means $\pi_i = F_1^*(\boldsymbol{y}^{\text{q}}_i)$. We can neglect the data points $(\boldsymbol{y}^{\text{q}}_i, \pi_i)$ for which at least one of the corresponding Lagrange multipliers $\lambda^{\text{sub}}_{p_{\text{link}},i}$ considerably exceeds a reasonable price for the related product p_{link}. This is because such values $\boldsymbol{y}^{\text{q}}_i$ can impossibly become the value $\boldsymbol{x}^{\text{sub}\,*}_{\mathcal{P}_{\text{link}}}$ in an optimal solution $(\boldsymbol{x}^{\text{q}\,*}_{\mathcal{S}\setminus\mathcal{S}_{\text{b}}}, \boldsymbol{x}^{\text{sub}\,*}_{\mathcal{P}_{\text{link}}})$ of problem (6.14).
In principle, we can similarly set a reasonable lower bound for the Lagrange multipliers $\lambda^{\text{sub}}_{p_{\text{link}}}$. However, in case of products with external demand, in order to define such a bound, we need a reasonable upper bound on the absolute value of $P'_{p_{\text{link}}}$, which is the slope of the demand-price relation, see equation (6.3).
Note that there are jumps in shadow prices $\lambda^{\text{sub}}_{p_{\text{link}}} = \frac{\partial F_1^*}{\partial x^{\text{sub}}_{p_{\text{link}}}}(\boldsymbol{x}^{\text{sub}}_{\mathcal{P}_{\text{link}}})$ caused by the piecewise linear production possibilities that lead to an only piecewise differentiable optimal value function F_1^*, as we will show in the following subsection. The optimal solution of problem (6.44) can lie at such a non-differentiable point of the subproblem's optimal value function, at which an active set change takes place because one process reaches its

capacity bound and another one starts to produce.
Note that, instead of a predetermined grid $\boldsymbol{y}_1^{\mathrm{q}}, ..., \boldsymbol{y}_n^{\mathrm{q}}$, an adaptive grid refinement would be very advantageous. However, this topic is beyond the scope of this thesis.

b. Properties of a Subproblem's Optimal Value Function

In the following, we consider the optimal value function $F_1^*(\boldsymbol{y}^{\mathrm{q}})$ of problem (6.13), which we aim to approximate. Using the results from section 3.2, we show some properties of this optimal value function on the domain $\tilde{D}$ (cf. equation (6.32)). These properties are basic for the choice of an appropriate parametric model for the approximating input/output-profit/cost function c.

Theorem 6.5.2. *The optimal value function $F_1^*(\boldsymbol{y}^{\mathrm{q}})$ of the parametric optimization problem* (6.13) *is continuous on $\tilde{D}^0$, and upper semicontinuous and concave on $\tilde{D}$.*

Proof *Continuity:* By theorem 3.2.4, F_1^* is continuous at $\bar{\boldsymbol{y}}^{\mathrm{q}}$, if F_1 is continuous on $\mathcal{F}_1(\bar{\boldsymbol{y}}^{\mathrm{q}}) \times \{\bar{\boldsymbol{y}}^{\mathrm{q}}\}$, $\mathcal{F}_1(\bar{\boldsymbol{y}}^{\mathrm{q}})$ is compact and the feasible set $\mathcal{F}_1$ (cf. equation (6.33)) is continuous at $\bar{\boldsymbol{y}}^{\mathrm{q}}$.
Let be $\bar{\boldsymbol{y}}^{\mathrm{q}} \in \tilde{D}^0$, where $\tilde{D}$ is defined in (6.32). The objective function

$$F_1(\boldsymbol{x}^{\mathrm{q}}) = \sum_{p_{out} \in \mathcal{P}_{out} \bigcap \mathcal{P}_{\mathrm{b}}} \Big(\sum_{s \in \mathcal{S}_{\mathrm{b}}} x_s^{\mathrm{q}} \cdot a_{s,p_{out}}^{\mathrm{f}} \Big) \cdot P_{p_{out}} \Big(\sum_{s \in \mathcal{S}_{\mathrm{b}}} x_s^{\mathrm{q}} \cdot a_{s,p_{out}}^{\mathrm{f}} \Big) + \sum_{\substack{p_{ex} \in \mathcal{P}_{ex}, \\ s \in \mathcal{S}_{\mathrm{b}}}} x_s^{\mathrm{q}} \cdot a_{s,p_{ex}}^{\mathrm{f}} \cdot a_{p_{ex}}^{\pi} \quad (6.51)$$

is independent of $\bar{\boldsymbol{y}}^{\mathrm{q}}$ and the sum of continuous functions in $\boldsymbol{x}^{\mathrm{q}}$, i.e., it is continuous on $\mathcal{F}_1(\bar{\boldsymbol{y}}^{\mathrm{q}}) \times \{\bar{\boldsymbol{y}}^{\mathrm{q}}\}$. In addition, $\mathcal{F}_1(\bar{\boldsymbol{y}}^{\mathrm{q}})$ is a closed and bounded set and, therefore, compact.
To show the continuity of $\mathcal{F}_1$ at $\bar{\boldsymbol{y}}^{\mathrm{q}}$, we use theorem 3.2.5: The upper semicontinuity is given by part a) of the theorem through the compactness of $\mathcal{F}_1(\boldsymbol{y}^{\mathrm{q}}) \; \forall \; \boldsymbol{y}^{\mathrm{q}} \in \tilde{D}$. By part b), lower semicontinuity is given for $\boldsymbol{y}^{\mathrm{q}} \in \tilde{D}^0$, since they fulfill $\mathcal{F}_1^0(\boldsymbol{y}^{\mathrm{q}}) \neq \emptyset$ and $\overline{\mathcal{F}_1^0(\boldsymbol{y}^{\mathrm{q}})} = \mathcal{F}_1(\boldsymbol{y}^{\mathrm{q}})$. This proves continuity of F_1^* on $\tilde{D}^0$.
Since upper semicontinuity of $\mathcal{F}_1$ implies, by theorem 3.2.4, lower semicontinuity of $-F_1^*$ at $\boldsymbol{y}^{\mathrm{q}} \in \tilde{D}$, we have upper semicontinuity of F_1^* at every $\boldsymbol{y}^{\mathrm{q}} \in \tilde{D}$.[12]
Concavity: Translating theorem 3.2.7 from a minimization problem to a maximization problem while taking into account that F_1 is independent of the parameter $\boldsymbol{y}^{\mathrm{q}}$, we get that F_1^* is concave on $\tilde{D}$ if F_1 is concave, $\tilde{D}$ is convex, and $\mathcal{F}_1$ is essentially convex on $\tilde{D}$:

[12] Note that we consider a maximization problem in this chapter, which is in contrast to the minimization problem of section 3.2.

Let $\boldsymbol{y}_1^{\mathrm{q}}, \boldsymbol{y}_2^{\mathrm{q}} \in \tilde{D}, \boldsymbol{y}_1^{\mathrm{q}} \neq \boldsymbol{y}_2^{\mathrm{q}}$ and $c \in (0,1)$, then under the three foresaid conditions

$$\begin{aligned} F_1^*(c\boldsymbol{y}_1 + (1-c)\boldsymbol{y}_2) &= \max_{\boldsymbol{x} \in \mathcal{F}_1(c\boldsymbol{y}_1 + (1-c)\boldsymbol{y}_2)} F_1(\boldsymbol{x}, c\boldsymbol{y}_1 + (1-c)\boldsymbol{y}_2) \\ &= \max_{\boldsymbol{x} \in \mathcal{F}_1(c\boldsymbol{y}_1 + (1-c)\boldsymbol{y}_2)} F_1(\boldsymbol{x}) \\ &\geq \max_{\boldsymbol{x_1} \in \mathcal{F}_1(\boldsymbol{y}_1), \boldsymbol{x_2} \in \mathcal{F}_1(\boldsymbol{y}_2)} F_1(c\boldsymbol{x}_1 + (1-c)\boldsymbol{x}_2) \\ &\geq \max_{\boldsymbol{x_1} \in \mathcal{F}_1(\boldsymbol{y}_1), \boldsymbol{x_2} \in \mathcal{F}_1(\boldsymbol{y}_2)} \big(cF_1(\boldsymbol{x}_1) + (1-c)F_1(\boldsymbol{x}_2)\big) \\ &= c \max_{\boldsymbol{x_1} \in \mathcal{F}_1(\boldsymbol{y}_1)} F_1(\boldsymbol{x}_1) + (1-c) \max_{\boldsymbol{x_2} \in \mathcal{F}_1(\boldsymbol{y}_2)} F_1(\boldsymbol{x}_2) \\ &= cF_1^*(\boldsymbol{y}_1) + (1-c)F_1^*(\boldsymbol{y}_2). \end{aligned} \tag{6.52}$$

Since the concavity of F_1 has been shown in remark 6.1.2, the first condition is fulfilled. Furthermore, by remark 6.5.1, $\tilde{D}$ is convex. By definition 3.2.6, it remains to show that

$$c\mathcal{F}_1(\boldsymbol{y}_1^{\mathrm{q}}) + (1-c)\mathcal{F}_1(\boldsymbol{y}_2^{\mathrm{q}}) \subseteq \mathcal{F}_1(c\boldsymbol{y}_1^{\mathrm{q}} + (1-c)\boldsymbol{y}_2^{\mathrm{q}}) \tag{6.53}$$

$\forall\ \boldsymbol{y}_1^{\mathrm{q}}, \boldsymbol{y}_2^{\mathrm{q}} \in \tilde{D}, \boldsymbol{y}_1^{\mathrm{q}} \neq \boldsymbol{y}_2^{\mathrm{q}}$ and $\forall\ c \in (0,1)$. This holds, as showed in remark 6.5.1, because

$$\boldsymbol{x}^1 \in \mathcal{F}_1(\boldsymbol{y}^1), \boldsymbol{x}^2 \in \mathcal{F}_1(\boldsymbol{y}^2) \Rightarrow c\boldsymbol{x}^1 + (1-c)\boldsymbol{x}^2 \in \mathcal{F}_1(c\boldsymbol{y}^1 + (1-c)\boldsymbol{y}^2). \tag{6.54}$$

Theorem 6.5.3. *The optimal value function $F_1^*(\boldsymbol{y}^{\mathrm{q}})$ is almost everywhere differentiable on $\tilde{D}^0$. We have $\forall\ \boldsymbol{y}^{\mathrm{q}} \in \tilde{D}^0$: $\partial F_1^*(\boldsymbol{y}^{\mathrm{q}}) \neq \emptyset$ and*[13]

$$\boldsymbol{\lambda}^{\mathrm{sub}} \in \partial F_1^*(\boldsymbol{y}^{\mathrm{q}}) \tag{6.55}$$

if and only if $\boldsymbol{\lambda}^{\mathrm{sub}} = (\lambda^{\mathrm{sub}}_{p_{\mathrm{link}\_1}}, ..., \lambda^{\mathrm{sub}}_{p_{\mathrm{link}\_|\mathcal{P}_{\mathrm{link}}|}})^{\mathrm{T}}$ is part of an optimal solution of problem (6.13)'s dual problem as multiplier corresponding to the constraints

$$\forall\ p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}:\ 0 \leq y^{\mathrm{q}}_{p_{\mathrm{link}}} + \sum_{s \in \mathcal{S}_{\mathrm{b}}} x_s^{\mathrm{q}} \cdot a^{\mathrm{f}}_{s,p_{\mathrm{link}}}. \tag{6.56}$$

Proof Since $-F_1^*(\boldsymbol{y}^{\mathrm{q}})$ is finite and convex on $\tilde{D}$, the directional derivative[14] of $F_1^*(\boldsymbol{y}^{\mathrm{q}})$ exists in every direction $d \in \mathbb{R}^{|\mathcal{P}_{\mathrm{link}}|}$ at every point $\boldsymbol{y}^{\mathrm{q}} \in \tilde{D}$ and the subdifferential is nonempty, $\partial F_1^*(\boldsymbol{y}^{\mathrm{q}}) \neq \emptyset$, at every $\boldsymbol{y}^{\mathrm{q}} \in \tilde{D}^0$, cf. [Flo95, Roc70]. By Rademacher's theorem, since F_1^* is Lipschitz continuous on $\tilde{D}^0$, it is almost everywhere differentiable on $\tilde{D}^0$, i.e., $\partial F_1^*(\boldsymbol{y}^{\mathrm{q}}) = \{\nabla F_1^*(\boldsymbol{y}^{\mathrm{q}})\}$ almost everywhere.
By corollary 3.2.12, we have under the assumptions of theorem 3.2.10 in a neighborhood of $\boldsymbol{y}^{\mathrm{q}} \in \tilde{D}$ for each $p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}$

$$\frac{\partial F_1^*}{\partial y^{\mathrm{q}}_{p_{\mathrm{link}}}}(\boldsymbol{y}^{\mathrm{q}}) = \lambda^{\mathrm{sub}}_{p_{\mathrm{link}}}. \tag{6.57}$$

[13] The subdifferential of a function f at a point $\boldsymbol{x}$, denoted by $\partial f(\boldsymbol{x})$, is introduced in definition 3.4.1.
[14] See definition 3.4.1.

Since strong duality holds for problem (6.13), cf. remark 3.3.5, we have the following generalized result if LICQ does not hold and the Lagrangian multipliers are not unique:

$$\boldsymbol{\lambda}^{\text{sub}} \in \partial F_1^*(\boldsymbol{y}^{\text{q}}) \tag{6.58}$$

if and only if $\boldsymbol{\lambda}^{\text{sub}}$ is part of an optimal solution of problem (6.13)'s dual problem as multiplier corresponding to the constraints (6.56), see theorem 3.4.6 and [Flo95].

Note that there can be subdomains on which the optimal value function $F_1^*(\boldsymbol{y}^{\text{q}})$ is linear. This piecewise linearity is due to the structure of a submodel and appears if only production with regard to external products varies over the subdomain, i.e., if the production with regard to products with external demand is constant and sales quantities of these products do not vary. A small example of a piecewise linear optimal value function is given in the third example of section 6.3. Since the considered subproblem does not contain any product with external demand, it is a linear parametric problem. Its optimal value function consists of two linear parts with slightly different slopes.

c. Selecting a Parametric Model for the Submodel's Optimal Value Function

Because of F_1^*'s properties discussed in paragraph b., we propose a multivariate quadratic polynomial $c : \mathbb{R}^{|\mathcal{P}_{\text{link}}|} \to \mathbb{R}$ to model the optimal value function F_1^*:

$$c(\boldsymbol{y}) = \theta_0 + \sum_{i=1}^{|\mathcal{P}_{\text{link}}|} \theta_{1_i} y_i + \sum_{1 \le i \le j \le |\mathcal{P}_{\text{link}}|} \theta_{2_{ij}} y_i y_j, \tag{6.59}$$

with parameters $\theta_0, \theta_{1_i}, \theta_{2_{ij}} \in \mathbb{R}$. The parameters are estimated using the generated data $(\boldsymbol{y}_i^{\text{q}}, F_1^*(\boldsymbol{y}_i^{\text{q}}))_{i=1,\dots,n}$. The most common parameter estimation method to fit the data is the least squares method, see section 3.7.
Alternatively, there are, of course, many other reasonable parametric models that could be used. For example, in case $|\mathcal{P}_{\text{link}}| = 1$, one can choose the hyperbolic tangent function in view of modeling the profit bound, which is due to limited process capacities, cf. the numerical examples of section 6.3. Note that, in general, it is difficult to identify and model the non-differentiable points of the optimal value function so that, mostly, a differentiable function with few parameters is preferable. Furthermore, a non-differentiable submodel approximation would necessitate methods for nonsmooth convex optimization to solve the overall problem, which are usually much slower than methods for differentiable problems.
As mentioned at the end of section 5.3, one could aim to globally underestimate the subsystem profit (or, if F_1^* is negative, to globally overestimate the costs occurring in the subsystem,) to be able to compute a lower bound on the optimal value of the overall system. For this, $\boldsymbol{c}$ must be a lower bound on the optimal value function F_1^*, at least on the relevant domain $\hat{D}$ defined in (6.41). Therefore, the corresponding least squares parameter estimation should contain additional constraints forcing $c(\boldsymbol{y}_i^{\text{q}}) \le F_1^*(\boldsymbol{y}_i^{\text{q}}) \; \forall \, i = 1, \dots, n$.

d. Alternative Formulation of the Subsystem Optimization

In this paragraph, we deduce a way to generate data $(\boldsymbol{y}_i^{\mathrm{q}}, F_1^*(\boldsymbol{y}_i^{\mathrm{q}}))_{i=1,\dots,n}$ that differs from solving for varying parameters $\boldsymbol{y}_i^{\mathrm{q}} \in \mathbb{R}^{|\mathcal{P}_{\mathrm{link}}|}$ the subsystem optimization problem (6.13) given by

$$\max_{\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}_{\mathrm{b}}}} F_1(\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}_{\mathrm{b}}}) = \max_{\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}_{\mathrm{b}}}} \sum_{\substack{p_{out}\in \\ \mathcal{P}_{out}\bigcap \mathcal{P}_{\mathrm{b}}}} \Big(\sum_{s\in\mathcal{S}_{\mathrm{b}}} x_s^{\mathrm{q}} \cdot a^{\mathrm{f}}_{s,p_{out}} \Big) \cdot P_{p_{out}}\Big(\sum_{s\in\mathcal{S}_{\mathrm{b}}} x_s^{\mathrm{q}} \cdot a^{\mathrm{f}}_{s,p_{out}} \Big) + \sum_{\substack{p_{ex}\in\mathcal{P}_{ex}, \\ s\in\mathcal{S}_{\mathrm{b}}}} x_s^{\mathrm{q}} \cdot a^{\mathrm{f}}_{s,p_{ex}} \cdot a^{\pi}_{p_{ex}}$$

such that

$$\begin{aligned} &\forall\ s \in \mathcal{S}_{\mathrm{b}}: && 0 \le x_s^{\mathrm{q}} \le a_s^{\mathrm{cap}} \\ &\forall\ p \in (\mathcal{P}_{out} \dot{\bigcup} \mathcal{P}_{mid}) \bigcap \mathcal{P}_{\mathrm{b}}: && 0 \le \sum_{s\in\mathcal{S}_{\mathrm{b}}} x_s^{\mathrm{q}} \cdot a^{\mathrm{f}}_{s,p} \\ &\forall\ p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}: && 0 \le y^{\mathrm{q}}_{p_{\mathrm{link}},i} + \sum_{s\in\mathcal{S}_{\mathrm{b}}} x_s^{\mathrm{q}} \cdot a^{\mathrm{f}}_{s,p_{\mathrm{link}}}. \end{aligned}$$

Let us consider the following parametric problem. Instead of presetting submodel input/output quantities $\boldsymbol{y}^{\mathrm{q}} \in \mathbb{R}^{|\mathcal{P}_{\mathrm{link}}|}$ of the linking products, we specify prices of linking products by the parameter $\boldsymbol{\theta} \in (\mathbb{R}_0^+)^{|\mathcal{P}_{\mathrm{link}}|}$:

$$\begin{aligned} \max_{\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}_{\mathrm{b}}}} \tilde{F}_1(\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}_{\mathrm{b}}}, \boldsymbol{\theta}) = \max_{\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}_{\mathrm{b}}}} & \sum_{\substack{p_{out}\in \\ \mathcal{P}_{out}\bigcap \mathcal{P}_{\mathrm{b}}}} \Big(\sum_{s\in\mathcal{S}_{\mathrm{b}}} x_s^{\mathrm{q}} \cdot a^{\mathrm{f}}_{s,p_{out}} \Big) \cdot P_{p_{out}}\Big(\sum_{s\in\mathcal{S}_{\mathrm{b}}} x_s^{\mathrm{q}} \cdot a^{\mathrm{f}}_{s,p_{out}} \Big) \\ & + \sum_{\substack{p_{ex}\in\mathcal{P}_{ex}, \\ s\in\mathcal{S}_{\mathrm{b}}}} x_s^{\mathrm{q}} \cdot a^{\mathrm{f}}_{s,p_{ex}} \cdot a^{\pi}_{p_{ex}} + \sum_{\substack{p_{\mathrm{link}}\in\mathcal{P}_{\mathrm{link}}, \\ s\in\mathcal{S}_{\mathrm{b}}}} x_s^{\mathrm{q}} \cdot a^{\mathrm{f}}_{s,p_{\mathrm{link}}} \cdot \theta_{p_{\mathrm{link}}} \end{aligned} \tag{6.60a}$$

such that

$$\forall\ s \in \mathcal{S}_{\mathrm{b}}: \quad 0 \le x_s^{\mathrm{q}} \le a_s^{\mathrm{cap}} \tag{6.60b}$$

$$\forall\ p \in (\mathcal{P}_{out} \dot{\bigcup} \mathcal{P}_{mid}) \bigcap \mathcal{P}_{\mathrm{b}}: \quad 0 \le \sum_{s\in\mathcal{S}_{\mathrm{b}}} x_s^{\mathrm{q}} \cdot a^{\mathrm{f}}_{s,p}. \tag{6.60c}$$

A solution $\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}_{\mathrm{b}}}(\boldsymbol{\theta})$ of this problem for a fixed parameter $\boldsymbol{\theta}$ constitutes some solution $\boldsymbol{x}^{\mathrm{q}}_{\mathcal{S}_{\mathrm{b}}}(\boldsymbol{y}^{\mathrm{q}})$ of problem (6.13) for an appropriately chosen parameter $\boldsymbol{y}^{\mathrm{q}}$:[15]

[15] Note the resemblance of the objective function in (6.60) to the relaxation of the constraints (6.13d) in problem (6.13), which yields an objective

$$\sum_{\substack{p_{out}\in \\ \mathcal{P}_{out}\bigcap \mathcal{P}_{\mathrm{b}}}} \Big(\sum_{s\in\mathcal{S}_{\mathrm{b}}} x_s^{\mathrm{q}} \cdot a^{\mathrm{f}}_{s,p_{out}} \Big) \cdot P_{p_{out}}\Big(\sum_{s\in\mathcal{S}_{\mathrm{b}}} x_s^{\mathrm{q}} \cdot a^{\mathrm{f}}_{s,p_{out}} \Big) + \sum_{\substack{p_{ex}\in\mathcal{P}_{ex}, \\ s\in\mathcal{S}_{\mathrm{b}}}} x_s^{\mathrm{q}} \cdot a^{\mathrm{f}}_{s,p_{ex}} \cdot a^{\pi}_{p_{ex}} + \sum_{p_{\mathrm{link}}\in\mathcal{P}_{\mathrm{link}}} \lambda^{\mathrm{sub}}_{p_{\mathrm{link}},i} \Big(y^{\mathrm{q}}_{p_{\mathrm{link}},i} + \sum_{s\in\mathcal{S}_{\mathrm{b}}} x_s^{\mathrm{q}} \cdot a^{\mathrm{f}}_{s,p_{\mathrm{link}}} \Big).$$

