INDUSTRIAL APPLICATIONS OF THE P-GRAPH FRAMEWORK

PhD Dissertation

Máté József Bárány
Supervisor: Botond Bertók

DOI: 10.18136/PE.2015.595

University of Pannonia
Faculty of Information Technology
Doctoral School of Information Science and Technology

2015

INDUSTRIAL APPLICATIONS OF THE P-GRAPH FRAMEWORK

Értekezés doktori (PhD) fokozat elnyerése érdekében

Írta: Máté József Bárány

Készült a University of Pannonia Doktori Iskolája keretében

Témavezető: Botond Bertók

Elfogadásra javaslom (igen / nem)

(aláírás)

A jelölt a doktori szigorlaton ....%-ot ért el

Veszprém ....

a Szigorlati Bizottság elnöke

Az értekezést bírálóként elfogadásra javaslom:

Bíráló neve: .... (igen / nem)

(aláírás)

Bíráló neve: .... (igen / nem)

(aláírás)

A jelölt az értekezés nyilvános vitáján ....%-ot ért el

Veszprém ....

a Bíráló Bizottság elnöke

A doktori (PhD) oklevél minősítése ....

....

Az EDT elnöke

Table of Contents

List of Tables

List of Figures

Kivonat

A P-gráf módszertan ipari alkalmazásai

Döntéshozói szinten a matematikai programozási modellek alkalmazása mindennapos a különböző ipari alkalmazásokban. Számos olyan algoritmus és megoldási módszer áll a döntéshozók rendelkezésére, amelyek segítségével egy adott probléma egzaktan, heurisztikák alkalmazása nélkül megoldható. Ennek ellenére még az egzakt módszerek által adott eredmény sem feltétlenül optimális, hiszen lehet, hogy a kiindulási modell hibásan van megkonstruálva.

A P-gráf módszertant Friedler és szerzőtársai dolgozták ki a 90-es évek elején komplex vegyipari termelőrendszerek optimalizálására. Formálisan bizonyított, hogy a módszertan által generált matematikai programozási modell garantáltan tartalmazza a feladat optimális megoldását.

A dolgozatban bemutatásra kerül, hogy a P-gráf módszertannal nemcsak komplex vegyipari termelőrendszerek, hanem egyéb ipari, pl. logisztikai alkalmazások is modellezhetőek, mint a jármű-hozzárendelés, vagy a fuvarszervezés. Dolgozatomban bizonyítom, hogy az így generált matematikai modellek garantáltan tartalmazzák a feladat optimális megoldását.

A P-gráf módszertan segítségével egyéb ipari alkalmazások, mint pl. a reakcióút-azonosítás is modellezhető. A dolgozat megmutatja, hogy a reakcióút-azonosításban használt fogalmak, mint a direkt út, az extrém út, vagy a strukturálisan minimális út ekvivalensek. A feladat alternatív megfogalmazásai más-más megoldó módszerekhez vezettek. A jelen dolgozatban bemutatott bizonyítás a feladatok ekvivalenciájáról új utat nyit a megoldó módszerek szinergikus továbbfejlesztésére is.

Abstract

Industrial applications of the P-graph Framework

It is common practice by decision makers to use mathematical programming models in various industrial applications. Several algorithms and solution methods exist which render it possible to solve a given problem without any heuristics. However, even a solution yielded by these exact methods may not be optimal since the initial mathematical model may be poorly constructed.

The P-graph framework was developed by Friedler and his associates in the early nineties for the optimization of complex chemical systems. It is formally proven that the algorithm generated by the framework contains the global optimum of the problem.

In this work it will be shown that besides the chemical systems the P-graph framework can be used to model other industrial applications like vehicle scheduling or vehicle routing. In this work I will prove that the generated mathematical models contain the global optimum of the problem.

With the aid of the P-graph framework it is possible to model the reaction-pathway identification problem as well. This work will show, that the concepts used in reaction-pathway identification, like direct pathway, extreme pathway and structurally minimal pathway are equivalent. The different concepts lead to different solution methods. However, this new result lays the foundations for new solution methods which combine the different concepts.

Abstrakt

Den industriella applikationen av P-graf modellen

Gällande beslutstagande i användningen av den industriella applikationen så är matematiska programmeringsmodeller vanligt förekommande. Det finns ett flertal algoritmer och lösningsmetoder, med vilka problemet i sig exakt kan lösas, utan något behov av heuristiker. Trots detta är resultaten av de exakta metoderna inte nödvändigtvis optimala, för det är möjligt att utgångsmodellen är konstruerad på ett fel sätt.

För att optimalisera komplexa kemiska applikationer, konstruerade Friedler och hans författarkollegor P-graf metoden i början av 90-talet. Det är formellt bevisat att den matematiska programmeringsmodell som genereras av denna metod, garanterat innehåller uppgiftens optimala lösning.

I redovisningen kartlägger jag att P-graf modellen inte enbart behöver användas till att modellera komplexa kemiska system, utan också till att modellera andra applikationer inom logistik, liksom fordonsschemaläggning eller fordonsdirigering. I redovisningen visar jag även att de matematiska modeller som är genererade på detta sätt kommer säkerligen att innehålla uppgiftens optimala lösning.

Med hjälp av P-graf metoden blir det också möjligt att modellera reaktionsvägens problematik. Redovisningen påpekar att de begrepp som används till att identifiera vägen, liksom “direkt väg”, “extrem väg” och “den strukturellt minimala vägen” är motsvarande. De olika idéerna leder till olika lösningsmetoder. Detta nya resultat lägger grunden för nya lösningsmetoder, som förenar de olika idéerna.

Acknowledgement

This is the third time that I have been granted the opportunity to write an acknowledgement but I still find it a fascinating process. I would like to use this opportunity to express my gratitude to everyone who supported me throughout my PhD studies.

First of all, I would like to thank my supervisor, Botond Bertók for his supervision. His previous research results laid the foundations for this work. Without his ideas and guidance this work would have been never accomplished. I would like to express my gratitude to Professor Ferenc Friedler for similar reasons.

I have to admit that I have experienced many ups and downs in the recent years and I was more than once ready to quit. However many of my friends supported me during these periods and they have always motivated me to go on. Here, I would like to thank József Smidla for all the little things he has done during all these years.

My colleagues at Ericsson provided a very supportive environment which was crucial during the preparation of this thesis. They were also very patient with me for which I cannot be thankful enough. Péter Ocsovai is the best team leader I have ever worked with but if I want to be honest I highly doubt that he will read this unless I show him explicitly.

Most importantly, none of this could have happened without my family. Words cannot express how grateful I am to my mother for all of the sacrifices that she made on my behalf.

Chapter 1

Introduction

Mathematical programming models are often used in complex optimization problems to support decision making. The range of applications is very wide, it includes healthcare, logistics, waste management, supply chain design and many more. The website of informs [1] has a nice collection of success stories and Table 1.1 highlights some of them. It is clear that often enormous savings were achieved through optimization.

However, these models are often constructed based on intuition and earlier experience. If the initial model is not constructed properly, even a complete enumeration may miss the optimal solution since it may not be embedded in the model.

The P-graph framework was introduced more than two decades ago to solve complex optimization problems arising in chemical and allied industries. The framework is able to generate the mathematical model directly and systematically from the input parameters. Moreover, it is formally proven, that the generated model contains the global optimum of the problem.

The framework has been extended and adapted to solve various other practical problems. In many cases this has been carried out by simply transforming the problems into a process-network synthesis problem and then they were solved by the effective algorithms of the framework.

In this work it will be shown that the P-graph framework can be used to model two common problems arising in the transportation industry, vehicle scheduling and vehicle routing.

Table 1.1: Success stories of Operations Research

Organization	Problem	Achievement
Texas Children's Hospital	Minimizing Financial Risk	Profitable revenue increase of up to $17 million
U.S. Army Recruiting	Reinventing U.S. Army Recruiting	Savings of $204 million from a $1 billion program
Compañía Sud Americana de Vapores	Empty Container Logistics Optimization	Empty container inventory stocks were reduced by 50%, CSAV saved $56 million
Waste Management Inc.	Improve Route Efficiency	Savings of $52 million
Eli Lilly & Co.	Identifying and Neutralizing the Cause of Deadly Side Effects of Anti-cancer Drug	N/A - value of human life is immeasurable
British Telecommunications	Dynamic Workforce Scheduling	Savings of $150 million in a year on operational costs
Motorola	Optimizing Parts Purchasing Processes	During the first 18 months of the implementation, Motorola saved $600 million
IBM Microelectronics	Matching Assets to Supply Chain Demand	Asset utilization improvements of $80 million

The P-graph framework was used earlier to solve the reaction-pathway identification problem. This work expands these earlier results and it will show that the different concepts used in reaction-pathway identification, like direct path, extreme path and structurally minimal path are equivalent.

1.1 Aim

The main aim of this thesis is to utilize the P-graph framework to find the optimal solution of industrial applications. Vehicle scheduling and vehicle routing are two common problems often arising in the transportation industry. These problems are often solved by mathematical programming but the quality of the solution yielded by these models depends heavily on the quality of the model. If the initial model is not constructed properly then even an exact solution method may miss the optimum. Since it is formally proven that mathematical programming model generated by the P-graph framework contains the global optimum of the problem it is subservient to use the framework to find the optimum of these logistic problems.