Theorem 6.5.4. *Let $\boldsymbol{x}_{\mathcal{S}_\mathrm{b}}^{\mathrm{q}*}$ be a global solution of problem* (6.60) *for a fixed parameter $\boldsymbol{\theta} \in (\mathbb{R}_0^+)^{|\mathcal{P}_\mathrm{link}|}$ and let $\boldsymbol{y}^\mathrm{q} = (y^\mathrm{q}_{p_{\mathrm{link}\_1}}, ..., y^\mathrm{q}_{p_{\mathrm{link}\_|\mathcal{P}_\mathrm{link}|}})^\mathrm{T}$ be given by*

$$y^\mathrm{q}_{p_\mathrm{link}} := \sum_{s \in \mathcal{S}_\mathrm{b}} x_s^{\mathrm{q}*} \cdot a^\mathrm{f}_{s,p_\mathrm{link}}. \tag{6.61}$$

Then $\boldsymbol{x}_{\mathcal{S}_\mathrm{b}}^{\mathrm{q}*}$ is also a global solution of problem (6.13) *for the parameter $\boldsymbol{y}^\mathrm{q}$. Furthermore, we have*

$$F_1^*(\boldsymbol{y}^\mathrm{q}) = \tilde{F}_1^*(\boldsymbol{\theta}) \; - \sum_{\substack{p_\mathrm{link} \in \mathcal{P}_\mathrm{link}, \\ s \in \mathcal{S}_\mathrm{b}}} x_s^\mathrm{q} \cdot a^\mathrm{f}_{s,p_\mathrm{link}} \cdot \theta_{p_\mathrm{link}}. \tag{6.62}$$

Proof By remark 3.3.5, theorem 3.1.13 holds for convex problems with linear constraints in general. Concerning problems (6.13) and (6.60), this means that for each maximizer there exist Lagrange multipliers constituting, together with the maximizer, a KKT-point. Conversely, each KKT-point includes a maximizer.
Let us consider the global maximizer $\boldsymbol{x}_{\mathcal{S}_\mathrm{b}}^{\mathrm{q}*}$ of problem (6.60). There exist $\boldsymbol{\lambda}^\mathrm{zero}, \boldsymbol{\lambda}^\mathrm{cap} \in (\mathbb{R}_0^+)^{|\mathcal{S}_\mathrm{b}|}$, and $\boldsymbol{\lambda}^{\mathrm{net\_prod}} \in (\mathbb{R}_0^+)^{|(\mathcal{P}_{out} \,\dot\cup\, \mathcal{P}_{mid}) \bigcap \mathcal{P}_\mathrm{b}|}$ such that

$$\nabla_{\boldsymbol{x}} \tilde{\mathcal{L}}(\boldsymbol{x}_{\mathcal{S}_\mathrm{b}}^{\mathrm{q}*}, \boldsymbol{\lambda}^\mathrm{zero}, \boldsymbol{\lambda}^\mathrm{cap}, \boldsymbol{\lambda}^{\mathrm{net\_prod}}, \boldsymbol{\theta}) = \boldsymbol{0}, \tag{6.63}$$

where the Lagrangian function $\tilde{\mathcal{L}}$ is given by

$$\begin{aligned} \tilde{\mathcal{L}}(\boldsymbol{x}_{\mathcal{S}_\mathrm{b}}^\mathrm{q}, \boldsymbol{\lambda}^\mathrm{zero}, \boldsymbol{\lambda}^\mathrm{cap}, \boldsymbol{\lambda}^{\mathrm{net\_prod}}, \boldsymbol{\theta}) &= \tilde{F}_1(\boldsymbol{x}_{\mathcal{S}_\mathrm{b}}^\mathrm{q}, \boldsymbol{\theta}) + \sum_{s \in \mathcal{S}_\mathrm{b}} \lambda_s^\mathrm{zero} \cdot x_s^\mathrm{q} \\ &+ \sum_{s \in \mathcal{S}_\mathrm{b}} \lambda_s^\mathrm{cap}(a_s^\mathrm{cap} - x_s^\mathrm{q}) + \sum_{p \in (\mathcal{P}_{out} \,\dot\cup\, \mathcal{P}_{mid}) \bigcap \mathcal{P}_\mathrm{b}} \big(\lambda_p^{\mathrm{net\_prod}} \sum_{s \in \mathcal{S}_\mathrm{b}} x_s^\mathrm{q} \cdot a^\mathrm{f}_{s,p}\big), \end{aligned} \tag{6.64}$$

and such that complementarity holds:

$$\sum_{s \in \mathcal{S}_\mathrm{b}} \lambda_s^\mathrm{zero} \cdot x_s^{\mathrm{q}*} + \sum_{s \in \mathcal{S}_\mathrm{b}} \lambda_s^\mathrm{cap}(a_s^\mathrm{cap} - x_s^{\mathrm{q}*}) + \sum_{p \in (\mathcal{P}_{out} \,\dot\cup\, \mathcal{P}_{mid}) \bigcap \mathcal{P}_\mathrm{b}} \big(\lambda_p^{\mathrm{net\_prod}} \sum_{s \in \mathcal{S}_\mathrm{b}} x_s^{\mathrm{q}*} \cdot a^\mathrm{f}_{s,p}\big) = 0. \tag{6.65}$$

Considering these equations, we see that $(\boldsymbol{x}_{\mathcal{S}_\mathrm{b}}^{\mathrm{q}*}, \boldsymbol{\lambda}^\mathrm{zero}, \boldsymbol{\lambda}^\mathrm{cap}, \boldsymbol{\lambda}^{\mathrm{net\_prod}}, \boldsymbol{\theta})$ is a KKT-point of the parametric problem (6.13) with parameter $\boldsymbol{y}^\mathrm{q}$ defined by (6.61): complementarity obviously holds by (6.65) and the definition of $\boldsymbol{y}^\mathrm{q}$, and for the Lagrangian function $\mathcal{L}$ of problem (6.13), we have

$$\nabla_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}_{\mathcal{S}_\mathrm{b}}^{\mathrm{q}*}, \boldsymbol{\lambda}^\mathrm{zero}, \boldsymbol{\lambda}^\mathrm{cap}, \boldsymbol{\lambda}^{\mathrm{net\_prod}}, \boldsymbol{\theta}) = \nabla_{\boldsymbol{x}} \tilde{\mathcal{L}}(\boldsymbol{x}_{\mathcal{S}_\mathrm{b}}^{\mathrm{q}*}, \boldsymbol{\lambda}^\mathrm{zero}, \boldsymbol{\lambda}^\mathrm{cap}, \boldsymbol{\lambda}^{\mathrm{net\_prod}}, \boldsymbol{\theta}) = 0. \tag{6.66}$$

This means, by theorem 3.1.13, $\boldsymbol{x}_{\mathcal{S}_\mathrm{b}}^{\mathrm{q}*}$ is a global solution of problem (6.13) for the parameter $\boldsymbol{y}^\mathrm{q}$. Furthermore, we have

$$F_1^*(\boldsymbol{y}^\mathrm{q}) = F_1(\boldsymbol{x}_{\mathcal{S}_\mathrm{b}}^{\mathrm{q}*}) = \tilde{F}_1(\boldsymbol{x}_{\mathcal{S}_\mathrm{b}}^{\mathrm{q}*}, \boldsymbol{\theta}) - \sum_{\substack{p_\mathrm{link} \in \mathcal{P}_\mathrm{link}, \\ s \in \mathcal{S}_\mathrm{b}}} x_s^\mathrm{q} \cdot a^\mathrm{f}_{s,p_\mathrm{link}} \cdot \theta_{p_\mathrm{link}} = \tilde{F}_1^*(\boldsymbol{\theta}) - \sum_{\substack{p_\mathrm{link} \in \mathcal{P}_\mathrm{link}, \\ s \in \mathcal{S}_\mathrm{b}}} x_s^\mathrm{q} \cdot a^\mathrm{f}_{s,p_\mathrm{link}} \cdot \theta_{p_\mathrm{link}}. \tag{6.67}$$

Theorem 6.5.4 shows that, aiming to approximate the optimal value function F_1^*, we can generate data points $(\boldsymbol{y}_i^{\mathrm{q}}, F_1^*(\boldsymbol{y}_i^{\mathrm{q}}))_{i=1,\dots,n}$ by solving problem (6.60) for varying $\boldsymbol{\theta}_i \in (\mathbb{R}_0^+)^{|\mathcal{P}_{\mathrm{link}}|}$ instead of solving problem (6.13) for varying $\boldsymbol{y}_i^{\mathrm{q}} \in \mathbb{R}^{|\mathcal{P}_{\mathrm{link}}|}$. This means, instead of presetting submodel input/output quantities of the linking products in problem (6.13), we specify prices of linking products in problem (6.60). The alternative approach is of interest if the optimal value function is nonlinear or nonsmooth, since the price parameter $\boldsymbol{\theta}$ in problem (6.60) corresponds to the shadow prices of linking products in problem (6.13) and hence to the (sub)gradient of the optimal value function F_1^*.
Let us regard the restriction of the domain on which the optimal value function should be approximated by taking into account shadow prices. We discussed this topic in paragraph a. The alternative approach to generate data $(\boldsymbol{y}_i^{\mathrm{q}}, F_1^*(\boldsymbol{y}_i^{\mathrm{q}}))_{i=1,\dots,n}$ is useful in this context since it allows to predetermine reasonable prices of linking products instead of computing first a solution of problem (6.13) and deciding afterwards, by means of shadow prices, if the computed data point is suitable.
In general, it is useful to bear both ways to compute submodel solutions in mind, since they allow varying analyses. Note that a linear piece of problem (6.13)'s optimal value function F_1^* corresponds to a certain parameter $\boldsymbol{\theta}$ for which problem (6.60) has infinite many solutions. Conversely, a non-differentiable point of problem (6.13)'s optimal value function F_1^* corresponds to a set of infinite many parameters $\boldsymbol{\theta}$ leading to the same optimal solution of problem (6.60), however inducing different optimal values.

6.6 Various Aspects of a Sensitivity Analysis

In this section, we analyze the influence of parameter values on the solutions of our network optimization model by using the theoretical results of section 3.2. Regarding real applications that need reliable price simulations, this topic is of special interest due to the high procurement costs of detailed market data. The knowledge about sensitivities allows to decide which data is worth to expend much effort or money. Furthermore, sensitivities are essential for the risk management of companies that buy products of the considered market. Considering long-term decisions, sensitivity analysis becomes relevant also regarding investments in additional production capacities.
With regard to the market modeling, sensitivity analysis possibly justifies the fixing of some model variables. By this, edges of the overall network may disappear, which leads to a simpler network structure and, perhaps, allows a new network partition, i.e., a new decomposition of the overall problem into subproblems.

Particularly, we aim to

- conduct submodel analysis without any restrictive assumptions on the parameters of the remaining part of the model.
- systematically identify model parameters, as prices of external products and process capacities, that strongly influence our price simulations, i.e., the price solutions

of our optimization model.

- analyze to which extent submodel solutions are influenced by submodel parameters and in what way they are influenced by the input/output quantities of linking products.
- differentiate between several submodels, i.e., to find out which ones have strong influence on the remaining part of the network (problem (6.14)) and which not, and in which cases, the optimal decisions in the main part of the model highly depend on the amount of the linking products that is injected into (or withdrawn from) the submodel.
- determine the influence of subsystem parameters on the aggregation of the subsystem, i.e., on the input/output-profit/cost function, since this function represents the subsystem in the problems (6.14) and (6.15), respectively, which approximate the overall problem (6.5).
- find out if some submodel input/output-profit/cost functions c could be simplified, for instance, be modeled additively separable, i.e., $c(\boldsymbol{y}^{\mathrm{q}}) = \sum_{p \in \mathcal{P}_{\mathrm{link}}} c_p(y_p^{\mathrm{q}})$.

Given a parametric optimization problem as in (3.20)

$$\min_{\boldsymbol{x} \in \mathcal{F}(\boldsymbol{\theta})} f(\boldsymbol{x}, \boldsymbol{\theta}) \tag{6.68a}$$

with

$$\mathcal{F}(\boldsymbol{\theta}) = \{\ x \in \mathbb{R}^n \mid g_i(\boldsymbol{x}, \boldsymbol{\theta}) \leq 0\ (i \in I),\ h_j(\boldsymbol{x}, \boldsymbol{\theta}) = 0\ (j \in J)\ \}, \tag{6.68b}$$

where $f : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}$, $g_i : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}$ $(i \in I)$, and $h_j : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}$ $(j \in J)$, we distinguish between optimal value sensitivities

$$\nabla_{\boldsymbol{\theta}} f^*(\boldsymbol{\theta}) \tag{6.69}$$

and solution sensitivities

$$\nabla_{\boldsymbol{\theta}} \boldsymbol{x}(\boldsymbol{\theta}). \tag{6.70}$$

Since the aim of our modeling approach is mainly in correct price simulations, we are particularly interested in solution sensitivities, specifically in price/sales sensitivities of products with external demand. However, regarding our model reduction approach, optimal value sensitivities become relevant with regard to the input/output-profit/cost function $c(\boldsymbol{y}^{\mathrm{q}})$ of a submodel, which approximates the optimal value function $F_1^*(\boldsymbol{y}^{\mathrm{q}})$ of problem (6.13).
In the following, we first outline sensitivity analysis within the original optimization problem (6.5). Afterwards, we discuss possibilities of sensitivity analysis within the subproblem (6.13) as well as their advantages and results regarding the objectives mentioned above.

a. Sensitivity Analysis within the Original Optimization Problem (6.5):

In this paragraph, we calculate optimal value and solution sensitivities with respect to external prices $a^{\pi}_{p_{ex}}, p_{ex} \in \mathcal{P}_{ex}$, and with respect to process capacities $a^{\mathrm{cap}}_s, s \in \mathcal{S}$. To compute solution sensitivities, we need first and second partial derivatives of the objective function with respect to process variables $x^{\mathrm{q}}_s, s \in \mathcal{S}$. For processes $i, j \in \mathcal{S}$, these are given by

$$\frac{\partial F}{\partial x^{\mathrm{q}}_i}(\boldsymbol{x}^{\mathrm{q}}) = \sum_{p_{out}\in\mathcal{P}_{out}} \Big(a^{\mathrm{f}}_{i,p_{out}} P_{p_{out}}\big(\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s a^{\mathrm{f}}_{s,p_{out}}\big) + \big(\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s a^{\mathrm{f}}_{s,p_{out}}\big) \cdot \underbrace{\frac{\partial P_{p_{out}}}{\partial x^{\mathrm{q}}_i}\big(\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s a^{\mathrm{f}}_{s,p_{out}}\big)}_{a^{\mathrm{f}}_{i,p_{out}} \cdot P'_{p_{out}}(\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s a^{\mathrm{f}}_{s,p_{out}})} \Big) + \sum_{p_{ex}\in\mathcal{P}_{ex}} a^{\mathrm{f}}_{i,p_{ex}} a^{\pi}_{p_{ex}} \tag{6.71}$$

and

$$\begin{aligned}\frac{\partial^2 F}{\partial x^{\mathrm{q}}_i \partial x^{\mathrm{q}}_j}(\boldsymbol{x}^{\mathrm{q}}) &= \sum_{p_{out}\in\mathcal{P}_{out}} \Big(a^{\mathrm{f}}_{i,p_{out}} \frac{\partial P_{p_{out}}}{\partial x^{\mathrm{q}}_j}\big(\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s a^{\mathrm{f}}_{s,p_{out}}\big) + \sum_{p_{out}\in\mathcal{P}_{out}} a^{\mathrm{f}}_{j,p_{out}} \frac{\partial P_{p_{out}}}{\partial x^{\mathrm{q}}_i}\big(\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s a^{\mathrm{f}}_{s,p_{out}}\big) \\ &\qquad + \big(\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s a^{\mathrm{f}}_{s,p_{out}}\big) \cdot \frac{\partial^2 P_{p_{out}}}{\partial x^{\mathrm{q}}_i \partial x^{\mathrm{q}}_j}\big(\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s a^{\mathrm{f}}_{s,p_{out}}\big)\Big) \\ &= \sum_{p_{out}\in\mathcal{P}_{out}} \Big(2\, a^{\mathrm{f}}_{i,p_{out}} a^{\mathrm{f}}_{j,p_{out}} P'_{p_{out}}\big(\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s a^{\mathrm{f}}_{s,p_{out}}\big) \\ &\qquad + \big(\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s a^{\mathrm{f}}_{s,p_{out}}\big) a^{\mathrm{f}}_{i,p_{out}} a^{\mathrm{f}}_{j,p_{out}} P''_{p_{out}}\big(\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s a^{\mathrm{f}}_{s,p_{out}}\big)\Big),\end{aligned} \tag{6.72}$$

where $P'_{p_{out}}$ and $P''_{p_{out}}$ are given by (6.3) and (6.4), respectively. Since problem (6.5) is a maximization problem, we consider the minimization of $-F$ and the Lagrangian function

$$\begin{aligned}\mathcal{L}(\boldsymbol{x}^{\mathrm{q}}, \boldsymbol{\lambda}) = -F(\boldsymbol{x}^{\mathrm{q}}) &+ \sum_{s\in\mathcal{S}} \lambda^{\mathrm{zero}}_s(-x^{\mathrm{q}}_s) + \sum_{s\in\mathcal{S}} \lambda^{\mathrm{cap}}_s(x^{\mathrm{q}}_s - a^{\mathrm{cap}}_s) \\ &+ \sum_{p\in\mathcal{P}_{out}\bigcup\mathcal{P}_{mid}} \lambda^{\mathrm{net\_prod}}_s(-\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s a^{\mathrm{f}}_{s,p}),\end{aligned} \tag{6.73}$$

where $\boldsymbol{\lambda} = (\boldsymbol{\lambda}^{\mathrm{zero}}, \boldsymbol{\lambda}^{\mathrm{cap}}, \boldsymbol{\lambda}^{\mathrm{net\_prod}})$ is the vector of Lagrange multipliers corresponding to the production bounds (6.5b) and the positive net production (6.5c), respectively. Taking these expressions into account, the matrix $\boldsymbol{M}$ of theorem 3.2.11, equation (3.28), is for

problem (6.5) given by

$$\boldsymbol{M} = \left(\begin{array}{ccc|cc|ccc} & & & -1 & \mathbf{0} & 1 & \mathbf{0} & -a^{\mathrm{f}}_{1,1} & \dots & -a^{\mathrm{f}}_{1,n_p} \\ & -\nabla^2_{\boldsymbol{x}^{\mathrm{q}}\boldsymbol{x}^{\mathrm{q}}}F & & & \ddots & & \ddots & \vdots & \ddots & \vdots \\ & & & \mathbf{0} & -1 & \mathbf{0} & 1 & -a^{\mathrm{f}}_{n_s,1} & \dots & -a^{\mathrm{f}}_{n_s,n_p} \\ \hline -\lambda^{\mathrm{zero}}_1 & & \mathbf{0} & -x^{\mathrm{q}}_1 & \mathbf{0} & & & & & \\ & \ddots & & & \ddots & & \mathbf{0} & & \mathbf{0} & \\ \mathbf{0} & & -\lambda^{\mathrm{zero}}_{n_s} & \mathbf{0} & -x^{\mathrm{q}}_{n_s} & & & & & \\ \hline \lambda^{\mathrm{cap}}_1 & & \mathbf{0} & & & x^{\mathrm{q}}_1 - a^{\mathrm{cap}}_1 & \mathbf{0} & & & \\ & \ddots & & & \mathbf{0} & & \ddots & & \mathbf{0} & \\ \mathbf{0} & & \lambda^{\mathrm{cap}}_{n_s} & & & \mathbf{0} & x^{\mathrm{q}}_{n_s} - a^{\mathrm{cap}}_{n_s} & & & \\ \hline -\lambda^{\mathrm{net\_prod}}_1 a^{\mathrm{f}}_{1,1} & \dots & -\lambda^{\mathrm{net\_prod}}_1 a^{\mathrm{f}}_{n_s,1} & & & & & -\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s a^{\mathrm{f}}_{s,1} & & \mathbf{0} \\ \vdots & \ddots & \vdots & & \mathbf{0} & & \mathbf{0} & & \ddots & \\ -\lambda^{\mathrm{net\_prod}}_{n_p} a^{\mathrm{f}}_{1,n_p} & \dots & -\lambda^{\mathrm{net\_prod}}_{n_p} a^{\mathrm{f}}_{n_s,n_p} & & & & & \mathbf{0} & & -\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s a^{\mathrm{f}}_{s,n_p} \end{array}\right) \tag{6.74}$$

where $n_s := |\mathcal{S}|$ and $n_p := |\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}|$. $\boldsymbol{M}$ is a $(3n_s + n_p) \times (3n_s + n_p)$ matrix. Primal and dual variable values in $\boldsymbol{M}$ depend on the considered parameter.