- The P-graph framework should be adapted to vehicle scheduling problems. The easiest way is perhaps the transformation of the vehicle scheduling problem into a process-network synthesis problem. Thus, a P-graph model should be designed which properly expresses the unique characteristics of the vehicle scheduling problem. Furthermore, an algorithm should be designed, which systematically generates this model from the vehicle scheduling problem. The optimal solution of the problem is embedded in this model and it could be solved by the effective algorithms of the P-graph framework or by any general mathematical solver.

- The P-graph framework should be adapted to vehicle routing problems, specifically for the capacitated vehicle routing problem. Essentially, a P-graph model should be designed which properly expresses the unique characteristics of the vehicle scheduling problem. Furthermore, an algorithm should be designed which systematically generates this model from the vehicle scheduling problem. The optimal solution of the problem is embedded in this model and it could be solved by the effective algorithms of the P-graph framework or by any general mathematical solver. The procedure involves the conversion of the vehicle routing problem into a process-network synthesis problem.

Another practical problem often arising in applications related to chemical systems is the so-called reaction-pathway identification. The P-graph framework has been adapted earlier to solve this problem, which slightly differs from the typical optimization problems. Here, those earlier results will be extended.

The structurally minimal pathway is a central concept in the P-graph model of the reaction-pathway identification problem. There are several similar concepts in the literature and the aim here is to investigate the exact nature of these relationships.

- The exact relationship between the concept of the structurally minimal pathway and the concept of the direct path should be determined.
- The exact relationship between the concept of the structurally minimal pathway and the concept of the extreme pathway should be determined.

1.2 Emphasizing my own results

The major part of this thesis is written in passive voice. My own results are emphasized unambigously at the end of each chapter in the respective summary. These results are collected and conceived again at the end of the thesis. Whenever I used a previous result of someone else I made a citation and indicated the source in the bibliography.

1.3 Notations

This thesis covers many different topics with many different notations. However, the following guideline is generally true unless explicitly stated otherwise.

- A calligraphic letter or a sequence of calligraphic letters always denotes a set. For example $\mathcal{P}$, $\mathcal{R}$ $\mathcal{O}$, $\mathcal{m}$, $\mathcal{o}$, $\mathcal{solutions}$, $\mathcal{subproblems}$ all denotes sets.

- An italic lowercase letter denotes an integer or a real variable. For example x, y, i, j, k all denote variables. However, an italic lowercase latter followed by brackets denotes a function, for example $f^R(\mathbf{m})$, $l_s(T_i)$, $t_e(T_i)$, etc.

- An italic uppercase letter denotes an identifier which is used to denote materials or operating units. For example A, B, C, EUR, CO_2.

- Italic greek letters denote structural mappings in the P-graph framework, like φ, ψ, ν, ω, φ^+, ψ^- but there are also exceptions, like α, β which denote sets. In latter chapters greek letters may denote stoichiometric coefficients, like λ, γ. The meaning can be always determined from the context unambigously.

- A boldface lowercase letter denotes a vector, for example $\mathbf{x}, \mathbf{y}, \boldsymbol{\lambda}$. A boldface uppercase letter denotes a matrix for example $\mathbf{A}, \mathbf{D}$.

Chapter 2

Problem definitions

This chapter introduces the four main topics of the thesis, all of them related to combinatorial optimization. All of these problems are of utmost practical importance and extremely challenging. Process-network synthesis and reaction-pathway identification are ubiquitous in chemical and allied industries while vehicle routing and vehicle scheduling are everyday problems in distribution and logistics.

2.1 Process-network synthesis

Essentially, every product of the chemical and allied industries are manufactured by process networks. Thus, such problems are ubiquitous in chemical and allied industries and they have enormous practical importance.

2.1.1 General introduction to process-network synthesis

In a process network, raw materials are consumed through various chemical, physical and biological transformations to yield desired products. The equipment or functional units accomplishing these transformations are termed operating units. Hence, a process network can be viewed as a system of operating units, each operating unit transforming a specified number of input materials with known quality into a specified number of output materials by altering their physical, chemical, or biological properties.

Given a process system, the aim of process-network synthesis in short is to determine the optimal structure of the system and to identify the optimal types, configurations and capacites of the functional units performing various operations within the system [99, 132]. The role of process synthesis is to enhance profitability through the reduction of material and energy consumption. It is well documented [120], that effective process synthesis can reduce the energy consumption by 50% and net-present cost by 35%.