We first treat problem (6.5) as parametric problem in the external prices $a^{\pi}_{p_{ex}}, p_{ex} \in \mathcal{P}_{ex}$. Optimal value sensitivities with respect to external prices are, by theorem 3.2.11 b) under the respective regularity conditions, given by

$$\frac{\partial F^*}{\partial a^{\pi}_{p_{ex}}}(\boldsymbol{a}^{\pi}) = -\frac{\partial \mathcal{L}}{\partial a^{\pi}_{p_{ex}}}(\boldsymbol{x}^{\mathrm{q}}(\boldsymbol{a}^{\pi}), \boldsymbol{a}^{\pi}) = \frac{\partial F}{\partial a^{\pi}_{p_{ex}}}(\boldsymbol{x}^{\mathrm{q}}(\boldsymbol{a}^{\pi}), \boldsymbol{a}^{\pi}) = \sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s(\boldsymbol{a}^{\pi}) a^{\mathrm{f}}_{s,p_{ex}}, \tag{6.75}$$

where $\boldsymbol{x}^{\mathrm{q}}(\boldsymbol{a}^{\pi})$ is the solution of the optimization problem (6.5) as function of the parameter $\boldsymbol{a}^{\pi}$. Usually, most external products are either only processed or only produced, except for some special cases of modeling, as in our example product 128, cf. the last part of section 6.3. Thus, we commonly have the sensitivity bounds

$$\Big|\frac{\partial F^*}{\partial a^{\pi}_{p_{ex}}}(\boldsymbol{a}^{\pi})\Big| \le \Big|\sum_{s\in\mathcal{S}} a^{\mathrm{cap}}_s a^{\mathrm{f}}_{s,p_{ex}}\Big|. \tag{6.76}$$

To compute solution sensitivities, we need to calculate the matrix $\boldsymbol{N}$ of theorem 3.2.11:

$$\boldsymbol{N}_{\boldsymbol{a}^{\pi}} = \left(\begin{array}{ccc} -a^{\mathrm{f}}_{1,1} & \dots & -a^{\mathrm{f}}_{1,|\mathcal{P}_{ex}|} \\ \vdots & \ddots & \vdots \\ -a^{\mathrm{f}}_{|\mathcal{S}|,1} & \dots & -a^{\mathrm{f}}_{n_s,|\mathcal{P}_{ex}|} \\ \hline \\ \mathbf{0} \\ \\ \end{array}\right). \tag{6.77}$$

$\boldsymbol{N}_{\boldsymbol{a}^\pi}$ is of size $(3n_s + n_p) \times |\mathcal{P}_{ex}|$. By (3.27), the corresponding solution sensitivities are given by

$$\begin{pmatrix} \nabla_{\boldsymbol{a}^\pi} \boldsymbol{x}^{\mathrm{q}}(\boldsymbol{a}^\pi) \\ \nabla_{\boldsymbol{a}^\pi} \boldsymbol{\lambda}^{\mathrm{zero}}(\boldsymbol{a}^\pi) \\ \nabla_{\boldsymbol{a}^\pi} \boldsymbol{\lambda}^{\mathrm{cap}}(\boldsymbol{a}^\pi) \\ \nabla_{\boldsymbol{a}^\pi} \boldsymbol{\lambda}^{\mathrm{net\_prod}}(\boldsymbol{a}^\pi) \end{pmatrix} = -\boldsymbol{M}^{-1} \boldsymbol{N}_{\boldsymbol{a}^\pi}. \tag{6.78}$$

Defining price variables $x^\pi_{p_{out}} := P_{p_{out}}(\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s a^{\mathrm{f}}_{s,p_{out}})\ \forall\ p_{out} \in \mathcal{P}_{out}$ by the price-demand relationship (6.2), we get the sensitivities of these prices with respect to external prices by the chain rule

$$\frac{\partial x^\pi_{p_{out}}}{\partial a^\pi_{p_{ex}}}(\boldsymbol{a}^\pi) = P'_{p_{out}}\Big(\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s(\boldsymbol{a}^\pi) a^{\mathrm{f}}_{s,p_{out}}\Big) \begin{pmatrix} a^{\mathrm{f}}_{1,p_{out}} \\ \vdots \\ a^{\mathrm{f}}_{|\mathcal{S}|,p_{out}} \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \frac{\partial x^{\mathrm{q}}_1}{\partial a^\pi_{p_{ex}}}(\boldsymbol{a}^\pi) \\ \vdots \\ \frac{\partial x^{\mathrm{q}}_{|\mathcal{S}|}}{\partial a^\pi_{p_{ex}}}(\boldsymbol{a}^\pi) \end{pmatrix}. \tag{6.79}$$

Similarly, we calculate sensitivities with respect to capacities $a^{\mathrm{cap}}_s, s \in \mathcal{S}$. Therefore, we consider problem (6.5) as parametric problem in the capacity vector $\boldsymbol{a}^{\mathrm{cap}}$. By corollary 3.2.12, we have

$$\frac{\partial F^*}{\partial a^{\mathrm{cap}}_s}(\boldsymbol{a}^{\mathrm{cap}}) = -\frac{\partial \mathcal{L}}{\partial a^{\mathrm{cap}}_s}(\boldsymbol{x}^{\mathrm{q}}(\boldsymbol{a}^{\mathrm{cap}}), \boldsymbol{a}^{\mathrm{cap}}) = \lambda^{\mathrm{cap}}_s(\boldsymbol{a}^{\mathrm{cap}}). \tag{6.80}$$

The matrix $\boldsymbol{N}$ of theorem 3.2.11 regarding capacities is of size $(3n_s + n_p) \times |\mathcal{S}|$ and reads

$$\boldsymbol{N}_{\boldsymbol{a}^{\mathrm{cap}}} = \begin{pmatrix} \mathbf{0} \\ \hline \mathbf{0} \\ \hline -\lambda^{\mathrm{cap}}_1(\boldsymbol{a}^{\mathrm{cap}}) & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & -\lambda^{\mathrm{cap}}_{|\mathcal{S}|}(\boldsymbol{a}^{\mathrm{cap}}) \\ \hline \mathbf{0} \end{pmatrix}. \tag{6.81}$$

The corresponding solution sensitivities are

$$\begin{pmatrix} \nabla_{\boldsymbol{a}^{\mathrm{cap}}} \boldsymbol{x}^{\mathrm{q}}(\boldsymbol{a}^{\mathrm{cap}}) \\ \nabla_{\boldsymbol{a}^{\mathrm{cap}}} \boldsymbol{\lambda}^{\mathrm{zero}}(\boldsymbol{a}^{\mathrm{cap}}) \\ \nabla_{\boldsymbol{a}^{\mathrm{cap}}} \boldsymbol{\lambda}^{\mathrm{cap}}(\boldsymbol{a}^{\mathrm{cap}}) \\ \nabla_{\boldsymbol{a}^{\mathrm{cap}}} \boldsymbol{\lambda}^{\mathrm{net\_prod}}(\boldsymbol{a}^{\mathrm{cap}}) \end{pmatrix} = -\boldsymbol{M}^{-1} \boldsymbol{N}_{\boldsymbol{a}^{\mathrm{cap}}}. \tag{6.82}$$

Using the sensitivities of the primal variables $x^{\mathrm{q}}_s, s \in \mathcal{S}$, we can also calculate sensitivities

of the prices $x^{\pi}_{p_{out}} = P_{p_{out}}(\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s a^{\mathrm{f}}_{s,p_{out}})$, $p_{out} \in \mathcal{P}_{out}$, with respect to capacities:

$$\frac{\partial x^{\pi}_{p_{out}}}{\partial a^{\mathrm{cap}}_s}(\boldsymbol{a}^{\mathrm{cap}}) = P'_{p_{out}}\Big(\sum_{s\in\mathcal{S}} x^{\mathrm{q}}_s(\boldsymbol{a}^{\mathrm{cap}}) a^{\mathrm{f}}_{s,p_{out}}\Big) \begin{pmatrix} a^{\mathrm{f}}_{1,p_{out}} \\ \vdots \\ a^{\mathrm{f}}_{|\mathcal{S}|,p_{out}} \end{pmatrix}^{\mathrm{T}} \begin{pmatrix} \frac{\partial x^{\mathrm{q}}_1}{\partial a^{\mathrm{cap}}_s}(\boldsymbol{a}^{\mathrm{cap}}) \\ \vdots \\ \frac{\partial x^{\mathrm{q}}_{|\mathcal{S}|}}{\partial a^{\mathrm{cap}}_s}(\boldsymbol{a}^{\mathrm{cap}}) \end{pmatrix}. \tag{6.83}$$

Of course, one can also compute all solution sensitivities at once by $-\boldsymbol{M}^{-1}(\boldsymbol{N}_{\boldsymbol{a}^{\pi}}, \boldsymbol{N}_{\boldsymbol{a}^{\mathrm{cap}}})$. Note that there can be discontinuities in the discussed optimal value and solution sensitivities caused by active set changes. In the case that LICQ or strict complementarity are not fulfilled, optimal value and solution derivatives can be analyzed by the method of [CCC+06, CCMGB06], which we presented at the end of section 3.2.

Remark 6.6.1. The sensitivities calculated for a parameter value $\boldsymbol{a}_0$ can be used to approximate a solution of a perturbed problem in a parameter $\boldsymbol{a}$. First order estimates are given by

$$\begin{pmatrix} \boldsymbol{x}(\boldsymbol{a}) \\ \boldsymbol{\lambda}(\boldsymbol{a}) \end{pmatrix} \approx \begin{pmatrix} \boldsymbol{x}(\boldsymbol{a}_0) \\ \boldsymbol{\lambda}(\boldsymbol{a}_0) \end{pmatrix} + \begin{pmatrix} \frac{d\boldsymbol{x}(\boldsymbol{a}_0)}{d\boldsymbol{a}} \\ \frac{d\boldsymbol{\lambda}(\boldsymbol{a}_0)}{d\boldsymbol{a}} \end{pmatrix} (\boldsymbol{a} - \boldsymbol{a}_0). \tag{6.84}$$

To compute these approximations, one can, for instance, apply the program sIPOPT, see [PLNB12, PLNB11]. It is based on the interior point solver IPOPT, cf. section 3.6. Note that a special proceeding is required if the considered perturbation causes an active set change. For this purpose, sIPOPT provides a bound correction algorithm.

Remark 6.6.2. In the context of sensitivities, we would also like to draw attention to the Lagrange multipliers $\lambda^{\mathrm{net\_prod}}_p$, $p \in \mathcal{P}_{mid}$, corresponding to the constraints that avoid negative net production of intermediate products. At an optimal solution, they display the system's marginal prices of the intermediates, cf. also remark 2.1.1.

b. Sensitivity Analysis within the Subnetwork Optimization Problem (6.13)

Submodel sensitivities can be computed similar to the sensitivities of the overall model. The related matrix $\boldsymbol{M}^{\mathrm{sub}}$ differs from $\boldsymbol{M}$ in (6.74) in the following way: $\boldsymbol{M}^{\mathrm{sub}}$ contains only those rows and columns of $\boldsymbol{M}$ that correspond to subsystem processes $\mathcal{S}_{\mathrm{b}}$ and subsystem products $(\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}) \bigcap \mathcal{P}_{\mathrm{b}}$. Furthermore, we summarize in the remaining entries only over processes $s \in \mathcal{S}_{\mathrm{b}}$. Let $\boldsymbol{M}^{\mathrm{red}}$ denote the described modification of $\boldsymbol{M}$. Additionally, $\boldsymbol{M}^{\mathrm{sub}}$ includes rows and columns corresponding to the constraints (6.13d).

Hence, it is a square matrix of order $3|\mathcal{S}_\mathrm{b}| + |(\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}) \bigcap \mathcal{P}_\mathrm{b}| + |\mathcal{P}_\mathrm{link}|$ and reads

$$\boldsymbol{M}^\mathrm{sub} = \left(\begin{array}{ccc|c|ccc} & & & & -a^\mathrm{f}_{1,1} & \dots & -a^\mathrm{f}_{1,n_p} \\ & & & & \vdots & \ddots & \vdots \\ & \boldsymbol{M}^\mathrm{red} & & & -a^\mathrm{f}_{n_s,1} & \dots & -a^\mathrm{f}_{n_s,n_p} \\ & & & & \multicolumn{3}{c}{\multirow{2}{*}{$\mathbf{0}$}} \\ & & & & & & \\ \hline -\lambda^\mathrm{sub}_1 a^\mathrm{f}_{1,1} & \dots & -\lambda^\mathrm{sub}_1 a^\mathrm{f}_{n_s,1} & & -y^\mathrm{q}_1 - \sum_{s\in\mathcal{S}_\mathrm{b}} x^\mathrm{q}_s a^\mathrm{f}_{s,1} & & \mathbf{0} \\ \vdots & \ddots & \vdots & \mathbf{0} & & \ddots & \\ -\lambda^\mathrm{sub}_{n_p} a^\mathrm{f}_{1,n_p} & \dots & -\lambda^\mathrm{sub}_{n_p} a^\mathrm{f}_{n_s,n_p} & & \mathbf{0} & & -y^\mathrm{q}_{n_p} - \sum_{s\in\mathcal{S}_\mathrm{b}} x^\mathrm{q}_s a^\mathrm{f}_{s,n_p} \end{array}\right) \tag{6.85}$$

where $n_s := |\mathcal{S}_\mathrm{b}|$, $n_p := |\mathcal{P}_\mathrm{link}|$, and $\boldsymbol{\lambda}^\mathrm{sub}$ consists of the Lagrange multipliers corresponding to the constraints (6.13d).
Let us consider sensitivities with respect to the submodel input/output parameters $y^\mathrm{q}_{p_\mathrm{link}}, p_\mathrm{link} \in \mathcal{P}_\mathrm{link}$. As stated in (6.43), we have

$$\frac{\partial F^*_1}{\partial y^\mathrm{q}_{p_\mathrm{link}}}(\boldsymbol{y}^\mathrm{q}) = \lambda^\mathrm{sub}_{p_\mathrm{link}}(\boldsymbol{y}^\mathrm{q}). \tag{6.86}$$

With the $(3|\mathcal{S}_\mathrm{b}| + |(\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}) \bigcap \mathcal{P}_\mathrm{b}| + |\mathcal{P}_\mathrm{link}|) \times |\mathcal{P}_\mathrm{link}|$ matrix

$$\boldsymbol{N}_{\boldsymbol{y}^\mathrm{q}} = \left(\begin{array}{c|ccc} & \multicolumn{3}{c}{\mathbf{0}} \\ \hline -\lambda^\mathrm{sub}_1(\boldsymbol{y}^\mathrm{q}) & & \mathbf{0} & \\ & \ddots & & \\ \mathbf{0} & & -\lambda^\mathrm{sub}_{|\mathcal{P}_\mathrm{link}|}(\boldsymbol{y}^\mathrm{q}). & \end{array}\right), \tag{6.87}$$

we have the following sensitivities of the submodel's primal and dual variables $\boldsymbol{x}^\mathrm{q} \in (\mathbb{R}^+_0)^{|\mathcal{S}_\mathrm{b}|}$, $\boldsymbol{\lambda}^\mathrm{zero} \in (\mathbb{R}^+_0)^{|\mathcal{S}_\mathrm{b}|}$, $\boldsymbol{\lambda}^\mathrm{cap} \in (\mathbb{R}^+_0)^{|\mathcal{S}_\mathrm{b}|}$, $\boldsymbol{\lambda}^\mathrm{net\_prod} \in (\mathbb{R}^+_0)^{|(\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}) \bigcap \mathcal{P}_\mathrm{b}|}$, and $\boldsymbol{\lambda}^\mathrm{sub} \in (\mathbb{R}^+_0)^{|\mathcal{P}_\mathrm{link}|}$:

$$\begin{pmatrix} \nabla_{\boldsymbol{y}^\mathrm{q}} \boldsymbol{x}^\mathrm{q}(\boldsymbol{y}^\mathrm{q}) \\ \nabla_{\boldsymbol{y}^\mathrm{q}} \boldsymbol{\lambda}^\mathrm{zero}(\boldsymbol{y}^\mathrm{q}) \\ \nabla_{\boldsymbol{y}^\mathrm{q}} \boldsymbol{\lambda}^\mathrm{cap}(\boldsymbol{y}^\mathrm{q}) \\ \nabla_{\boldsymbol{y}^\mathrm{q}} \boldsymbol{\lambda}^\mathrm{net\_prod}(\boldsymbol{y}^\mathrm{q}) \\ \nabla_{\boldsymbol{y}^\mathrm{q}} \boldsymbol{\lambda}^\mathrm{sub}(\boldsymbol{y}^\mathrm{q}) \end{pmatrix} = -\boldsymbol{M}^{\mathrm{sub}\,-1} \boldsymbol{N}_{\boldsymbol{y}^\mathrm{q}}. \tag{6.88}$$

Similar to (6.75) and (6.80), optimal value sensitivities with respect to external prices are given by

$$\frac{\partial F^*_1}{\partial a^\pi_{p_{ex}}}(\boldsymbol{a}^\pi) = \sum_{s\in\mathcal{S}_\mathrm{b}} x^\mathrm{q}_s(\boldsymbol{a}^\pi) a^\mathrm{f}_{s,p_{ex}}, \tag{6.89}$$

and, with respect to the production bounds of processes $s \in \mathcal{S}_\mathrm{b}$, we have

$$\frac{\partial F_1^*}{\partial a_s^{\mathrm{cap}}}(\boldsymbol{a}^{\mathrm{cap}}) = \lambda_s^{\mathrm{cap}}(\boldsymbol{a}^{\mathrm{cap}}). \tag{6.90}$$

Solution sensitivities with respect to external prices and production bounds can be obtained by calculating $-\boldsymbol{M}^{\mathrm{sub}^{-1}}(\boldsymbol{N}^{\mathrm{sub}}_{\boldsymbol{a}^\pi}, \boldsymbol{N}^{\mathrm{sub}}_{\boldsymbol{a}^{\mathrm{cap}}})$, where the matrices $\boldsymbol{N}^{\mathrm{sub}}_{\boldsymbol{a}^\pi}$ and $\boldsymbol{N}^{\mathrm{sub}}_{\boldsymbol{a}^{\mathrm{cap}}}$ are modifications of $\boldsymbol{N}_{\boldsymbol{a}^\pi}$ and $\boldsymbol{N}_{\boldsymbol{a}^{\mathrm{cap}}}$ analogously to the modification $\boldsymbol{M}^{\mathrm{sub}}$ of $\boldsymbol{M}$.
Sensitivities of the price variables $x^\pi_{p_{out}}, p_{out} \in \mathcal{P}_{out} \bigcap \mathcal{P}_\mathrm{b}$, can be calculated similar to (6.79) and (6.83) by the chain rule using sensitivities of production variables.

Before discussing in the following the consequences that one can possibly draw from sensitivity analyses within the subsystem, let us make a short remark: Contrary to the objective function F_1^*, which is non-decreasing in each component $y^{\mathrm{q}}_{p_{\mathrm{link}}}, p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}$, the production variables, summarized in $\boldsymbol{x}^{\mathrm{q}}$, the price variables, summarized in $\boldsymbol{x}^\pi$, as well as the dual variables, summarized in $\boldsymbol{\lambda}$, are in general not monotonically increasing or decreasing in the input/output quantities $y^{\mathrm{q}}_{p_{\mathrm{link}}}, p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}$, of a submodel. This confutes the possible hasty presumption that production of each subsystem process should increase or, at least, remain constant as long as an input $y^{\mathrm{q}}_{p_{\mathrm{link}}}$ to a subsystem increases. Considering equations (6.89) and (6.90), this implicates that there can be changes in the monotonicity of the optimal value sensitivities regarding a fixed external price or capacity parameter when input/output quantities vary.
However, under the regularity conditions of theorem 3.2.10, for a parameter value $\boldsymbol{y}^{\mathrm{q}} = \boldsymbol{y}^{\mathrm{q}}_0$, we have once continuously differentiable functions $\boldsymbol{x}^{\mathrm{q}}(\boldsymbol{y}^{\mathrm{q}})$, $\boldsymbol{x}^\pi(\boldsymbol{y}^{\mathrm{q}})$, and $\boldsymbol{\lambda}(\boldsymbol{y}^{\mathrm{q}})$ in a neighborhood of $\boldsymbol{y}^{\mathrm{q}}_0$. The respective regularity conditions are usually fulfilled if there is no constraint turning from active to inactive, or reversely, at the input/output value $\boldsymbol{y}^{\mathrm{q}}_0$.

Let us now discuss which conclusions we can draw from a sensitivity analysis and which simplifications of the model a sensitivity analysis might justify. Note that, if we solve problem (6.13) for varying values of the parameter $\boldsymbol{y}^{\mathrm{q}}$, commonly, all sensitivities can be obtained very cheaply for each single problem since necessary calculations are made during the numerical optimization procedure. For instance, in IPOPT, the matrix $\boldsymbol{M}$ is available in factored form. For all the following considerations, it is expedient to first determine, respectively restrict as much as possible, the relevant domain of the function c, in which the optimal solution could lie, cf. section 6.5, paragraph a. Let $\{\boldsymbol{y}^{\mathrm{q}}_i = (y^{\mathrm{q}}_{1,i}, ..., y^{\mathrm{q}}_{|\mathcal{P}_{\mathrm{link}}|,i})^{\mathrm{T}},\ i \in \{1, ..., n\}\}$ be a set of input/output grid points that sufficiently covers the relevant domain.

Sensitivity analysis with respect to external prices $a^\pi_{p_{ex}}$, $p_{ex} \in \mathcal{P}_{p_{ex}}$, and capacities a^{cap}_s, $s \in \mathcal{S}_\mathrm{b}$

We first consider sensitivities at an optimal subproblem solution regarding a certain input/output quantity $\boldsymbol{y}^{\mathrm{q}}_i$. Let $\boldsymbol{x}^{\mathrm{q}*} := \boldsymbol{x}^{\mathrm{q}}(\boldsymbol{y}^{\mathrm{q}}_i)$ be the corresponding vector of optimal production quantities and let $\boldsymbol{\lambda}^{\mathrm{zero}*} := \boldsymbol{\lambda}^{\mathrm{zero}}(\boldsymbol{y}^{\mathrm{q}}_i)$, $\boldsymbol{\lambda}^{\mathrm{cap}*} := \boldsymbol{\lambda}^{\mathrm{cap}}(\boldsymbol{y}^{\mathrm{q}}_i)$ be the Lagrange

multipliers corresponding to the production bounds, $\mathbf{0} \leq \boldsymbol{x}^{\mathrm{q}} \leq \boldsymbol{a}^{\mathrm{cap}}$. We assume that strict complementarity holds. For each process $s \in \mathcal{S}_{\mathrm{b}}$ with $a_s^{\mathrm{cap}} > 0$ there are three possibilities:

1. $x_s^{\mathrm{q}*} = 0 \;\Rightarrow\; \lambda_s^{\mathrm{cap}*} = 0$, $\lambda_s^{\mathrm{zero}*} > 0$, and, $\forall\, p_{ex} \in \mathcal{P}_{ex}$, we have $\frac{\partial x_s^{\mathrm{q}*}}{\partial a_{p_{ex}}^{\pi}}(\boldsymbol{a}^{\pi}, \boldsymbol{a}_{\mathcal{S}_{\mathrm{b}}}^{\mathrm{cap}}) = 0$.
2. $x_s^{\mathrm{q}*} = a_s^{\mathrm{cap}} \;\Rightarrow\; \lambda_s^{\mathrm{zero}*} = 0$, $\lambda_s^{\mathrm{cap}*} > 0$, and, $\forall\, p_{ex} \in \mathcal{P}_{ex}$, we have $\frac{\partial x_s^{\mathrm{q}*}}{\partial a_{p_{ex}}^{\pi}}(\boldsymbol{a}^{\pi}, \boldsymbol{a}_{\mathcal{S}_{\mathrm{b}}}^{\mathrm{cap}}) = 0$.
3. $0 < x_s^{\mathrm{q}*} < a_s^{\mathrm{cap}} \;\Rightarrow\; \lambda_s^{\mathrm{zero}*} = \lambda_s^{\mathrm{cap}*} = 0$.

In 1. and 2., the solution sensitivities with respect to external prices are zero because, since strict complementarity holds, the respective active constraints must remain active after an infinitesimal parameter perturbation. See also conditions (3.31). These two cases of production variables at one of their bounds are of interest regarding model simplifications: in the most extreme case, the optimal production quantity $x_s^{\mathrm{q}}(\boldsymbol{y}_i^{\mathrm{q}})$ of a certain process $s \in \mathcal{S}_{\mathrm{b}}$ is at one of its bounds $\forall\, i \in \{1, ..., n\}$, i.e., $x_s^{\mathrm{q}}(\boldsymbol{y}_i^{\mathrm{q}}) = 0\ \forall\ i$ or $x_s^{\mathrm{q}}(\boldsymbol{y}_i^{\mathrm{q}}) = a_s^{\mathrm{cap}}\ \forall\ i$. Assuming fixed subsystem parameters, this situation justifies the fixing of the variable x_s^{q} to the respective bound even in the original problem (6.5) where changes in parameters that do not appear in the subsystem are allowed.[16] This is because the variation of the input/output quantity $\boldsymbol{y}^{\mathrm{q}}$ reflects the whole range of possible influences of latter parameters on the subsystem. Processes whose production quantity is at zero for all relevant input/output quantities $\boldsymbol{y}_i^{\mathrm{q}}$, $i \in \{1, ..., n\}$, are non-profitable and processes whose production quantity is at a_s^{cap} for all relevant input/output quantities $\boldsymbol{y}_i^{\mathrm{q}}$, $i \in \{1, ..., n\}$, are highly profitable, both independently of the exogenous influences. Highly profitable processes are obviously suited for future investments in terms of capacity expansion, at least, under the assumption that subsystem parameters will not vary too much in future. Let $s \in \mathcal{S}_{\mathrm{b}}$ be such a process that runs always at its capacity bound with Lagrange multipliers $\lambda_s^{\mathrm{cap}}(\boldsymbol{y}_i^{\mathrm{q}}) > 0\ \forall\, i \in \{1, ..., n\}$. Let us assume that there is not any external product $p_{ex} \in \mathcal{P}_{ex}$ with $a_{s,p_{ex}}^{\mathrm{f}} \neq 0$ that is included in any other process of the set $\mathcal{S}_{\mathrm{b}}$. Then, taking account of the production factors $a_{s,p_{ex}}^{\mathrm{f}}$, a Lagrange multiplier $\lambda_s^{\mathrm{cap}}(\boldsymbol{y}_i^{\mathrm{q}})$ contains information about how much the related external prices can vary without influencing the optimal production quantity x_s^{q}. This is because a change $c \in [-a_{p_{ex}}^{\pi}, \infty[$ of such an external price $a_{p_{ex}}^{\pi}$ causes a change $\max[ca_{s,p_{ex}}^{\mathrm{f}}, -\lambda_s^{\mathrm{cap}}(\boldsymbol{y}_i^{\mathrm{q}})]$ of the Lagrange multiplier $\lambda_s^{\mathrm{cap}}(\boldsymbol{y}_i^{\mathrm{q}})$. If $ca_{s,p_{ex}}^{\mathrm{f}} < -\lambda_s^{\mathrm{cap}}(\boldsymbol{y}_i^{\mathrm{q}})$, the bound becomes inactive, i.e., the production of the process s decreases. Note, however, that, under the assumptions $a_{s,p_{ex}}^{\mathrm{f}} = 0\ \forall\ s \in \mathcal{S}_{\mathrm{b}} \setminus \{s\}$ and $ca_{s,p_{ex}}^{\mathrm{f}} \geq -\lambda_s^{\mathrm{cap}}(\boldsymbol{y}_i^{\mathrm{q}})$, such a parameter perturbation changes the optimal value function F_1^* by $ca_{s,p_{ex}}^{\mathrm{f}} a_s^{\mathrm{cap}}$. Analogous conclusions can be drawn in case of non-profitable processes $s \in \mathcal{S}_{\mathrm{b}}$ with positive multipliers $\lambda_s^{\mathrm{zero}}(\boldsymbol{y}_i^{\mathrm{q}}) > 0\ \forall\, i \in \{1, ..., n\}$. Accumulating the effects of changes in all external prices, we can, starting from optimal

[16] In general, a variable x_s^{q} can diverge from a bound and revert to the bound between the grid points $\boldsymbol{y}_i^{\mathrm{q}}$, $i \in \{1, ..., n\}$, remember our above comment on the non-monotonic dependence of variables on the input/output quantity $\boldsymbol{y}^{\mathrm{q}}$. However, since there is a finite number of possible active sets, the described situation can be avoided by choosing grid points sufficiently close.

solutions corresponding to a parameter vector $\boldsymbol{a}^\pi$, determine external price parameters $\widetilde{\boldsymbol{a}}^\pi \in (\mathbb{R}_0^+)^{|\mathcal{P}_{ex}|}$ which do not cause any change in the optimal production quantities $\boldsymbol{x}^{\mathrm{q}}$, but only in the Lagrange multipliers $(\boldsymbol{\lambda}^{\mathrm{zero}}, \boldsymbol{\lambda}^{\mathrm{cap}})$:

Theorem 6.6.3. *Let $\boldsymbol{x}^{\mathrm{q}*} \in (\mathbb{R}_0^+)^{|\mathcal{S}_{\mathrm{b}}|}$ be a maximizer of problem (6.13) for an input/output parameter $\boldsymbol{y}^{\mathrm{q}} \in D \subset \mathbb{R}^{|\mathcal{P}_{\mathrm{link}}|}$ and an external price parameter $\boldsymbol{a}^\pi \in (\mathbb{R}_0^+)^{|\mathcal{P}_{ex}|}$. Let $(\boldsymbol{x}^{\mathrm{q}*}, \boldsymbol{\lambda}^{\mathrm{zero}*}, \boldsymbol{\lambda}^{\mathrm{cap}*}, \boldsymbol{\lambda}^{\mathrm{net\_prod}*}, \boldsymbol{\lambda}^{\mathrm{sub}*}) \in (\mathbb{R}_0^+)^{(3|\mathcal{S}_{\mathrm{b}}|+|(\mathcal{P}_{out} \bigcup \mathcal{P}_{mid}) \bigcap \mathcal{P}_{\mathrm{b}}|+|\mathcal{P}_{\mathrm{link}}|)}$ be a corresponding KKT-point. Let $\widetilde{\boldsymbol{a}}^\pi \in \mathcal{A}$, where*

$$\begin{aligned}\mathcal{A} :=& \{\boldsymbol{a} \in (\mathbb{R}_0^+)^{|\mathcal{P}_{ex}|} \mid \sum_{p_{ex} \in \mathcal{P}_{ex}} a^{\mathrm{f}}_{s,p_{ex}}(a_{p_{ex}} - a^\pi_{p_{ex}}) \leq \lambda^{\mathrm{zero}}_s(\boldsymbol{y}^{\mathrm{q}}) \ \forall\ s \in \mathcal{S}_{\mathrm{b}}\} \\ & \bigcap \{\boldsymbol{a} \in (\mathbb{R}_0^+)^{|\mathcal{P}_{ex}|} \mid \sum_{p_{ex} \in \mathcal{P}_{ex}} a^{\mathrm{f}}_{s,p_{ex}}(a^\pi_{p_{ex}} - a_{p_{ex}}) \leq \lambda^{\mathrm{cap}}_s(\boldsymbol{y}^{\mathrm{q}}) \ \forall\ s \in \mathcal{S}_{\mathrm{b}}\}.\end{aligned} \tag{6.91}$$

Then, with

$$\widetilde{\lambda}^{\mathrm{zero}*}_s := \lambda^{\mathrm{zero}*}_s - \mathbb{1}_{\{\lambda^{\mathrm{zero}*}_s > 0\}} \cdot \sum_{p_{ex} \in \mathcal{P}_{ex}} a^{\mathrm{f}}_{s,p_{ex}}(\widetilde{a}^\pi_{p_{ex}} - a^\pi_{p_{ex}}) \qquad \textit{and} \tag{6.92a}$$

$$\widetilde{\lambda}^{\mathrm{cap}*}_s := \lambda^{\mathrm{cap}*}_s - \mathbb{1}_{\{\lambda^{\mathrm{cap}*}_s > 0\}} \cdot \sum_{p_{ex} \in \mathcal{P}_{ex}} a^{\mathrm{f}}_{s,p_{ex}}(a^\pi_{p_{ex}} - \widetilde{a}^\pi_{p_{ex}}) \tag{6.92b}$$

$\forall\ s \in \mathcal{S}_{\mathrm{b}}$, $(\boldsymbol{x}^{\mathrm{q}*}, \widetilde{\boldsymbol{\lambda}}^{\mathrm{zero}*}, \widetilde{\boldsymbol{\lambda}}^{\mathrm{cap}*}, \boldsymbol{\lambda}^{\mathrm{net\_prod}*}, \boldsymbol{\lambda}^{\mathrm{sub}*})$ is a KKT-point of the perturbed problem with external price parameter $\widetilde{\boldsymbol{a}}^\pi$, i.e., by theorem 3.1.13, $\boldsymbol{x}^{\mathrm{q}*}$ is also a maximizer of the perturbed problem.

Proof: Complementarity at the point $(\boldsymbol{x}^{\mathrm{q}*}, \boldsymbol{\lambda}^{\mathrm{zero}*}, \boldsymbol{\lambda}^{\mathrm{cap}*}, \boldsymbol{\lambda}^{\mathrm{net\_prod}*}, \boldsymbol{\lambda}^{\mathrm{sub}*})$ implies complementaritiy at $(\boldsymbol{x}^{\mathrm{q}*}, \widetilde{\boldsymbol{\lambda}}^{\mathrm{zero}*}, \widetilde{\boldsymbol{\lambda}}^{\mathrm{cap}*}, \boldsymbol{\lambda}^{\mathrm{net\_prod}*}, \boldsymbol{\lambda}^{\mathrm{sub}*})$, cf. equation (3.11). Condition (3.10) reads for problem (6.13) as follows:

$$-\frac{\partial F_1}{\partial x^{\mathrm{q}}_i}(\boldsymbol{x}^{\mathrm{q}}) - \lambda^{\mathrm{zero}}_s + \lambda^{\mathrm{cap}}_s - \sum_{\substack{p \in \mathcal{P}_{\mathrm{b}} \bigcap \\ (\mathcal{P}_{out} \bigcup \mathcal{P}_{mid})}} \lambda^{\mathrm{net\_prod}}_s a^{\mathrm{f}}_{s,p} - \sum_{p \in \mathcal{P}_{\mathrm{link}}} \lambda^{\mathrm{sub}}_s a^{\mathrm{f}}_{s,p} = 0 \quad \forall\ s \in \mathcal{S}_{\mathrm{b}}, \tag{6.93}$$

where $\frac{\partial F_1}{\partial x^{\mathrm{q}}_i}(\boldsymbol{x}^{\mathrm{q}})$ computes analogously to $\frac{\partial F}{\partial x^{\mathrm{q}}_i}(\boldsymbol{x}^{\mathrm{q}})$ given by (6.71). Compare also equation (6.73), which states the Lagrange function of the original problem (6.5). Therefore, we must show that $\forall\ s \in \mathcal{S}_{\mathrm{b}}$

$$-\lambda^{\mathrm{zero}*}_s + \lambda^{\mathrm{cap}*}_s - \sum_{p_{ex} \in \mathcal{P}_{ex}} a^{\mathrm{f}}_{s,p_{ex}} a^\pi_{p_{ex}} = -\widetilde{\lambda}^{\mathrm{zero}*}_s + \widetilde{\lambda}^{\mathrm{cap}*}_s - \sum_{p_{ex} \in \mathcal{P}_{ex}} a^{\mathrm{f}}_{s,p_{ex}} \widetilde{a}^\pi_{p_{ex}}. \tag{6.94}$$

Case 1: $\lambda^{\mathrm{zero}*}_s = \lambda^{\mathrm{cap}*}_s = 0$. (6.92) implies $\widetilde{\lambda}^{\mathrm{zero}*}_s = \widetilde{\lambda}^{\mathrm{cap}*}_s = 0$ and, by (6.91), we have $\sum_{p_{ex} \in \mathcal{P}_{ex}} a^{\mathrm{f}}_{s,p_{ex}}(a^\pi_{p_{ex}} - \widetilde{a}^\pi_{p_{ex}}) = 0$. $\Rightarrow$ (6.94) holds.
Case 2: $\lambda^{\mathrm{zero}*}_s > 0, \lambda^{\mathrm{cap}*}_s = 0$. (6.92b) implies $\widetilde{\lambda}^{\mathrm{cap}*}_s = 0$. (6.92a) implies $\widetilde{\lambda}^{\mathrm{zero}*}_s =$

$\lambda_s^{\text{zero}*} - \sum_{p_{ex}\in\mathcal{P}_{ex}} a^{\text{f}}_{s,p_{ex}}(\widetilde{a}^{\pi}_{p_{ex}} - a^{\pi}_{p_{ex}})$, which is ≥ 0 because of (6.91). $\Rightarrow$ (6.94) holds.
Case 3: $\lambda_s^{\text{zero}*} = 0$, $\lambda_s^{\text{cap}*} > 0$. (6.92a) implies $\widetilde{\lambda}_s^{\text{zero}*} = 0$. (6.92b) implies $\widetilde{\lambda}_s^{\text{cap}*} = \lambda_s^{\text{cap}*} - \sum_{p_{ex}\in\mathcal{P}_{ex}} a^{\text{f}}_{s,p_{ex}}(a^{\pi}_{p_{ex}} - \widetilde{a}^{\pi}_{p_{ex}})$, which is ≥ 0 because of (6.91). $\Rightarrow$ (6.94) holds.

Intersecting the sets $\mathcal{A}_i$ that correspond to the varying input/output parameters $\boldsymbol{y}_i^{\text{q}}$, $i \in \{1, ..., n\}$, we get the set

$$\begin{aligned}\mathcal{A}^{\text{red}} :=& \{\boldsymbol{a} \in (\mathbb{R}_0^+)^{|\mathcal{P}_{ex}|} \mid \sum_{p_{ex}\in\mathcal{P}_{ex}} a^{\text{f}}_{s,p_{ex}}(a_{p_{ex}} - a^{\pi}_{p_{ex}}) \leq \min_{i\in\{1,...,n\}} \lambda_s^{\text{zero}}(\boldsymbol{y}_i^{\text{q}}) \;\forall\; s \in \mathcal{S}_{\text{b}}\} \\ & \bigcap \{\boldsymbol{a} \in (\mathbb{R}_0^+)^{|\mathcal{P}_{ex}|} \mid \sum_{p_{ex}\in\mathcal{P}_{ex}} a^{\text{f}}_{s,p_{ex}}(a^{\pi}_{p_{ex}} - a_{p_{ex}}) \leq \min_{i\in\{1,...,n\}} \lambda_s^{\text{cap}}(\boldsymbol{y}_i^{\text{q}}) \;\forall\; s \in \mathcal{S}_{\text{b}}\}\end{aligned} \tag{6.95}$$

of perturbed external price parameters that do not influence any of the optimal solutions $\boldsymbol{x}^{\text{q}}(\boldsymbol{y}_i^{\text{q}})$. Regarding the effect on the input/output-profit/cost relationship of the subproblem, these perturbations lead to an up or down shift of the optimal value function F_1^*, but they do not effect its derivatives $\lambda^{\text{sub}}_{p_{\text{link}}}(\boldsymbol{y}_i^{\text{q}}), p_{\text{link}} \in \mathcal{P}_{\text{link}}$ for $i \in \{1, ..., n\}$.
Since, by (6.79), price changes of products $p_{ex} \in \mathcal{P}_{ex}$ influence the prices of products $p_{out} \in \mathcal{P}_{out}$ via the changes in the production quantities, the perturbed external prices $\widetilde{\boldsymbol{a}}^{\pi}$ do not effect the price solutions $\boldsymbol{x}_i^{\pi*}$ either. Let us finally consider a certain external product $p_{ex} \in \mathcal{P}_{ex}$ and its influence on the price of a certain product $p_{out} \in \mathcal{P}_{out}$ with external demand, which is in general given by equation (6.79). By the previous discussion, we see that the sensitivity of the price variable $x^{\pi}_{p_{out}}$ with respect to the external price $a^{\pi}_{p_{ex}}$ is zero if the production of all processes s related to p_{out}, i.e., with $a^{\text{f}}_{s,p_{out}} \neq 0$, is either zero or at the capacity bound and strong complementarity holds.

Sensitivity analysis with respect to input/output quantities $y^{\text{q}}_{p_{\text{link}}}, p_{\text{link}} \in \mathcal{P}_{\text{link}}$

The optimal value sensitivity of a subproblem (6.13) with respect to an input/output component $y^{\text{q}}_{p_{\text{link}}}, p_{\text{link}} \in \mathcal{P}_{\text{link}}$, is equal to the Lagrange multiplier of the corresponding input/output constraint (6.13d), cf. (6.86). Therefore, in addition to the optimal values $F_1^*(\boldsymbol{y}_i^{\text{q}})$ of a subsystem for the input/output quantities $\boldsymbol{y}_i^{\text{q}}$, $i \in \{1, ..., n\}$, it is very interesting to analyze the respective values $\boldsymbol{\lambda}^{\text{sub}}(\boldsymbol{y}_i^{\text{q}})$.
For example, let us consider two subsystems of processes $\mathcal{S}_{\text{b}}^1$ and $\mathcal{S}_{\text{b}}^2$ with $\mathcal{S}_{\text{b}}^1 \bigcap \mathcal{S}_{\text{b}}^2 = \emptyset$ and with the same single linking product p_{link}, i.e., $\mathcal{P}^1_{\text{link}} = \mathcal{P}^2_{\text{link}} = \{p_{\text{link}}\}$. We assume to know the optimal value functions $F^*_{1,1}$ and $F^*_{1,2}$ and consider the optimization of the remaining processes $\mathcal{S} \setminus (\mathcal{S}_{\text{b}}^1 \bigcup \mathcal{S}_{\text{b}}^2)$, cf. problem (6.15). Subsystem profits can be directly compared through the functions $F^*_{1,1}$ and $F^*_{1,2}$. Furthermore, it is easy to see that, usually[17], the derivatives $\lambda^{\text{sub}}_{p_{\text{link}},1}$ and $\lambda^{\text{sub}}_{p_{\text{link}},2}$ of $F^*_{1,1}$ and $F^*_{1,2}$, respectively, are equal at an

[17] If p_{link} is a product with external demand, we assume that $\sum_{s\in\mathcal{S}\setminus(\mathcal{S}_{\text{b}}^1 \bigcup \mathcal{S}_{\text{b}}^2)} x_s^{\text{q}*} \cdot a^{\text{f}}_{s,p_{\text{link}}} - x^{\text{sub}*}_{p_{\text{link}},1} - x^{\text{sub}*}_{p_{\text{link}},2}$ is a positive quantity that is sold at a positive price $P_{p_{\text{link}}}(\sum_{s\in\mathcal{S}\setminus(\mathcal{S}_{\text{b}}^1 \bigcup \mathcal{S}_{\text{b}}^2)} x_s^{\text{q}*} \cdot a^{\text{f}}_{s,p_{\text{link}}} - x^{\text{sub}*}_{p_{\text{link}},1} - x^{\text{sub}*}_{p_{\text{link}},2})$.

optimum of problem (6.15). To show this equality, we first assume $p_{\text{link}} \in \mathcal{P}_{out}$ and that a positive quantity of the product p_{link} is sold at a positive price, i.e., constraint (6.15d) is inactive. By the KKT conditions of theorem 3.1.10, equation (3.10), we have in the regular case at an optimum of problem (6.15) for $j \in \{1, 2\}$:[18]

$$\begin{aligned} 0 = &- P_{p_{\text{link}}}\Big(\sum_{\substack{s \in \\ \mathcal{S} \setminus (\mathcal{S}_{\text{b}}^1 \bigcup \mathcal{S}_{\text{b}}^2)}} x_s^{\text{q}*} \cdot a_{s,p_{\text{link}}}^{\text{f}} - x_{p_{\text{link}},1}^{\text{sub}*} - x_{p_{\text{link}},2}^{\text{sub}*} \Big) \\ &- \Big(\sum_{\substack{s \in \\ \mathcal{S} \setminus (\mathcal{S}_{\text{b}}^1 \bigcup \mathcal{S}_{\text{b}}^2)}} x_s^{\text{q}*} \cdot a_{s,p_{\text{link}}}^{\text{f}} - x_{p_{\text{link}},1}^{\text{sub}*} - x_{p_{\text{link}},2}^{\text{sub}*} \Big) P'_{p_{\text{link}}}\Big(\sum_{\substack{s \in \\ \mathcal{S} \setminus (\mathcal{S}_{\text{b}}^1 \bigcup \mathcal{S}_{\text{b}}^2)}} x_s^{\text{q}*} \cdot a_{s,p_{\text{link}}}^{\text{f}} - x_{p_{\text{link}},1}^{\text{sub}*} - x_{p_{\text{link}},2}^{\text{sub}*} \Big) \\ &+ \lambda_{p_{\text{link}},j}^{\text{sub}}(x_{p_{\text{link}},j}^{\text{sub}*}). \end{aligned} \tag{6.96}$$

This means that, at an optimum, each marginal subsystem profit $\lambda_{p_{\text{link}},j}^{\text{sub}}(x_{p_{\text{link}},j}^{\text{sub}*})$ is equal to the marginal profit from selling the link product at the market.
Second, in case $p_{\text{link}} \in \mathcal{P}_{mid}$, we have for $j \in \{1, 2\}$:

$$0 = \lambda_{p_{\text{link}},j}^{\text{sub}}(x_{p_{\text{link}},j}^{\text{sub}*}) - \lambda_{p_{\text{link}}}^{\text{net\_prod}*}, \tag{6.97}$$

where $\lambda_{p_{\text{link}}}^{\text{net\_prod}*}$ is the Lagrange multiplier corresponding to constraint (6.15e). (6.97) means that, at an optimum, each marginal subsystem profit $\lambda_{p_{\text{link}},j}^{\text{sub}}(x_{p_{\text{link}},j}^{\text{sub}*})$ is equal to the marginal price of the link product within the system of processes $\mathcal{S} \setminus (\mathcal{S}_{\text{b}}^1 \bigcup \mathcal{S}_{\text{b}}^2)$, cf. also remark 6.6.2.

c. Conclusion

With regard to sensitivity analysis, our decomposition approach is particularly advantageous because it allows to describe the connection between a subproblem (6.13) and the main problem (6.14) by the input/output quantities of only a few linking products. After determining the relevant range of input/output quantities with respect to a subsystem, their variation within this range represents the main system's accumulated possible impact on a subsystem. In this sense, we are able to accomplish global analyses within subsystems. Without the identification of the important linking quantities, it is, in general, impossible to analyze the potential effects of unrestricted parameter variations within a large model component on an interesting component with fixed parameters. This is due to the vast number of model parameters as external prices and capacities.
Finally, we summarize how the presented sensitivity results meet the objectives that we stated in the beginning of this section:

- Computing a subsystem's sensitivities for varying input/output quantities leads to a sensitivity overview that does not assume any parameter outside the submodel to be fixed.

[18] We choose a problem formulation without the slack variable $x_{p_{\text{link}}}^{\text{sales}}$.

- Price sensitivities with respect to external prices and capacities are calculated by (6.79) and (6.83) via production sensitivities. If price sensitivities within a submodel are low over the whole range of relevant input/output quantities regarding the submodel, the related parameters are much less important than those with high or varying price sensitivities. Given a set of subsystem parameters and optimal subsystem solutions for varying input/output qunatities, the set (6.95) describes the perturbations of external prices that do not cause any change in the optimal production quantities and prices as long as all other subsystem parameters are fixed.

- Sensitivity analyses with respect to parameters of the subproblem over the whole domain of relevant input/output quantities show how far a subsystem's sensitivities depend on the input/output quantities that link it to the main system. Sensitivities with respect to these linking parameters show their immediate impact on the subsystem, cf. (6.86) and (6.88).

- We can study the influence of subsystems on the main problem (6.14) in terms of their optimal value functions F_1^* and their approximating input/output-profit/cost functions c, respectively. Regarding several submodels and comparing the values of the corresponding functions at certain input/output quantities shows how much the overall profit may be influenced by a subsystem. Furthermore, it is useful to analyze the derivatives/sensitivities of these functions since they effect the optimal solution of problem (6.14), as can be seen by the KKT conditions, cf. equations (6.96) and (6.97).

- The subsystem parameters influence the remaining system by their impact on the subproblem's optimal value function F_1^*, i.e., by their impact on the input/ouput-profit/cost function. The influence of subsystem parameters on F_1^* is given by the sensitivities (6.89) and (6.90). Let us consider a certain subsystem parameter. We distinguish three cases. Firstly, if the optimal value sensitivity with respect to this parameter is low over the whole relevant domain of input/output quantities, its influence on the remaining system is low. Secondly, if the optimal value sensitivity with respect to this parameter is highly varying over the relevant domain of input/output quantities, the parameter influences the shape of the optimal value function F_1^*. This means, it influences, in addition to the values of the function, $F_1^*(\boldsymbol{y}^{\mathrm{q}})$, its derivatives with respect to the input/output quantities, which are $\lambda^{\mathrm{sub}}_{p_{\mathrm{link}}}(\boldsymbol{y}^{\mathrm{q}}), p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}$. Thirdly, if the optimal value sensitivity with respect to this parameter is uniformly high over the whole relevant domain of input/output quantities, a change in the parameter shifts the function F_1^* up or down, but it does not effect its derivatives $\lambda^{\mathrm{sub}}_{p_{\mathrm{link}}}(\boldsymbol{y}^{\mathrm{q}}), p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}$. Regarding the approximating problems (6.14) and (6.15), a shift in the functions c effects the optimal objective values, but not the optimal production and price solutions. Changes in the derivatives $\lambda^{\mathrm{sub}}_{p_{\mathrm{link}}}(\boldsymbol{y}^{\mathrm{q}}), p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}$, such as in the second case, can considerably influence

the optimal solutions of the problems (6.14) and (6.15).

- If, in the relevant domain, the sensitivity $\lambda^{\mathrm{sub}}_{\bar{p}_{\mathrm{link}}}$ of each link product $\bar{p}_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}}$ of a subsystem depends only on the input/output quantity $y^{\mathrm{q}}_{\bar{p}_{\mathrm{link}}}$ and not on the quantities $y^{\mathrm{q}}_{p_{\mathrm{link}}}$ for $p_{\mathrm{link}} \in \mathcal{P}_{\mathrm{link}} \setminus \{\bar{p}_{\mathrm{link}}\}$, the subsystem's optimal value function is separable. The separability may be caused by subsystem variables which are independent of $\boldsymbol{y}^{\mathrm{q}}$. Such quantities can be fixed as long as subsystem parameters are not allowed to vary. This simplifies the overall problem structure and may lead to a new partition of the overall problem into subproblems.

6.7 Numerical Results 2

In each of the following four examples, we aggregate one component of the petrochemical network that we partitioned in section 6.4, cf. figure 6.17. Using the resulting approximation of the input/output-profit/cost relation for the respective component, we calculate in each example a solution of the overall network optimization problem. Finally, in example 2.5, we solve the overall problem using approximations of all four subsystems. Since, in the examples of section 6.3, we got much better fitting results for the function $c(y)$ by a polynomial than by a scaled hyperbolic tangent function, we fit the generated subsystem data in the following examples by quadratic polynomials.

Example 2.1

The first submodel that we discuss arises from the linking product 84. It includes only one possible linking product, namely product 89, see also figure 6.16. The subsystem is defined by the following products and processes and is visualized in figure 6.20.

$\mathcal{P}_{\mathrm{link\_1}} = \{84\}$
$\mathcal{P}_{\mathrm{b(link)\_1}} = \{89\}$
$\mathcal{P}_{\mathrm{b\_1}} = \{54, 57, 89, 102, 249\}$
$\mathcal{S}_{\mathrm{b\_1}} = \{51, 72, 97, 120, 121, 135, 136, 139, 165, 185, 229, 274, 275, 278, 288, 291, 305\}$

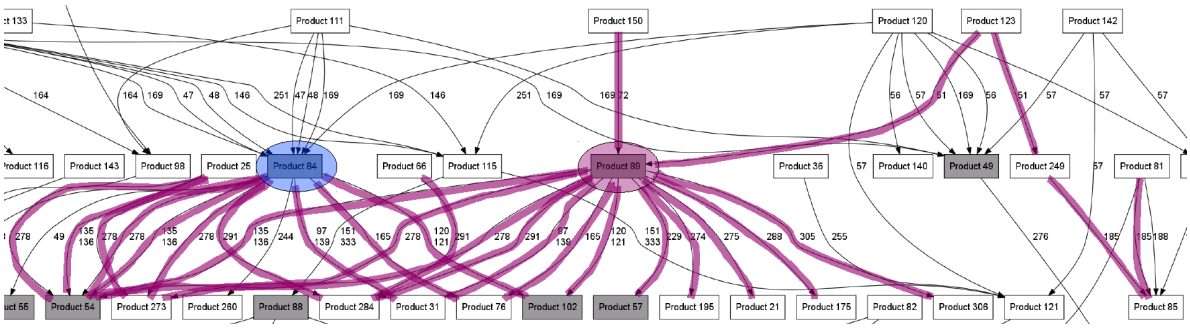

Figure 6.20: *Subsystem 1*

$y^{\mathrm{q}}_{84,i}$	0	144	288	432	576	720	864	1008
π_i	1350634	3488735	4913883	5722030	6446265	7096845	7664180	8176319
$\lambda^{\mathrm{sub}}_{84,i}$	16830	12744	6251	5069	4849	4183	3588	3558

$y^{\mathrm{q}}_{84,i}$	1151	1295	1439	1583	1727	1871	2015	2159
π_i	8688355	9197226	9698354	10192067	10670489	11034336	11241227	11257860
$\lambda^{\mathrm{sub}}_{84,i}$	3558	3508	3455	3405	3046	1901	644	0

Table 6.15: *Optimal values π_i of problem (6.13) for varying input parameters $y^{\mathrm{q}}_{84,i}$ together with sensitivities $\lambda^{\mathrm{sub}}_{84,i}$ (example 2.1). Highlighted data is used for curve fitting.*

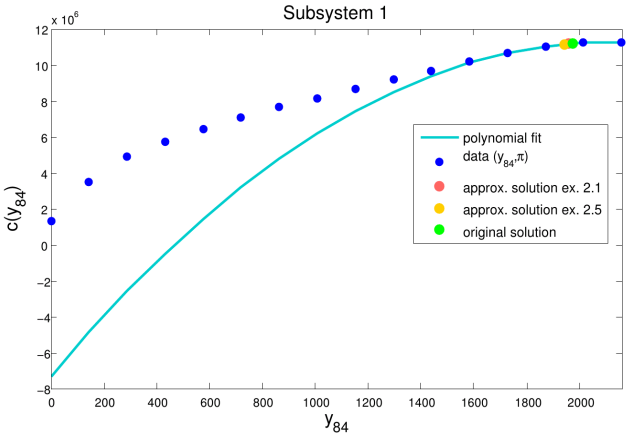

Figure 6.21: *Polynomial fit $c_{\mathrm{poly},1}$, (6.98), of subproblem 1's input/output-profit/cost function (example 2.1)*

	$z^{\mathrm{q}*}_{49}$	$x^{\mathrm{sub}*}_{49}$	$z^{\mathrm{q}*}_{52}$	$x^{\mathrm{sub}*}_{52}$	$z^{\mathrm{q}*}_{68}$	$x^{\mathrm{sub}*}_{68}$	$z^{\mathrm{q}*}_{84}$	$x^{\mathrm{sub}*}_{84}$
original problem (6.5)	-	-342,-1128	-	3372	-	-1846,2854	-	1975,-1441
subsys 1 agg. (ex. 2.1)	-	-	-	-	-	-	3112	1959
subsys 2 agg. (ex. 2.2)	-	-	-	-	437	-1606	58	-865
subsys 3 agg. (ex. 2.3)	1881	-313	-	-	5421	2854	-	-
subsys 4 agg. (ex. 2.4)	1239	-956	3381	3376	-	-	-	-
all subsys agg. (ex. 2.5)	1051	-313,-872	3381	3377	2924	-1974,2854	2445	1941,-451

Table 6.16: *Linking products' quantitative solutions of problem (6.14) with different aggregated subsystems, where $z^{\mathrm{q}*}_{p_{\mathrm{link}}} := \sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\mathrm{b}}} x^{\mathrm{q}*}_s a^{\mathrm{f}}_{s,p_{\mathrm{link}}}$. The values of $x^{\mathrm{sub}*}_{p_{\mathrm{link}}}$ in the first row are in order of the subsystems.*

Solving problem (6.13) for varying values $y^{\mathrm{q}}_{84,i}$ leads to optimal solutions $\pi_i := F_1^*(y^{\mathrm{q}}_{84,i})$ that are shown in table 6.15 together with Lagrange multipliers $\lambda^{\mathrm{sub}}_{84,i} := \lambda^{\mathrm{sub}}_{84}(y^{\mathrm{q}}_{84,i})$ that correspond to constraint (6.13d). Thereby, lower and upper bounds for $y^{\mathrm{q}}_{84,i}$ are chosen according to equations (6.29) and (6.30). In the present example, these bounds are determined by the capacities within the subsystem. Assuming a realistic system price of

	48	**49**	**50**	**52**	**54**	**55**	**57**	**61**	**68**
original problem (6.5)	777	2141	1785	1492	2472	1930	1864	1247	1322
subsys 1 agg. (ex. 2.1)	777	2141	1785	1492	2501	1930	1862	1247	1322
subsys 2 agg. (ex. 2.2)	777	2084	1785	1492	2500	1930	1862	1247	1372
subsys 3 agg. (ex. 2.3)	777	2141	1785	1492	2472	1930	1864	1247	1322
subsys 4 agg. (ex. 2.4)	777	2141	1785	1543	2472	1930	1864	1237	1322
all subsys agg. (ex. 2.5)	777	2124	1785	1543	2525	1930	1861	1236	1372

	84	**88**	**89**	**102**	**105**	**114**	**179**	**obj. value**
original problem (6.5)	1616	2336	1616	3578	2147	1307	2145	35610041
subsys 1 agg. (ex. 2.1)	1616	2336	1614	3588	2147	1307	2145	35608178
subsys 2 agg. (ex. 2.2)	1688	2336	1614	3588	2147	1307	2145	35534372
subsys 3 agg. (ex. 2.3)	1616	2336	1616	3578	2147	1307	2145	35589189
subsys 4 agg. (ex. 2.4)	1616	2336	1616	3578	2147	1298	2145	35576501
all subsys agg. (ex. 2.5)	1678	2336	1612	3597	2147	1298	2145	35517976

Table 6.17: *Price solutions $P_{p_{out}}(x^{\text{sales}*}_{p_{out}})$ of problem* (6.14) *with different aggregated subsystems, where $x^{\text{sales}*}_{p_{\text{link}}} = \sum_{s\in\mathcal{S}\setminus\mathcal{S}_{\text{b}}} x^{\text{q}*}_s a^{\text{f}}_{s,p_{\text{link}}} - x^{\text{sub}*}_{p_{\text{link}}}$ for linking products and $x^{\text{sales}*}_{p_{out}} = \sum_{s\in\mathcal{S}} x^{\text{q}*}_s a^{\text{f}}_{s,p_{out}}$ for other products with external demand. Strong deviations from the original solution through the approximation error are highlighted. Gray subsystem results are calculated afterwards by solving* (6.13) *with $\boldsymbol{y}^{\text{q}} = \boldsymbol{x}^{\text{sub}*}$.*

	# variables	# equality constr.	# inequality constr.	# iterations
original problem (6.5)	92	16	8	56
subsys 1 agg. (ex. 2.1)	72	12	7	52
subsys 2 agg. (ex. 2.2)	73	14	6	49
subsys 3 agg. (ex. 2.3)	83	15	5	53
subsys 4 agg. (ex. 2.4)	81	12	7	51
all subsys agg. (ex. 2.5)	32	5	1	38

Table 6.18: *Compared to the original problem* (6.5)*, the number of variables, constraints, and iterations of problem* (6.14) *and* (6.15)*, respectively, is reduced by using subsystem aggregations.*

product 84 in an optimal solution of problem (6.14), cf. paragraph a. of section 6.5, we can already see that the submodel input should be more than 1727. This is because the shadow price of 3046 is much higher than historical prices of product 84. Therefore, we use only the last four data points for the curve fitting. The result of the least squares estimation is

$$c_{\text{poly},1}(y_{84}) = -7314280 + 17646.3y_{84} - 4.19y_{84}^2. \tag{6.98}$$

Figure 6.21 visualizes this curve along with the resulting optimal subsystem input of problem (6.14) under this subsystem approximation, which we highlight red. For comparison, it also includes the original optimal subsystem input of problem (6.5). Table 6.16 summarizes the linking products' quantitative solutions, where the first line

shows the optimal production quantities and subsystem inputs/outputs of the original problem (6.5) without any approximation.[19] The second line includes the approximated quantitative solutions that result from solving problem (6.14) with the estimated input-profit function (6.98) of our first subsystem. The approximated price solutions $P_{p_{out}}(x^{\text{sales}*}_{p_{out}})$ of this example as well as of the following examples are shown in table 6.17. The gray numbers in the second lines result from solving subproblem (6.13) with $y^{\text{q}}_{84} = x^{\text{sub}*}_{84} = 1959$, which is equal to $z^{\text{q}*}_{84} - x^{\text{sales}*}_{84}$.

Example 2.2

The next submodel is the first one that comes from two linking products, namely product 68 and product 84. It includes the possible linking product 98. In addition to the minimal set of processes $\{49, 164, 170\}$, we added the processes that include only the possible linking products 68 and 84, cf. remark 6.4.4 and remark 6.4.5. Alternatively, we could have added some of them to the third and first submodel, respectively. All in all, the submodel consists of the following products and processes and is shown in figure 6.22.

$\mathcal{P}_{\text{link\_2}} = \{68, 84\}$
$\mathcal{P}_{\text{b(link)\_2}} = \{98\}$
$\mathcal{P}_{\text{b\_2}} = \{55, 78, 98, 179\}$
$\mathcal{S}_{\text{b\_2}} = \{45, 46, 47, 48, 49, 58, 76, 164, 170, 174, 175, 206, 231, 234, 236, 244, 257, 266, 307\}$

Figure 6.23 visualizes the fitting of the data that we generated by successively optimizing the submodel for varying values $y^{\text{q}}_{68,i}$ and $y^{\text{q}}_{84,i}$. The lower bounds for these values given by (6.29) are much too low regarding a realistic overall solution because of the multitude of production and processing facilities for product 68 and product 84. Therefore we increased the lower bounds a bit to determine a reasonable domain for the submodel input/output-profit/cost function. A representative subset of the data points $(y^{\text{q}}_{68,i}, y^{\text{q}}_{84,i}, \pi_i)$ is specified in table 6.19 together with the Lagrange multipliers $\lambda^{\text{sub}}_{68,i}$ and $\lambda^{\text{sub}}_{84,i}$ corresponding to the constraints (6.13d). The estimated polynomial to model these data is

$$\begin{aligned} c_{\text{poly},2}(\boldsymbol{y}) \;=\; c_{\text{poly},2}(y_{68}, y_{84}) \;=\; & 10450500 \;+\; 1067.8y_{68} \;+\; 1260.51y_{84} \\ & -\, 0.0243y^2_{68} \;-\; 0.0463y^2_{84} \;-\; 0.0435y_{68}y_{84}. \end{aligned} \tag{6.99}$$

An optimization of the overall model under this approximation of the submodel results in an optimal submodel output -1606 of product 68 and -865 of product 84 as listed in the third row of table 6.16. The corresponding price results are shown in the third lines of table 6.17. As in the first example, the gray numbers result from solving the subproblem with the optimal subsystem input/output quantities, here $(y^{\text{q}}_{68}, y^{\text{q}}_{84}) = (x^{\text{sub}*}_{68}, x^{\text{sub}*}_{84}) = (-1606, -865)$.

[19] If there are two values in a cell because of two submodels using the same linking product, the order is according to the order of the submodels in the following examples.

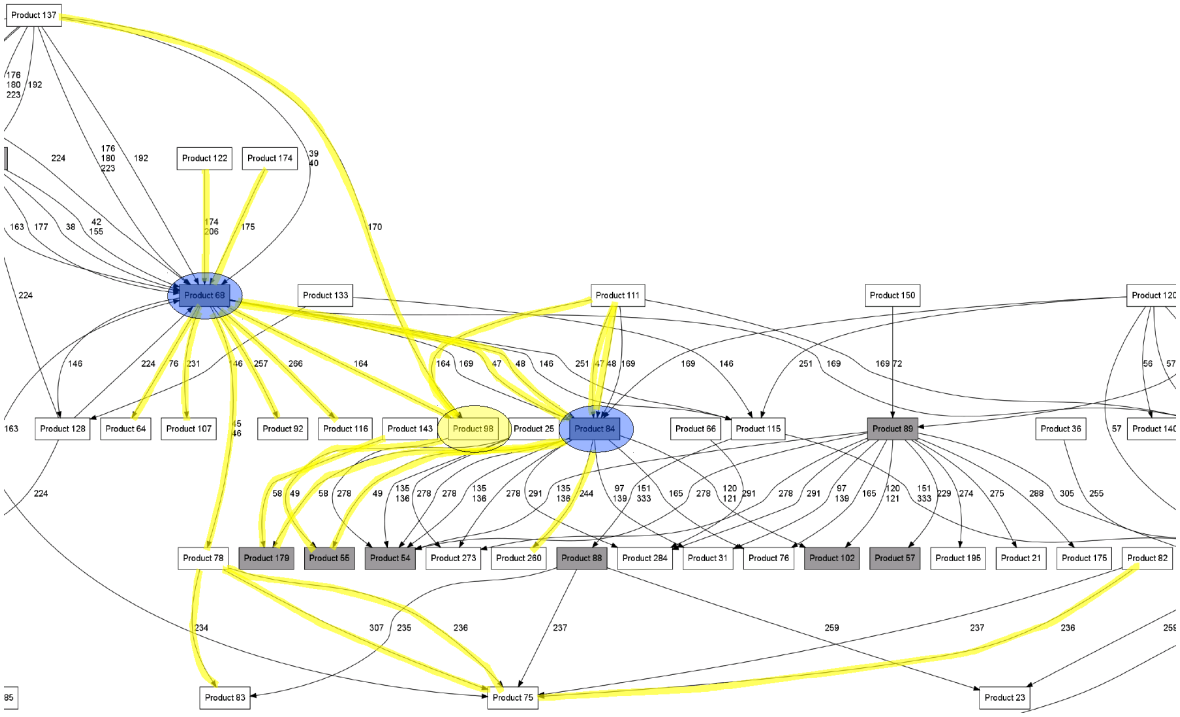

Figure 6.22: *Subsystem 2*

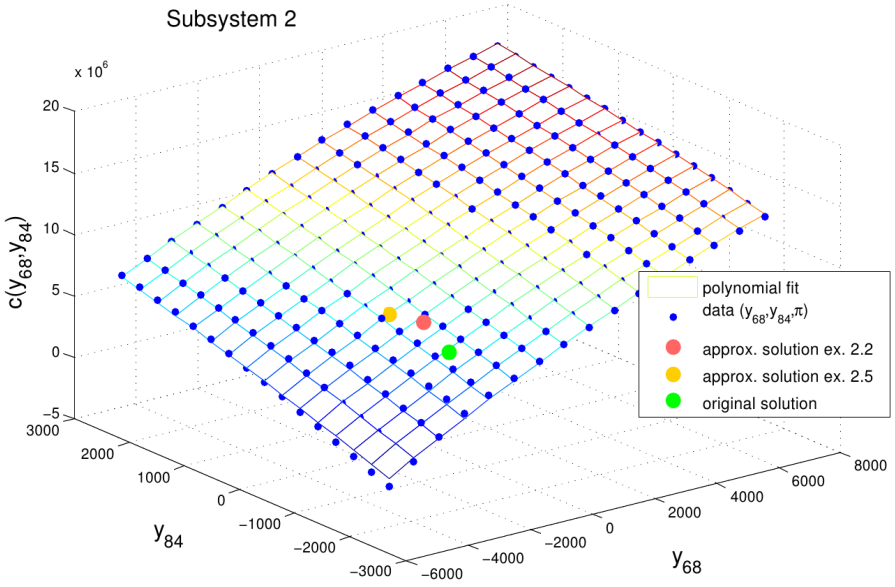

Figure 6.23: *Polynomial fit* $c_{\text{poly},2}$, (6.99), *of subproblem 2's input/output-profit/cost function (example 2.2)*

$y^{\mathrm{q}}_{68,i} \rightarrow$ $y^{\mathrm{q}}_{84,i} \downarrow$	-5452	-3021	-589	1842	4274	6705
-2375	-566943	3852450	6548281	9143955	11739634	13716359
	1906	1641	1068	1068	1068	774
	1914	1699	1615	1615	1615	1617
-1411	1252701	5308959	7908885	10504558	13100238	15075069
	1827	1148	1068	1068	1068	774
	1850	1300	1235	1235	1235	1173
-447	3027900	6503513	9099189	11694862	14180278	16062572
	1807	1068	1068	1068	774	774
	1834	1235	1235	1235	997	997
517	4734383	7693820	10289495	12885168	15141498	16984373
	1750	1068	1068	1068	774	501
	1722	1235	1235	1235	997	776
1481	6241902	8868504	11464179	14059852	16102717	17731947
	1291	1068	1068	1068	774	501
	1364	1190	1190	1190	997	776
2445	7419959	10015638	12611314	15148496	17030790	18416129
	1068	1068	1068	774	774	0
	1190	1190	1190	961	961	357

Table 6.19: *Optimal values π_i of problem* (6.13) *for varying input/output parameters $(y^{\mathrm{q}}_{68,i}, y^{\mathrm{q}}_{84,i})$ together with sensitivities $(\lambda^{\mathrm{sub}}_{68,i}, \lambda^{\mathrm{sub}}_{84,i})$ (example 2.2). All data points are used for the curve fitting.*

Example 2.3

The third submodel arises from the linking products 49 and 68 and includes three possible linking products 115 and 121. It consists of the following products and processes and is visualized in figure 6.24.

$\mathcal{P}_{\mathrm{link\_3}} = \{49, 68\}$
$\mathcal{P}_{\mathrm{b(link)\_3}} = \{115, 121\}$
$\mathcal{P}_{\mathrm{b\_3}} = \{88, 115, 121, 176\}$
$\mathcal{S}_{\mathrm{b\_3}} = \{57, 146, 151, 188, 235, 237, 251, 255, 259, 333\}$

Some of the profit/cost data π_i that we generated by optimizing the submodel for varying values $y^{\mathrm{q}}_{49,i}$ and $y^{\mathrm{q}}_{68,i}$ are listed along with Lagrange multipliers in table 6.20. Lower and upper bounds for $y^{\mathrm{q}}_{49,i}$ and $y^{\mathrm{q}}_{68,i}$ are chosen according to equations (6.29) and (6.30). As in example 2.1, we use only the data points with Lagrange multipliers that do not exceed reasonable prices of the linking products for the curve fitting. The visualization of the complete fitting situation is given by figure 6.25. The corresponding estimated polynomial is

$$c_{\mathrm{poly},3}(\boldsymbol{y}) \;=\; c_{\mathrm{poly},3}(y_{49}, y_{68}) \;=\; -478879 \;+\; 447 y_{49} \;+\; 2888.1 y_{68} \;-\; 0.2527 y^2_{68}. \tag{6.100}$$

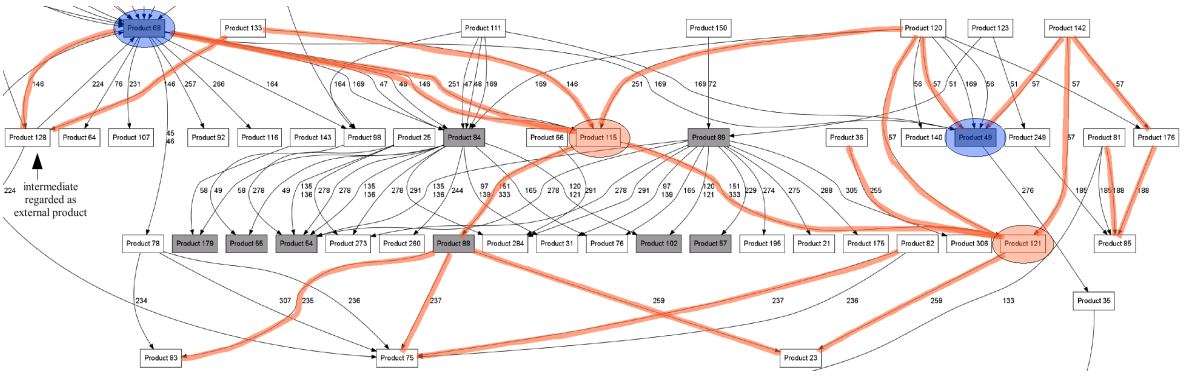

Figure 6.24: *Subsystem 3*

$y^{q}_{49,i} \rightarrow$ $y^{q}_{68,i} \downarrow$	-470	-376	-282	-188	-94	0
0	-514949	-249547	-125971	-83980	-41990	0
	2823	2823	447	447	447	0
	2583	2583	2583	2583	2583	2583
571	787268	1052670	1176247	1218237	1260227	1302217
	2823	2823	447	447	447	0
	2159	2159	2159	2159	2159	2159
1142	2019832	2285234	2408811	2450801	2492791	2534781
	2823	2823	447	447	447	0
	2159	2159	2159	2159	2159	2159
1712	3205004	3470407	3593983	3635973	3677964	3719954
	2823	2823	447	447	447	0
	2014	2014	2014	2014	2014	2014
2283	4284811	4550213	4673790	4715780	4757770	4799760
	2823	2823	447	447	447	0
	1744	1744	1744	1744	1744	1744
2854	5186757	5452160	5575736	5617726	5659717	5701707
	2823	2823	447	447	447	0
	493	493	493	493	493	493

Table 6.20: *Optimal values π_i of problem* (6.13) *for varying input/output parameters* $(y^{q}_{49,i}, y^{q}_{68,i})$ *together with sensitivities* $(\lambda^{sub}_{49,i}, \lambda^{sub}_{68,i})$ *(example 2.3). Highlighted data is used for curve fitting.*

Optimizing the overall system by using this polynomial as approximation of the submodel results in $(x^{sub*}_{49}, x^{sub*}_{68}) = (-313, 2854)$ as listed in the fourth line of table 6.16. The corresponding price results are shown in the fourth rows of table 6.17.

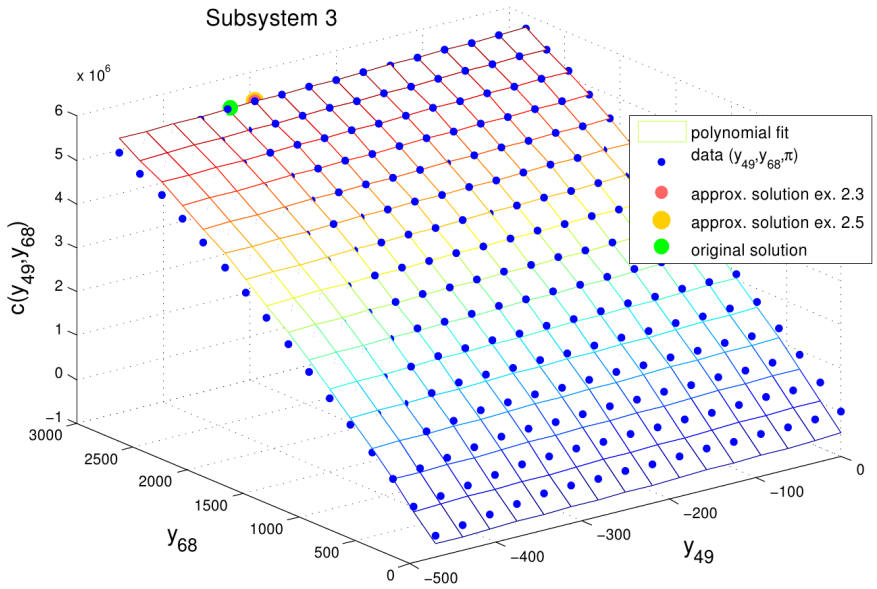

Figure 6.25: *Polynomial fit* $c_{\text{poly},3}$, (6.100), *of subproblem 3's input/output-profit/cost function (example 2.3)*

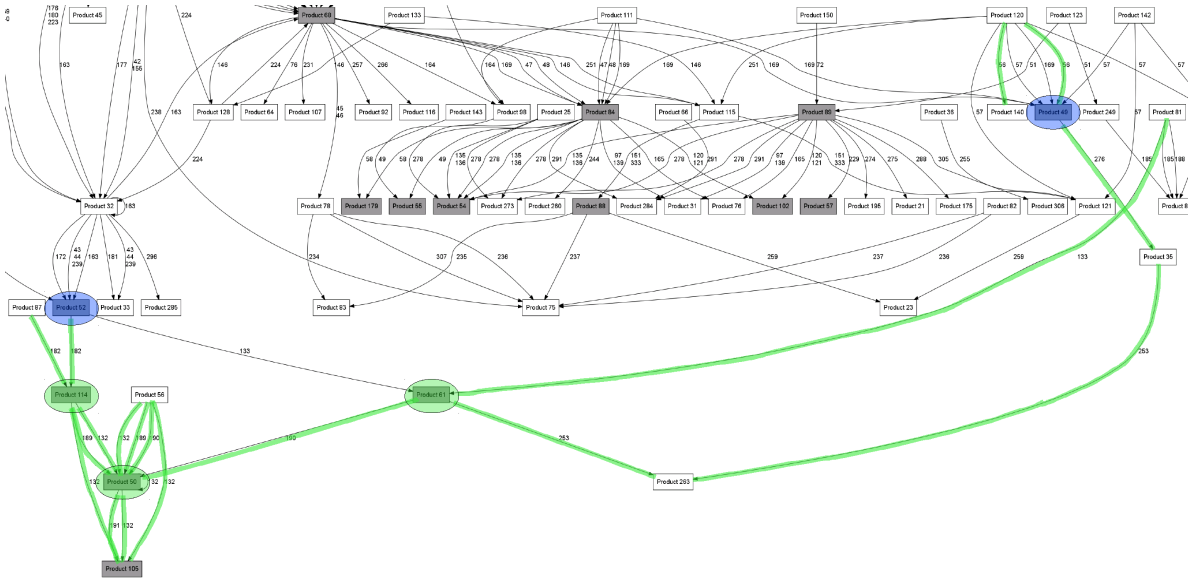

Figure 6.26: *Subsystem 4*

Example 2.4

This last submodel comes from the linking products 49 and 52 and includes three possible linking products, namely 50, 61, and 114. It consists of the following products and processes and is visualized in figure 6.26. Analogously to the assignment of processes,

$y^{\mathrm{q}}_{49,i} \rightarrow$ $y^{\mathrm{q}}_{52,i} \downarrow$	-1128	-849	-571	-293	-15	263
0	-1245617	-938328	-631039	-323750	-16461	0
	1105	1105	1105	1105	1105	0
	4032	4032	4032	4032	4032	4032
676	1306689	1613978	1921267	2228556	2535845	2552306
	1105	1105	1105	1105	1105	0
	3513	3513	3513	3513	3513	3513
1352	3493616	3800905	4108194	4415483	4722772	4739233
	1105	1105	1105	1105	1105	0
	2951	2951	2951	2951	2951	2951
2029	5348576	5655865	5963154	6270443	6577732	6594193
	1105	1105	1105	1105	1105	0
	2527	2527	2527	2527	2527	2527
2705	6896985	7204274	7511563	7818852	8126141	8142602
	1105	1105	1105	1105	1105	0
	2044	2044	2044	2044	2044	2044
3381	8098943	8406232	8713521	9020810	9328099	9357216
	1105	1105	1105	1105	1105	0
	1346	1346	1346	1346	1346	1505

Table 6.21: *Optimal values π_i of problem (6.13) for varying input/output parameters $(y^{\mathrm{q}}_{49,i}, y^{\mathrm{q}}_{52,i})$ together with sensitivities $(\lambda^{\mathrm{sub}}_{49,i}, \lambda^{\mathrm{sub}}_{52,i})$ (example 2.4). Highlighted data is used for curve fitting.*

which are not uniquely relatable, in example 2.2, we add process 56 to this fourth submodel. It is the only process connected exclusively to the possible linking products 49 and 52.

$\mathcal{P}_{\mathrm{link\_4}} = \{49, 52\}$
$\mathcal{P}_{\mathrm{b(link)\_4}} = \{50, 61, 114\}$
$\mathcal{P}_{\mathrm{b\_4}} = \{35, 50, 61, 105, 114\}$
$\mathcal{S}_{\mathrm{b\_4}} = \{56, 132, 133, 182, 189, 190, 191, 253, 276\}$

Table 6.21 and figure 6.27 show, as before, the estimation of the input/output-profit/cost relation of this submodel by a polynomial, which results in

$$\begin{aligned} c_{\mathrm{poly},4}(\boldsymbol{y}) \;=\; c_{\mathrm{poly},4}(y_{49}, y_{52}) \;=\; &-426641 \;+\; 692.5y_{49} \;+\; 4213.2y_{52} \\ &-0.2972y_{49}^2 \;-\; 0.3987y_{52}^2 + 0.0116y_{49}y_{52}. \end{aligned} \tag{6.101}$$

Thereby, we choose the bounds for $y^{\mathrm{q}}_{49,i}$ and $y^{\mathrm{q}}_{52,i}$ again by equations (6.29) and (6.30). The function is only supposed to fit the data points with Lagrange multipliers that do not exceed reasonable prices of the linking products. Using this approximation in the optimization of the overall model, we get the price simulations that are shown in the fifth lines of table 6.17, and optimal subsystem input/output quantities $(x^{\mathrm{sub}*}_{49}, x^{\mathrm{sub}*}_{52}) = (-956, 3376)$ as listed in the fifth row of table 6.16.

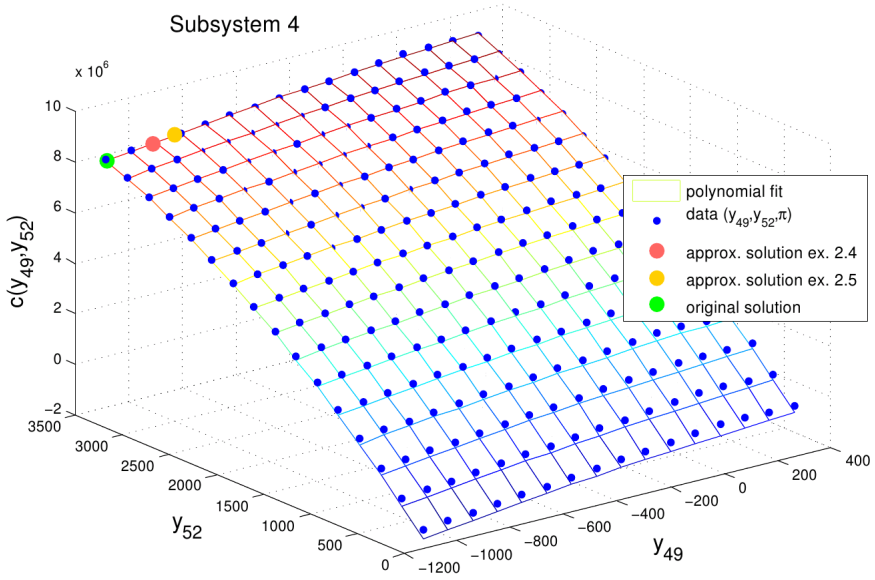

Figure 6.27: *Polynomial fit* $c_{\text{poly},4}$, (6.101), *of subproblem 4's input/output-profit/cost function (example 2.4)*

Example 2.5

In this example, we use all four submodel approximations of the above examples simultaneously, as visualized in figure 6.17. That means:

$$\mathcal{P}_{\text{link}} = \mathcal{P}_{\text{link\_1}} \bigcup \mathcal{P}_{\text{link\_2}} \bigcup \mathcal{P}_{\text{link\_3}} \bigcup \mathcal{P}_{\text{link\_4}} = \{49, 52, 68, 84\}$$
$$\mathcal{P}_{\text{b(link)}} = \mathcal{P}_{\text{b(link)\_1}} \dot{\bigcup} \mathcal{P}_{\text{b(link)\_2}} \dot{\bigcup} \mathcal{P}_{\text{b(link)\_3}} \dot{\bigcup} \mathcal{P}_{\text{b(link)\_4}}$$
$$\mathcal{P}_{\text{b}} = \mathcal{P}_{\text{b\_1}} \dot{\bigcup} \mathcal{P}_{\text{b\_2}} \dot{\bigcup} \mathcal{P}_{\text{b\_3}} \dot{\bigcup} \mathcal{P}_{\text{b\_4}}$$
$$\mathcal{S}_{\text{b}} = \mathcal{S}_{\text{b\_1}} \dot{\bigcup} \mathcal{S}_{\text{b\_2}} \dot{\bigcup} \mathcal{S}_{\text{b\_3}} \dot{\bigcup} \mathcal{S}_{\text{b\_4}}$$

The formulation of the corresponding optimization problem including multiple submodels is given by (6.15). Solving this problem, we obtain the quantitative results that are listed in the last line of table 6.16. If there are two values in a cell because of two submodels using the same linking product, the order is according to the order of the submodels. We have $z^{\text{q}*}_{p_{\text{link}}} = x^{\text{sales}*}_{p_{\text{link}}} + \sum_{m\in\mathcal{M}} x^{\text{sub}*}_{p_{\text{link}},m}$, where $z^{\text{q}*}_{p_{\text{link}}}$ is the total production of a linking product in the main part of the model, i.e., the production generated by the processes $\mathcal{S} \setminus \mathcal{S}_{\text{b}}$. The corresponding price results for the products 48, 49, 52, 68, and 84 as specified in the last lines of table 6.17. Prices for the remaining products are calculated by setting $\boldsymbol{y}^{\text{q}}_{\mathcal{P}_{\text{link}}} = \boldsymbol{x}^{\text{sub}*}_{\mathcal{P}_{\text{link}},m}$ and solving (6.13) for each submodel $m \in \mathcal{M}$.

Sensitivity Analysis

As stated in section 6.6, sensitivity analyses can be restricted to the relevant input/output quantities $\boldsymbol{y}^{\text{q}}_i$, which we have also used for fitting the input/output-profit/cost functions $c(\boldsymbol{y}^{\text{q}})$. In the following, we analyze if there are highly profitable or non-profitable pro-

cesses in a subsystem and, if existing, which simplifications these processes allow. We deduce how much several subsystem parameters can vary without influencing optimal solutions. The results considerably facilitate the data procurement for a real market simulation. Afterwards, we discuss the separability of some subsystems' input/output-profit/cost functions and possible consequences on the overall problem structure.

Highly profitable or non-profitable processes and related external products

For convenience, we discuss example 2.1, which includes only one linking product. Table 6.22 shows the Lagrange multipliers $\lambda_s^{\text{cap}}, s \in \mathcal{S}_{\text{b\_1}}$, corresponding to the capacity bounds. Ten of seventeen process variables are at their upper bound for all relevant subsystem inputs $y^{\text{q}}_{84,i}$.[20] Let $\mathcal{S}^{\text{profit}}_{\text{b\_1}}$ denote the set of these processes, i.e., $\mathcal{S}^{\text{profit}}_{\text{b\_1}} := \{51, 72, 97, 139, 165, 274, 275, 288, 291, 305\}$. Given the current values of the submodel parameters, the profitability of these ten processes is completely independent of model parameters outside the submodel. Therefore, precluding variations in submodel parameters, we can fix the production quantities of these submodel processes to their respective capacity a_s^{cap}. This fixing of select variables leads to a reduced version of the original problem (6.5), which can be used to simulate various scenarios of the overall market under the sole restriction that submodel parameters are, in general, prohibited from varying. However, as we will see in the following, there may even be exceptions from this restriction.

Remark 6.7.1. Through process 135, table 6.22 includes an example of a Lagrange multiplier that is not monotonic in y^{q}_{84}. Similarly, all other optimal solutions are not necessarily monotonic in the subsystem input. One should take this into account before fixing variables. As mentioned in section 6.6, changes in monotonicity occur at active set changes. In our example a proper analysis approves that the production quantities of the ten detected processes are at their upper bound $\forall\ y^{\text{q}}_{84} \in [1727, 2159]$. Moreover, the corresponding Lagrange multipliers are monotonic in y^{q}_{84}.

A Lagrange multiplier $\lambda_s^{\text{cap}}, s \in \mathcal{S}_{\text{b\_1}}$, states how much the optimal objective value F_1^* of the subproblem changes with the parameter a_s^{cap}. The value of λ_s^{cap} depends on the prices of the external products that the process s includes. Thus, let us look at the external prices that influence the Lagrange multipliers corresponding to the ten selected high profitable processes $\mathcal{S}^{\text{profit}}_{\text{b\_1}}$. These are $a^{\pi}_{p_{ex}},\ p_{ex} \in \{21, 31, 66, 76, 123, 150, 175, 195, 284, 306\}$ $=: \mathcal{P}^{\text{profit}}_{ex}$. None of these external products is included in any other process $s \in \mathcal{S} \setminus \mathcal{S}^{\text{profit}}_{\text{b\_1}}$. Furthermore, except for process 291, which is influenced by a^{π}_{66} and a^{π}_{284}, there is no further process that includes more than one external product. Therefore, computing the

[20] The optimal solutions $\boldsymbol{x}^{\text{q}}_i$, which we do not display here, show that there is not any process in the first subsystem with production quantity zero for all relevant values $y^{\text{q}}_{84,i}$. In general, conclusions from non-profitable processes can be drawn analogous to them from highly profitable processes, cf. the discussion in section 6.6.

$y^{\mathrm{q}}_{84,i}$	1727	1871	2015	2159	$\mathbf{a}^{\mathrm{cap}}$
$\lambda^{\mathrm{cap}}_{51}$	230	266	296	309	3214
$\lambda^{\mathrm{cap}}_{72}$	517	553	583	596	245
$\lambda^{\mathrm{cap}}_{97}$	181	817	1519	1879	1388
$\lambda^{\mathrm{cap}}_{120}$	0	0	0	0	295
$\lambda^{\mathrm{cap}}_{121}$	61	38	13	0	1247
$\lambda^{\mathrm{cap}}_{135}$	0	51	62	36	926
$\lambda^{\mathrm{cap}}_{136}$	0	0	0	0	205
$\lambda^{\mathrm{cap}}_{139}$	699	1140	1628	1879	248
$\lambda^{\mathrm{cap}}_{165}$	621	941	1299	1484	425
$\lambda^{\mathrm{cap}}_{185}$	0	0	0	0	2500
$\lambda^{\mathrm{cap}}_{229}$	0	0	0	0	811
$\lambda^{\mathrm{cap}}_{274}$	260	236	217	208	215
$\lambda^{\mathrm{cap}}_{275}$	235	212	192	184	164
$\lambda^{\mathrm{cap}}_{278}$	0	0	200	271	210
$\lambda^{\mathrm{cap}}_{288}$	152	125	103	93	28
$\lambda^{\mathrm{cap}}_{291}$	261	594	962	1151	95
$\lambda^{\mathrm{cap}}_{305}$	1399	1374	1355	1346	480

Table 6.22: *Lagrange multipliers $\lambda^{\mathrm{cap}}_s, s \in \mathcal{S}_{\mathrm{b\_1}}$ corresponding to the capacity bounds of problem* (6.13) *regarding example 2.1 for the relevant input parameters y^{q}_{84}. The last column shows process capacities $a^{\mathrm{cap}}_s, s \in \mathcal{S}_{\mathrm{b\_1}}$.*

set (6.95), we derive, except for products 66 and 284, independent ranges in which external prices can vary without influencing optimal production quantities $\boldsymbol{x}^{\mathrm{q}}$, but only Lagrange multipliers $\boldsymbol{\lambda}^{\mathrm{cap}}$, cf. table 6.23. The values regarding products 66 and 284 derive from (6.95) by taking process 291 into account. $a^{\pi\_\min}_{66}$ and $a^{\pi\_\max}_{284}$ must commonly fulfill

$$-0.4 \cdot (2700 - a^{\pi\_\max}_{66}) + (2700 - a^{\pi\_\min}_{284}) < 261, \tag{6.102}$$

i.e.,

$$a^{\pi\_\max}_{66} < (261 + 0.4 \cdot 2700 - (2700 - a^{\pi\_\min}_{284}))/0.4 = (a^{\pi\_\min}_{284} - 1359)/0.4 \quad \text{and} \tag{6.103}$$

$$a^{\pi\_\min}_{284} > -261 - 0.4 \cdot (2700 - a^{\pi\_\max}_{66}) + 2700 = 1359 + 0.4 \cdot a^{\pi\_\max}_{66}. \tag{6.104}$$

The sensitivities of the production and price solutions with respect to all external prod-

p_{ex}	21	31	66	76	123	150	175	195	284	306
$a^{\pi\_\mathrm{orig}}_{p_{ex}}$	1200	2630	2700	2500	827	813	1235	1240	2700	2393
$a^{\pi\_\min}_{p_{ex}}$	1016	2449	0	1879	0	0	1142	1032	$1359{+}0.4a^{\pi\_\max}_{66}$	1048
$a^{\pi\_\max}_{p_{ex}}$	∞	∞	$(a^{\pi\_\min}_{284}{-}1359)/0.4$	∞	934	1247	∞	∞	∞	∞

Table 6.23: *Ranges in which certain external prices can vary without influencing optimal production quantities of the first subsystem, cf.* (6.95).

ucts of table 6.23 are zero. On the contrary, optimal production quantities and prices are sensitive to other external prices as those of products 25 and 273.
We can make further conclusions considering table 6.22. The production quantities of processes 120, 136, 185, and 229 are below their upper bounds a_s^{cap} for all relevant values $y^{\text{q}}_{84,i}$. Therefore, the parameters a^{cap}_{120}, a^{cap}_{136}, a^{cap}_{185}, and a^{cap}_{229} can change, to a certain extent, without any effect on the optimal solutions and on the optimal objective value. By detecting the maximal optimal production quantities of the respective processes over all relevant values $y^{\text{q}}_{84,i}$, one can easily establish ranges in which the parameters a^{cap}_{120}, a^{cap}_{136}, a^{cap}_{185}, and a^{cap}_{229} can vary without influencing any reasonable optimal solution.

Separability of the input/output-profit/cost functions c

Table 6.20 and equation (6.100) show that the input/output-profit/cost function c_3 of subsystem 3 is additively separable. For $p_{\text{link}} \in \{49, 68\}$, the Lagrange multipliers $\lambda^{\text{sub}}_{p_{\text{link}},i}$ corresponding to the constraints (6.13d), which are in general dependent on the vector $\boldsymbol{y}^{\text{q}}_i$, depend solely on the component $y^{\text{q}}_{p_{\text{link}},i}$. We analyze the subsystem to detect the reason for the separability. It turns out that product 121 is a by-product in the processes 57 ($a^{\text{f}}_{57,121} = 0.1$), 151, and 333 ($a^{\text{f}}_{151,121} = a^{\text{f}}_{333,121} = 0.61$) and that its production by process 57 is not relevant for the subsystem. Therefore, we can change the production factor $a^{\text{f}}_{57,121}$ from 0.1 to 0, which splits subsystem 3 into 2 independent subsystems, one small subsystem with link product 49, which includes only the processes 57 and 188, and a larger one with link product 68, which includes all other processes of subsystem 3. In the network of figure 6.16, the edge between products 49 and 121 disappears.
Furthermore, table 6.21 and equation (6.101) indicate that the input/output-profit/cost function c_4 of subsystem 4 can be approximated by an additively separable function. In submodel 4, there is only a small number of values $y^{\text{q}}_{52,i}$, for which the value of $\lambda^{\text{sub}}_{52,i}$ depends on $y^{\text{q}}_{49,i}$, and, in those few cases, the influence of $y^{\text{q}}_{49,i}$ is comparatively low, cf. the last line of table 6.21: $\lambda^{\text{sub}}_{52}(-15, 3381) = 1346$ and $\lambda^{\text{sub}}_{52}(263, 3381) = 1505$. An analysis of subsystem 4 reveals that process 253 constitutes the sole connection between the linking products 49 and 52 within this subsystem. Its capacity, a^{cap}_{253}, is only 45. Therefore, we decide to fix process 253's production to half of its capacity, i.e., $x^{\text{q}}_{253} = a^{\text{cap}}_{253}/2 = 22.5$. The influence of this fixing on optimal solutions and optimal objective value is bounded by appropriate values. By this fixing, subsystem 4 decomposes into one subsystem with linking product 49 and production variables corresponding to processes 56 and 276 and one subsystem with linking product 52, which includes the remaining non-fixed production variables. In the network of figure 6.16, the edge between products 49 and 61 disappears.
Figure 6.28 shows the new network of possible linking products, which results from the discussed simplifications. Each of the new subnetworks 3 and 4 depends on only one linking product. The possible linking product 49 builds the basis of a new model component linked to the main network by the products 68 and 84. If desired, this new subsystem could be aggregated with subsystem 2 because of the same linking products.

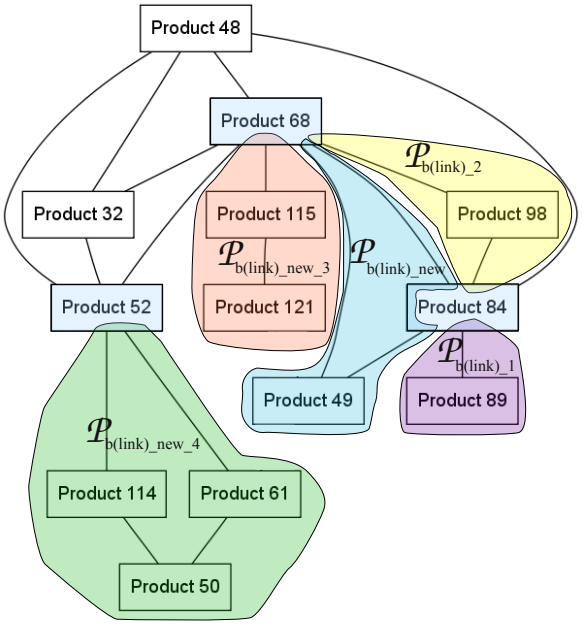

Figure 6.28: *New submodel components in the reduced network of possible linking products*

Besides the processes 56, 57, 188, and 276, the new subsystem should include process 169, which links the products 49, 68, and 84.

Conclusion

Regarding the optimal price solutions of table 6.17, we get, in principle, very satisfactory results. Using the submodel approximations of example 2.1 and 2.3 does not cause any difference in the prices of the remaining part of the network, in which we are mainly interested when choosing the respective subsystem aggregations. However, in examples 2.2 and 2.4, we get some considerable price deviations for the products 49, 54, 68, 84, and 52, respectively. These deviations reproduce in the price results of example 2.5. The differences in the quality of the submodel approximations can be explained in part by the different sizes of the domains on which we approximate the input/output-profit/cost function c, i.e., on the chosen grid points $\boldsymbol{y}_i^{\mathrm{q}}$. In the examples 2.1, 2.3, and 2.4, we can considerably restrict the relevant domains by the approach proposed in section 6.5, paragraph a. With a restriction of the domain on which the function c of the second subsystem is approximated, we expect better results. The wide domain of theoretically possible linking quantities in the second submodel is, among others, due to the assignment of "free" processes to this submodel, cf. section 6.4, paragraph c. In this case, the free processes are those related only to the linking products 68 and 84. Alternatively, one could divide this subproblem into two or three ones. Especially, there are many processes related only to product 68, of which none includes any product with external demand. These are the processes 45, 46, 76, 174, 175, 206, 231, 234, 236, 257,

266, and 307. An aggregation and a common analysis of these processes could reveal several possibilities for reduction, such as we exposed during the sensitivity analysis within the first subsystem.
Table 6.18 gives a short summary of the IPOPT console output for each optimization procedure. The number of variables, n^{var}, the number of equality constraints, n^{ec}, and the number of inequality constraints, n^{ic}, are calculated as follows:

$$n^{\text{var}} = |\mathcal{S} \setminus \mathcal{S}_{\text{b}}| + |\mathcal{P}_{out} \setminus \mathcal{P}_{\text{b}}| + |\mathcal{P}^{\mathcal{M}}_{\text{link}}|, \tag{6.105}$$

$$n^{\text{ec}} = |\mathcal{P}_{out} \setminus \mathcal{P}_{\text{b}}|, \tag{6.106}$$

$$n^{\text{ic}} = |\mathcal{P}_{mid} \setminus (\mathcal{P}_{\text{b}} \bigcup \mathcal{P}_{\text{link}})|, \tag{6.107}$$

where $|\mathcal{P}^{\mathcal{M}}_{\text{link}}|$ is the number of linking products whereby each product is weighted by the number of submodels in which it appears. For example, we have $|\mathcal{P}^{\mathcal{M}}_{\text{link}}| = 7$ in example 2.5. The corresponding variables are $\boldsymbol{x}^{\text{sub}}_{\mathcal{P}_{\text{link}},\mathcal{M}}$. The term $|\mathcal{P}_{out} \setminus \mathcal{P}_{\text{b}}|$ in n^{var} and n^{ec} comes from the additional variables $x^{\text{sales}}_{p_{\text{link}}} = \sum_{s \in \mathcal{S} \setminus \mathcal{S}_{\text{b}}} x^{\text{q}}_s a^{\text{f}}_{s,p_{\text{link}}} - x^{\text{sub}}_{p_{\text{link}}}$ for linking products and $x^{\text{sales}}_{p_{out}} = \sum_{s \in \mathcal{S}} x^{\text{q}}_s a^{\text{f}}_{s,p_{out}}$ for other products with external demand. Concerning the inequality constraints, the requirements of non-negativity for total production of products with external demand are automatically fulfilled in the optimal solution. Thus, we must postulate non-negativity only for the total production of intermediates.
The last column of table 6.18 shows the number of iterations that IPOPT needs to solve the optimization problems. First of all, we see that all numbers are very small, which means that our problem is relatively easy to solve and well formulated. Furthermore, we notice that the problems with aggregated submodels are solved even a bit faster than the original problem. This gives cause to expect good results with our model reduction approach for much larger network models, especially, when we look at the number of iterations to solve the system with four submodel aggregations, which are listed in the last line of the table.
In general, sensitivity analysis within a subsystem over the whole relevant domain of input/output quantities $\boldsymbol{y}^{\text{q}}$ is highly useful since it takes account of all reasonable variations of parameters outside the subsystem. In this sense, it is a global analysis. Regarding the first subsystem, our sensitivity analyses reveal various possible model simplifications and support the differentiation between more and less important data for simulations. Ten of seventeen processes run at their production capacities for all relevant input quantities of the linking product. The profitability of these processes depends solely on subsystem parameters. Therefore, these processes can be fixed for further analysis of the overall model in which variations of the parameters outside the subsystem are allowed. Furthermore, we are able to determine ranges in which several external prices and capacity parameters can vary without any influence on optimal prices and production quantities. Regarding real applications, our modeling, simulation, and analysis support suppliers' investment decisions since the detected highly profitable processes indicate investment opportunities.
Moreover, our numerical study shows that the input/output-profit/cost functions of the

third and fourth subsystem can be modeled additively separable. Further submodel analyses reveal the weak connections within the subsystems. Corresponding small modifications of the overall model lead to a reduced connectivity of the overall system, which is reflected on a new network of possible linking products. A new favorable partition into small subsystems, which are connected by only a few linking products, becomes possible.

7 Conclusions and Outlook

Our first aim was the development of a multi-commodity market model that allows for the simulation and analysis of price formation, sensitivity analyses with respect to different kinds of market parameters, and various scenario simulations. In this thesis, we proposed a deterministic optimization model that represents the profit maximization of the supply side given the price-demand relationship for products that are sold to consumers. This fundamental modeling approach allows high flexibility at the level of detail and also further model extensions. The simulation results for a small part of the petrochemical market are satisfactory and confirm our model, although there is room for improvement in some aspects. This might be due to the straightforward choice of demand parameters and insufficient availability of supply-side data. We expect better results by using demand functions whose parameters are estimated on an adequate data set.
Future applications of our model could include fixed costs for running plants and storage possibilities. Moreover, consumers with distinct demand behavior could be considered. We disregarded these model components in our simulations because of the lack of corresponding data.

To analyze the structure of constrained nonlinear optimization problems, we proposed a graph-theoretical decomposition approach. It allows for the exploration and comparison of possible decompositions into different sets of subproblems. Regarding parametric optimization problems, the approach facilitates the detailed analysis of subproblems while keeping the connection to the remaining part of the problem through only a small number of variables. The application of the approach to the multi-commodity market model, which is a constrained convex optimization problem with a large number of parameters, led to promising results.
Future research should apply the graph-theoretical decomposition approach to a set of benchmark problems including mixed-integer nonlinear programs in order to analyze, in a general way, its usefulness regarding efficient numerical optimization. It should be studied how far the choice of a decomposition influences the performance of an optimization procedure and if balanced decompositions are particularly advantageous. In this context, the benefit of different kinds of subproblem approximations can also be analyzed. In addition, there is a distinct interesting direction of research in the context of the reduction of optimization problems, which we would like to mention at this point. There might be a way to adapt some of the existing model reduction methods for dynamical systems, e.g., [Maa98, DHM96], to optimization problems. However, in our view, there is no evident way to treat inequality constraints. Furthermore, the approach proposed in this thesis is preferable concerning our market model, since the meaning of

the newly introduced variables is easily comprehensible, whereas the variables appearing in dynamical reduction methods are artificial.

We adapted the graph-theoretical decomposition approach to a special case of our multi-commodity market model. For the formulation of the decomposed problem, we chose the input/output quantities of selected products as linking variables between the subproblems and the remaining problem. A set of products must fulfill together with a set of processes certain properties to constitute a subsystem that is suited for aggregation. We developed an algorithm that identifies such sets. To approximate the input/output-profit/cost function of a market component that is suited for aggregation, the respective subproblem was solved for an appropriate set of fixed input/output quantities of the linking products. To generate subsystem data, we further deduced an alternative subproblem formulation, in which intrasystem prices represent the linking variables. The two formulations correspond to primal and dual decomposition. Since they have different benefits, it is worthwhile to bear both in mind regarding a subproblem approximation.
The main advantage of the decomposition approach lies in the small number of variables that represent the major connecting quantities within the overall problem. The input/output-profit/cost relationship describes the optimal value of a subproblem dependent on the values of the linking variables. The only way in which parameters of a subsystem effect the main system is through the input/output-profit/cost function. The impact of subsystem parameters on this function can be analyzed by the optimal value sensitivities of the subproblem. Conversely, the variation of parameters outside the subsystem affects the subsystem solely by the effect on the linking variables. The solution sensitivities within a subproblem with respect to input/output quantities quantify these effects. Summarizing these two aspects, sensitivity analyses within subsystems considerably increase the understanding of the relationship between a subsystem and the remaining system. Furthermore, these analyses show which parameters have little to no effect in view of a certain question and, thus, must be just roughly estimated for the respective market simulations.
Applying the approach to a part of the real petrochemical market, we could separate four subsystems. For the first submodel, we determined considerable ranges in which several parameters can vary without influencing the price simulations. In addition, two input/output-profit/cost functions turned out to be separable. The fixing of the respective subproblem variables simplifies the overall problem structure and leads to new weakly connected market components that are suited for aggregation.
In future, our decomposition approach should be applied to larger parts of the petrochemical market and also to different commodity markets. Furthermore, the approach should be adapted to models that include multiple regions and time periods. A decomposition seems also useful with regard to testing demand models and other model components. Solving a subproblem for varying input/output quantities leads to an overview of possible price and sales solutions within the subproblem, i.e., to ranges in which simulated submodel prices and sales must lie. This fact can be used to test if

a submodel is reasonable: the simulated range of submodel prices and sales should include the historical price and sales data. A rather different application area for our graph-theoretical decomposition approach is in biochemistry. Since metabolic networks are, due to the underlying chemical processes, quite similar to petrochemical production networks, our approach might be usefully applied within the flux-balance analysis of metabolic networks [PRP04, OTP10].

List of Figures

List of Tables

Notation

$\subseteq$	subset of a set (including equality)
$\subset$	proper subset of a set
$\supseteq$	superset of a set (including equality)
$\supset$	proper superset of a set
$\mathbb{N}$	positive integers
$\mathbb{N}_0$	non-negative integers, $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$
$\mathbb{R}$	real numbers
$\mathbb{R}^+$	positive real numbers, $\mathbb{R}^+ = \{x \in \mathbb{R} \vert x > 0\}$
$\mathbb{R}_0^+$	non-negative real numbers, $\mathbb{R}_0^+ = \{x \in \mathbb{R} \vert x \geq 0\}$
$B(\boldsymbol{x}, r)$	open ball with center $\boldsymbol{x}$ and radius r
$\vert A \vert$	cardinality of a set A
$\mathbb{1}_A$	indicator function of set A
$\mathrm{co}(A)$	convex hull of a set A, i.e., smallest convex set that contains A
$\mathrm{aff}(A)$	affine hull of a set A, $\mathrm{aff}(A) = \{\sum_{i=1}^{n} k_i a_i \vert n > 0, a_i \in A, k_i \in \mathbb{R}, \sum_{i=1}^{n} k_i = 1\}$
$\mathrm{relint}(A)$	relative interior of a set A, $\mathrm{relint}(A) = \{a \in A \vert \exists \epsilon > 0, B(a, \epsilon) \cap \mathrm{aff}(A) \subseteq A\}$
$\boldsymbol{M}^{\mathrm{T}}$	transpose of matrix $\boldsymbol{M}$
$\boldsymbol{I}_n$	$n \times n$ identity matrix
$\mathrm{dom}\ f$	domain of a function f
$\nabla f(\boldsymbol{x})$	gradient of the function f at the point $\boldsymbol{x}$
$\partial f(\boldsymbol{x})$	subdifferential of the function f at the point $\boldsymbol{x}$
$\mathbb{E}(X)$	expected value of a random variable X
$\mathrm{Var}(X)$	variance of a random variable X
$\mathrm{Cov}(X, Y)$	covariance of the random variables X and Y

Bibliography

[AA95] E. D. Andersen and K. D. Andersen. Presolving in linear programming. *Mathematical Programming*, 71:221–245, 1995.

[ACEP09] A. Arulselvan, C. W. Commander, L. Elefteriadou, and P. M. Pardalos. Detecting critical nodes in sparse graphs. *Computers & Operations Research*, 36(7):2193–2200, 2009.

[Alt02] W. Alt. *Nichtlineare Optimierung.* Vieweg Verlag, 2002.

[AMO93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, 1993.

[AQE09] K. Al-Qahtani and A. Elkamel. Multisite refinery and petrochemical network design: Optimal integration and coordination. *Industrial & Engineering Chemistry Research*, 48(2):814–826, 2009.

[ARV04] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 222–231, 2004.

[BC03] G. Bitran and R. Caldentey. An overview of pricing models for revenue management. *Manufacturing & Service Operations Management*, 5(3):203–229, 2003.

[BCFK08] N. M. Borisov, A. S. Chistopolsky, J. R. Faeder, and B. N. Kholodenko. Domain-oriented reduction of rule-based network models. *IET Systems Biology*, 2(5):342–351, 2008.

[BDG+08] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):172–188, 2008.

[BE05] U. Brandes and T. Erlebach, editors. *Network Analysis: Methodological Foundations.* Springer, Berlin Heidelberg, 2005.

[Ben62] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[Ber63] C. Berge. *Topological Spaces.* MacMillan, New York, 1963.

[Ber98] D. P. Bertsekas. *Network Optimization: Continuous and Discrete Models*. Athena Scientific, Belmont, Massachusetts, 1998.

[Ber99] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts, second edition, 1999.

[Ber03] D. P. Bertsekas. *Convex Analysis and Optimization*. Athena Scientific, Belmont, Massachusetts, 2003.

[Ber09] D. P. Bertsekas. *Convex Optimization Theory*. Athena Scientific, Belmont, Massachusetts, 2009.

[BGK+82] B. Bank, J. Guddat, D. Klatte, B. Kummer, and K. Tammer. *Nonlinear Parametric Optimization*. Akademie-Verlag, Berlin, 1982.

[BGLS06] J. F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal. *Numerical Optimization*. Springer, Berlin Heidelberg New York, second edition, 2006.

[BGN00] R. H. Byrd, J. C. Gilbert, and J. Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming A*, 89:149–185, 2000.

[BGS07] M. Burger, B. Graeber, and G. Schindlmayr. *Managing Energy Risk: An Integrated View on Power and Other Energy Markets*. John Wiley & Sons, 2007.

[BGS09] J. Burgschweiger, B. Gnädig, and M. C. Steinbach. Nonlinear programming techniques for operative planning in large drinking water networks. *Open Applied Mathematics Journal*, 3:14–28, 2009.

[BHN99] R. H. Byrd, M. E. Hribar, and J. Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

[Bie10] L. T. Biegler. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. SIAM Publications, Philadelphia, 2010.

[BJ92] T. N. Bui and C. Jones. Finding good approximate vertex and edge partitions is NP-hard. *Information Processing Letters*, 42(3):153–159, 1992.

[BK04] N. H. Bingham and R. Kiesel. *Risk-Neutral Valuation: Pricing and Hedging of Financial Derivatives*. Springer Finance, second edition, 2004.

[BKS07] H. G. Bock, E. Kostina, and J. P. Schlöder. Numerical methods for parameter estimation in nonlinear differential algebraic equations. *GAMM-Mitteilungen*, 30(2):376–408, 2007.

[BLN98] R. H. Byrd, G. Liu, and J. Nocedal. On the local behavior of an interior point method for nonlinear programming. In D. F. Griffiths, D. J. Higham, and G. A. Watson, editors, *Numerical Analysis 1997*, pages 37–56. Addison Wesley Longman, Harlow, UK, 1998.

[BLRS01] L. Bacaud, C. Lemaréchal, A. Renaud, and C. A. Sagastizábal. Bundle methods in stochastic optimal power management: A disaggregated approach using preconditioners. *Computational Optimization and Applications*, 20(3):227–244, 2001.

[BM11] M. D. Biha and M.-J. Meurs. An exact algorithm for solving the vertex separator problem. *Journal of Global Optimization*, 49(3):425–434, 2011.

[BMMN95] M. O. Ball, T. L. Magnanti, C. L. Monma, and G. L. Nemhauser, editors. *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*. Elsevier Science & Technology, 1995.

[BMW75] A. L. Brearley, G. Mitra, and H. P. Williams. Analysis of mathematical programming problems prior to applying the simplex algorithm. *Mathematical Programming*, 8:54–83, 1975.

[Bod12] A. Bodea. *Valuation of Swing Options in Electricity Commodity Markets*. PhD thesis, Naturwissenschaftlich-Mathematische Gesamtfakultät, Ruprecht-Karls-Universität Heidelberg, 2012.

[Bon72] P. Bonacich. Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2(1):113–120, 1972.

[Bor06] S. P. Borgatti. Identifying sets of key players in a social network. *Computational and Mathematical Organization Theory*, 12(1):21–34, 2006.

[BP84] H. G. Bock and K. J. Plitt. A Multiple Shooting algorithm for direct solution of optimal control problems. In *Proceedings of the 9th IFAC World Congress*, pages 242–247, Budapest, 1984. Pergamon Press.

[BS00] J. F. Bonnans and A. Shapiro. *Perturbation Analysis of Optimization Problems*. Springer, New York, 2000.

[BSH11] C. Borraz-Sánchez and D. Haugland. Minimizing fuel cost in gas transmission networks by dynamic programming and adaptive discretization. *Computers & Industrial Engineering*, 61(2):364–372, 2011.

[BTN] A. Ben-Tal and A. Nemirovski. Lectures on modern convex optimization, spring semester 2012. Extension of their book *Lectures on Modern Convex Optimization: Analysis, Algorithms, Engineering Applications, SIAM, Philadelphia, 2001.*

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[BXMM07] S. Boyd, L. Xiao, A. Mutapcic, and J. Mattingley. Notes on decomposition methods. *Lecture Notes for EE364B: Convex Optimization II, Stanford University, Winter 2006-07*, 2007.

[CCC+06] E. Castillo, A. J. Conejo, C. Castillo, R. Mínguez, and D. Ortigosa. Perturbation approach to sensitivity analysis in mathematical programming. *Journal of Optimization Theory and Applications*, 128(1):49–74, 2006.

[CCMGB06] A. J. Conejo, E. Castillo, R. Mínguez, and R. García-Bertrand. *Decomposition Techniques in Mathematical Programming: Engineering and Science Applications.* Springer, Berlin Heidelberg, 2006.

[CGJ+13] M. Chimani, C. Gutwenger, M. Jünger, G. W. Klau, K. Klein, and P. Mutzel. The open graph drawing framework (OGDF). In R. Tamassia, editor, *Handbook of Graph Drawing and Visualization*, chapter 17. Chapman & Hall/CRC Press, 2013.

[Cha05] A. K. Chakravarty. Global plant capacity and product allocation with pricing decisions. *European Journal of Operational Research*, 165:157–181, 2005.

[Cla83] F. H. Clarke. *Optimization and Nonsmooth Analysis.* John Wiley & Sons, 1983.

[CLCD07] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, 95(1):255–312, 2007.

[CMLW01] X. Cai, D. C. McKinney, L. S. Lasdon, and D. W. Watkins. Solving large nonconvex water resources management models using generalized Benders decomposition. *Operations Research*, 49(2):235–245, 2001.

[CSB+11] W.-Y. Chen, Y. Song, H. Bai, C.-J. Lin, and E. Y. Chang. Parallel spectral clustering in distributed systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):568–586, 2011.

[CSRS+06] H. Conzelmann, J. Saez-Rodriguez, T. Sauter, B. N. Kholodenko, and E. D. Gilles. A domain-oriented approach to the reduction of combinatorial complexity in signal transduction networks. *BMC Bioinformatics*, 7:34, 2006.

[DDGDA05] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(9):P09008, 2005.

[DHM96] P. Deuflhard, J. Heroth, and U. Maas. Towards dynamic dimension reduction in reactive flow problems. *Proc. 3rd Workshop on Modelling of Chemical Reaction Systems, Heidelberg*, 1996.

[Dij59] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.

[dSB05] C. de Souza and E. Balas. The vertex separator problem: algorithms and computations. *Mathematical Programming A*, 103(3):609–631, 2005.

[DW60] G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8:101–111, 1960.

[DXT+10] T. N. Dinh, Y. Xuan, M. T. Thai, E. K. Park, and T. Znati. On approximation of new optimization methods for assessing network vulnerability. In *INFOCOM, 2010 Proceedings IEEE*, 2010.

[EB99] M. G. Everett and S. P. Borgatti. The centrality of groups and classes. *Journal of Mathematical Sociology*, 23(3):181–201, 1999.

[EFS12] M. Engelhart, J. Funke, and S. Sager. A decomposition approach for a new test-scenario in complex problem solving. *Journal of Computational Science*, 2012. Available at: http://dx.doi.org/10.1016/j.jocs.2012.06.005.

[EG70] J. P. Evans and F. J. Gould. Stability in nonlinear programming. *Operations Research*, 18(1):107–118, 1970.

[EK10] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[EM92] J. R. Evans and E. Minieka. *Optimization Algorithms for Networks and Graphs*. Marcel Dekker, second edition, 1992.

[Eve63] H. Everett. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11(3):399–417, 1963.

[Eve79] S. Even. *Graph Algorithms.* Computer Science Press, 1979.

[EW03] A. Eydeland and K. Wolyniec. *Energy and Power Risk Management: New Developments in Modeling, Pricing, and Hedging.* John Wiley & Sons, 2003.

[FF56] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[FGK03] R. Fourer, D. M. Gay, and B. W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming.* Duxbury, second edition, 2003.

[FGW02] A. Forsgren, P. E. Gill, and M. H. Wright. Interior methods for nonlinear optimization. *SIAM Review*, 44(4):525–597, 2002.

[FHL08] U. Feige, M. Hajiaghayi, and J. R. Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM Journal on Computing*, 38(2):629–657, 2008.

[FHT96] L. Fahrmeir, A. Hamerle, and G. Tutz. *Multivariate statistische Verfahren.* Walter de Gryter & Co., second edition, 1996.

[FI90] A. V. Fiacco and Y. Ishizuka. Sensitivity and stability analysis for nonlinear programming. *Annals of Operations Research*, 27:215–236, 1990.

[Fia83] A. V. Fiacco. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming.* Academic Press, New York, 1983.

[Fie73] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.

[FK86] A. V. Fiacco and J. Kyparisis. Convexity and concavity properties of the optimal value function in parametric nonlinear programming. *Journal of Optimization Theory and Applications*, 48(1):95–126, 1986.

[FL02] R. Fletcher and S. Leyffer. Nonlinear programming without a penalty function. *Mathematical Programming*, 91(2):239–269, 2002.

[Fle87] R. Fletcher. *Practical Methods of Optimization.* John Wiley & Sons, second edition, 1987.

[Flo62] R. W. Floyd. Algorithm 97: Shortest path. *Communications of the ACM*, 5(6):345, 1962.

[Flo95] C. A. Floudas. *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications.* Oxford University Press, 1995.

[FM90] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. SIAM, Philadelphia, PA, 1990. Reprint of the 1968 original.

[Fox08] J. Fox. *Applied Regression Analysis and Generalized Linear Models*. Sage, second edition, 2008.

[Fre77] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.

[FT87] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3):596–615, 1987.

[FTT04] G. W. Flake, R. E. Tarjan, and K. Tsioutsiouliklis. Graph clustering and minimum cut trees. *Internet Mathematics*, 1(4):385–408, 2004.

[FW08] J. Ferrio and J. Wassick. Chemical supply chain network optimization. *Computers and Chemical Engineering*, 32:2481–2504, 2008.

[Gan11] E. R. Gansner. Drawing Graphs with Graphviz: Graphviz Drawing Library Manual. 2011.

[Gem05] H. Geman. *Commodities and Commodity Derivatives*. John Wiley & Sons, 2005.

[Geo72] A. M. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

[GG97] T. Gal and H. J. Greenberg, editors. *Advances in Sensitivity Analysis and Parametric Programming*. Kluwer Academic Publishers, 1997.

[GH61] R. E. Gomory and T. C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961.

[GK07] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. *SIAM Journal on Computing*, 37(2):630–652, 2007.

[GM01] C. Gutwenger and P. Mutzel. A linear time implementation of SPQR-trees. In *Proceedings of the 8th International Symposium on Graph Drawing (GD 2000)*, volume 1984 of *Lecture Notes in Computer Science*, pages 77–90. Springer, 2001.

[GMZ03] C. Gkantsidis, M. Mihail, and E. Zegura. Spectral analysis of Internet topologies. In *Proceedings of the 22nd IEEE International Conference on Computer Communications, INFOCOM*, pages 364–374, 2003.

[GN02] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[GN04] M. Girvan and M. E. J. Newman. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

[GOST01] N. I. M. Gould, D. Orban, A. Sartenaer, and P. L. Toint. Superlinear convergence of primal-dual interior point algorithms for nonlinear programming. *SIAM Journal on Optimization*, 11(4):974–1002, 2001.

[GOT05] N. I. M. Gould, D. Orban, and P. L. Toint. Numerical methods for large-scale nonlinear optimization. *Acta Numerica*, 14:299–361, 2005.

[GT88] A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *Journal of the Association for Computing Machinery*, 35(4):921–940, 1988.

[GvL96] G. H. Golub and C. F. van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.

[GvR97] G. Gallego and G. J. van Ryzin. A multiproduct dynamic pricing problem and its applications to network yield management. *Operations Research*, 45(1):24–41, 1997.

[Hac08] S. T. Hackman. *Production Economics: Integrating the Microeconomic and Engineering Perspectives*. Springer, Berlin Heidelberg, 2008.

[HFH+11] T. Huschto, G. Feichtinger, R. F. Hartl, P. M. Kort, S. Sager, and A. Seidl. Numerical solution of a conspicuous consumption model with constant control delay. *Automatica*, 47(9):1868–1877, 2011.

[HT73] J. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM Journal on Computing*, 2(3):135–158, 1973.

[Hüb07] R. Hübner. *Strategic Supply Chain Management in Process Industries: An Application to Specialty Chemicals Production Network Design*. Springer, Berlin Heidelberg, 2007.

[HUL93] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Springer, Berlin Heidelberg, 1993.

[Ipo] *Introduction to IPOPT: A tutorial for downloading, installing, and using IPOPT*. http://www.coin-or.org/Ipopt/documentation/.

[JM04] M. Jünger and P. Mutzel, editors. *Graph Drawing Software*. Springer, Berlin Heidelberg, 2004.

[Kal02] J. Kallrath. *Gemischt-ganzzahlige Optimierung: Modellierung in der Praxis*. Vieweg, 2002.

[Kan08] M. Kannegiesser. *Value Chain Management in the Chemical Industry: Global Value Chain Planning of Commodities*. Physica-Verlag Heidelberg, 2008.

[KC08] T.-H. Kuo and C.-T. Chang. Optimal planning strategy for the supply chains of light aromatic compounds in petrochemical industries. *Computers & Chemical Engineering*, 32:1147–1166, 2008.

[KC12] M. Kim and K. S. Candan. SBV-Cut: Vertex-cut based graph partitioning using structural balance vertices. *Data & Knowledge Engineering*, 72:285–303, 2012.

[Kel60] J. E. Kelley. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

[Kel13] S. Kellner. *Modeling and Analysis of Demand for Commodities and a Case Study of the Petrochemical Market*. PhD thesis, Naturwissenschaftlich-Mathematische Gesamtfakultät, Ruprecht-Karls-Universität Heidelberg, 2013. (submitted in November 2013).

[KGvB+09] M. Kannegiesser, H.-O. Günther, P. van Beek, M. Grunow, and C. Habla. Value chain management for commodities: a case study from the chemical industry. *OR Spectrum*, 31:63–93, 2009.

[KL70] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49:291–307, 1970.

[KLL07] K. C. Kiwiel, T. Larsson, and P. O. Lindberg. Lagrangian relaxation via ballstep subgradient methods. *Mathematics of Operations Research*, 32(3):669–686, 2007.

[KMT98] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control for communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49(3):237–252, 1998.

[Kol09] E. D. Kolaczyk. *Statistical Analysis of Network Data: Methods and Models*. Springer US, 2009.

[KPcA12] E. Kayaaslan, A. Pinar, Ü. Çatalyürek, and C. Aykanat. Partitioning hypergraphs in scientific computing applications through vertex separators on graphs. *SIAM Journal on Scientific Computing*, 34(2):A970–A992, 2012.

[Kra09] L. Kramer. Ein Fundamentalmodell zur stochastischen Modellierung von Elektrizitätspreisen. Diploma thesis, Fakultät für Mathematik und Informatik, Ruprecht-Karls-Universität Heidelberg, 2009.

[KVV04] R. Kannan, S. Vempala, and A. Vetta. On clusterings: Good, bad and spectral. *Journal of the ACM*, 51(3):497–515, 2004.

[Las70] L. S. Lasdon. *Optimization Theory for Large Systems*. Macmillan Company, New York, 1970.

[LB08] H. Lakkhanawat and M. J. Bagajewicz. Financial risk management with product pricing in the planning of refinery operations. *Industrial & Engineering Chemistry Research*, 47:6622–6639, 2008.

[LBBS03] D. B. Leineweber, I. Bauer, H. G. Bock, and J. P. Schlöder. An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part 1: theoretical aspects. *Computers and Chemical Engineering*, 27(2):157–166, 2003.

[Lem01] C. Lemaréchal. Lagrangian relaxation. In M. Jünger and D. Naddef, editors, *Computational Combinatorial Optimization: Optimal or Provably Near Optimal Solutions*, Lecture Notes in Computer Science, pages 112–156, 2001.

[LR99] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, 1999.

[Lud13] S. Ludwig. *Optimal Portfolio Allocation of Commodity Related Assets Using a Controlled Forward-backward Stochastic Algorithm*. PhD thesis, Naturwissenschaftlich-Mathematische Gesamtfakultät, Ruprecht-Karls-Universität Heidelberg, 2013.

[Lue08] D. G. Luenberger. *Linear and Nonlinear Programming*. Springer US, third edition, 2008.

[Maa98] U. Maas. Efficient calculation of intrinsic low-dimensional manifolds for the simplification of chemical kinetics. *Computing and Visualization in Science*, 1(2):69–81, 1998.

[MC09] J. G. Martin and E. R. Canfield. Ranks and representations for spectral graph bisection. *SIAM Journal on Scientific Computing*, 31(5):3529–3546, 2009.

[Men27] K. Menger. Zur allgemeinen Kurventhorie. *Fundamenta Mathematicae*, 10:96–115, 1927.

[MHH+04] J. Mohring, J. Hoffmann, T. Halfmann, A. Zemitis, G. Basso, and P. Lagoni. Automated model reduction of complex gas pipeline networks. In *Proceedings of the 36th Annual Meeting of the Pipeline Simulation Interest Group*, Palm Springs, California, 2004.

[Min01] M. Minoux. Discrete cost multicommodity network optimization problems and exact solution methods. *Annals of Operations Research*, 106:19–46, 2001.

[MSOI+02] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: Simple building blocks of complex networks. *Science*, 298:824–827, 2002.

[NB01] A. Nedić and D. P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12(1):109–138, 2001.

[Nes05a] Y. Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM Journal on Optimization*, 16(1):235–249, 2005.

[Nes05b] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming A*, 103(1):127–152, 2005.

[New04] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6):066133, 2004.

[New06] M. E. J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[New10] M. E. J. Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[NJW02] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2002.

[NN94] Y. Nesterov and A. Nemirovski. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM Publications, Philadelphia, 1994.

[NO08] A. Nedić and A. Ozdaglar. Approximate primal solutions and rate analysis for dual subgradient methods. *SIAM Journal on Optimization*, 19(4):1757–1780, 2008.

[Now05] I. Nowak. *Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming.* Birkhäuser, Basel Boston Berlin, 2005.

[NS08] I. Necoara and J. A. K. Suykens. Application of a smoothing technique to decomposition in convex optimization. *IEEE Transactions on Automatic Control*, 53(11):2674–2679, 2008.

[NT08] A. S. Nemirovski and M. J. Todd. Interior-point methods for optimization. *Acta Numerica*, 17:191–234, 2008.

[NW06] J. Nocedal and S. J. Wright. *Numerical Optimization.* Springer Science+Business Media, second edition, 2006.

[Oel87] O. R. Oellermann. On the l-connectivity of a graph. *Graphs and Combinatorics*, 3:285–291, 1987.

[OMV00] A. Ouorou, P. Mahey, and J.-P. Vial. A survey of algorithms for convex multicommodity flow problems. *Management Science*, 46(1):126–147, 2000.

[OTP10] J. D. Orth, I. Thiele, and B. Ø. Palsson. What is flux balance analysis? *Nature biotechnology*, 28(3):245–248, 2010.

[PC06] D. P. Palomar and M. Chiang. Alternative decompositions for distributed maximization of network utility: Framework and applications. In *Proceedings of the 25th IEEE International Conference on Computer Communications, INFOCOM*, 2006.

[PED07] R. Puzis, Y. Elovici, and S. Dolev. Fast algorithm for successive computation of group betweenness centrality. *Physical Review E*, 76(5), 2007. Id: 056709.

[Pil07] D. Pilipovic. *Energy Risk: Valuing and Managing Energy Derivatives.* McGraw-Hill, second edition, 2007.

[PLNB11] H. Pirnay, R. López-Negrete, and L. T. Biegler. sIPOPT Reference Manual. 2011.

[PLNB12] H. Pirnay, R. López-Negrete, and L. T. Biegler. Optimal sensitivity based on IPOPT. *Mathematical Programming Computation*, 4(4):307–331, 2012.

[PR10] R. S. Pindyck and D. L. Rubinfeld. *Microeconomics*. Prentice Hall, 7th edition, 2010.

[PRP04] N. D. Price, J. L. Reed, and B. Ø. Palsson. Genome-scale models of microbial cells: evaluating the consequences of constraints. *Nature Reviews Microbiology*, 2(11):886–897, 2004.

[PSL90] A. Pothen, H. D. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM Journal on Matrix Analysis and Applications*, 11(3):430–452, 1990.

[Ras11] S. Rasmussen. *Production Economics: The Basic Theory of Production Optimisation*. Springer, Berlin Heidelberg, 2011.

[RMWSB02] R. Z. Ríos-Mercado, S. Wu, L. R. Scott, and E. A. Boyd. A reduction technique for natural gas transmission network optimization problems. *Annals of Operations Research*, 117:217–234, 2002.

[Roc70] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[Ruj08] S. Rujivan. *Stochastic Modeling for Commodity Prices and Valuation of Commodity Derivatives under Stochastic Convenience Yields and Seasonality*. PhD thesis, Naturwissenschaftlich-Mathematische Gesamtfakultät, Ruprecht-Karls-Universität Heidelberg, 2008.

[Sab66] G. Sabidussi. The centrality index of a graph. *Psychometrica*, 31(4):581–603, 1966.

[SK07] H. Stadtler and C. Kilger, editors. *Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies*. Springer, Berlin Heidelberg, 4th edition, 2007.

[SLL02] J. G. Siek, L.-Q. Lee, and A. Lumsdaine. *The Boost Graph Library: User Guide and Reference Manual*. Addison-Wesley, 2002.

[Sno08] *User's Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming*. http://www.sbsi-sol-optimize.com/manuals/SNOPT Manual.pdf, 2008.

[SRWD+11] B. Scholz-Reiter, F. Wirth, S. Dashkovskiy, T. Makuschewitz, M. Schönlein, and M. Kosmykov. Structure-preserving model reduction of large-scale logistics networks. *European Physical Journal B*, 84(4):501–520, 2011.

[Stu02] A. Stuible. *Ein Verfahren zur graphentheoretischen Dekomposition und algebraischen Reduktion von komplexen Energiesystemmodellen*. PhD thesis, Institut für Energiewirtschaft und Rationelle Energieanwendung, Universität Stuttgart, 2002.

[SW97] M. Stoer and F. Wagner. A simple min-cut algorithm. *Journal of the ACM*, 44(4):585–591, 1997.

[Tar72] R. E. Tarjan. Depth first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[TBS12] M. Talebian, N. Boland, and M. Savelsbergh. Assortment and pricing with demand learning. *Optimization Online*, January 2012.

[TDNSD13] Q. Tran Dinh, I. Necoara, C. Savorgnan, and M. Diehl. An inexact perturbed path-following method for Lagrangian decomposition in large-scale separable convex optimization. *SIAM Journal on Optimization*, 23(1):95–125, 2013.

[TDSD13] Q. Tran Dinh, C. Savorgnan, and M. Diehl. Combining Lagrangian decomposition and excessive gap smoothing technique for solving large-scale separable convex optimization problems. *Computational Optimization and Applications*, 55(1):75–111, 2013.

[Var10] H. R. Varian. *Intermediate Microeconomics: A Modern Approach*. Norton, 8th edition, 2010.

[vL07] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[VW03] S. Voß and D. L. Woodruff. *Introduction to Computational Optimization Models for Production Planning in a Supply Chain*. Springer, Berlin Heidelberg, 2003.

[Wäc02] A. Wächter. *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2002.

[WB05] A. Wächter and L. T. Biegler. Line search filter methods for nonlinear programming: Motivation and global convergence. *SIAM Journal on Optimization*, 16(1):1–31, 2005.

[WB06] A. Wächter and L. T. Biegler. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[Whi32] H. Whitney. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, 54(1):150–168, 1932.

[ZLB08] V. M. Zavala, C. D. Laird, and L. T. Biegler. Interior-point decomposition approaches for parallel solution of large-scale nonlinear parameter estimation problems. *Chemical Engineering Science*, 63:4834–4845, 2008.

[ZSZ13] F. Zheng, A. R. Simpson, and A. C. Zecchin. A decomposition and multistage optimization approach applied to the optimization of water distribution systems with multiple supply sources. *Water Resource Research*, 49(1):380–399, 2013. doi:10.1029/2012WR013160.

[ZSZD13] F. Zheng, A. R. Simpson, A. C. Zecchin, and J. W. Deuerlein. A graph decomposition-based approach for water distribution network optimization. *Chemical Engineering Science*, 49(4):2093–2109, 2013. doi:10.1002/wrcr.20175.