A process-network or process synthesis problem is defined by the available raw materials, candidate operating units and desired products. Various other parameters for the operating units and materials are also given. These parameters include the coefficients for the functions expressing the costs of operating units depending on their load, and upper bounds on their respective capacities. The relations between the materials and operating units, i.e., the consumption rates of input and production rates of output materials by the operating units are also defined in the problem specification. The aim is to determine the optimal network where the objective can be either cost minimization or profit maximization.

2.1.2 Complexity of process-network synthesis

To determine the optimal network, the structure of the entire process and the configurations of the operating units constituting the process should be synthesized simultaneously because their performances influence each other. However, this is almost impossible to do so, since the problem is extremely complex and computationally very demanding. This follows from the dual nature of the problem, which is both continuous and discrete at the same time and the latter leads to profound combinatorial complexity.

The present work focuses solely on the so-called macroscopic phase which is often termed in itself as process-network synthesis. This phase involves the determination whether a candidate operating unit is present in the optimal structure of the process or not and consequently the existence or absence of connections between the candidate operating units.

Note, that the complexity of the problem magnifies exponentially due to its combinatorial nature, since the optimal network has to be found among $2^n - 1$ possible alternative networks where n denotes the number of candidate operating units. For instance, when n is 35, $2^n - 1$ yields 34.36×10^9 which is a huge number, and it essentially doubles, when n is increased to 36.

It is easy to see intuitively that process-network synthesis problems are really challenging. It has been also proven formally, that the process-network synthesis problem is equivalent to the set covering problem [69]. Since the set covering problem is one of Karp's original 21 NP-complete problems [73], it is a direct corollary of the theorem that the process-network synthesis problem is NP-complete.

2.1.3 Popular methods for process-network synthesis

Process-network synthesis has been studied extensively in the last few decades. As a result, numerous papers have been published and a huge number of methods have been devised to solve synthesis problems.

Exact methods

The use of mathematical programming is very popular, see e.g. [3, 28, 53, 54, 55, 56, 57, 72, 74, 78, 80, 109, 119]. The inclusion and the exclusion of the candidate operating units in the resultant models are usually expressed by integer variables. Note, that the value of the objective function is often affected more drastically by the integer than the continuous variables, i.e., the combinatorial part of the problem.

The generated models can be solved even by general purpose mathematical programming solvers. This is one of the greatest advantage of these approaches, since they can benefit from the rich tools of mathematical programming. As a result of algorithmic developments and inclusion of achievements of computer science and software engineering, the general purpose mathematical programming solvers available today are very capable even on a desktop PC [91].

Methods based on heuristic

However, practical process-network synthesis problems are often such complex that the resulting mathematical models cannot be solved in reasonable time. Thus, heuristic methods are also used frequently to determine near optimal networks in process synthesis, like

- genetic algorithms [4, 39, 84, 131]
- simulated annealing [27]
- tabu search [87]
- memetic algorithms [126]
- many other heuristic methods [5, 29, 96].

Other methods

Some of the methods combine the two above approaches, resulting in so called hybrid methods. Note, that this is only a small fraction of the papers published in the topic. Since process-network synthesis is essentially an optimization problem, any method designed to solve optimization problems is able to solve process-network synthesis problems as well (e.g. dynamic programming [40]).

2.2 Vehicle scheduling

Vehicle scheduling often arises in industrial applications, perhaps most frequently in public transportation. The scheduling of vehicles has become an extensively studied topic in the last four decades. Numerous exact and heuristical methods and algorithms have been published along with some well-known survey papers, see e.g. [19].

2.2.1 General introduction to vehicle scheduling

Vehicle scheduling begins at the strategic level with collecting or forecasting data of customer demand. In the following step, based on the demand, the infrastructure of the transportation network is defined. Then routes and stop points for different lines are established on the defined infrastructure. After that, particular trips are defined for given lines. The timetable specifies for each trip a departure and an arrival time as well as start and end stations. The further planning focuses on the efficient use of the resources.

Given a set of timetabled trips with fixed travel (departure and arrival) times and start and end locations as well as traveling times between all pairs of end stations, the objective is to find an assignment of trips to vehicles such that each trip is covered exactly once, each vehicle performs a feasible sequence of trips and the overall costs are minimized [19].

2.2.2 Variants, modifications, extensions

The problem has many variants and extensions. For example, the operational costs can be divided into fix (like investment) and operational (e.g. fuel) costs. In practice, operational costs are often expressed in various different ways and thus many applications leave the minimization of operational cost as a secondary objective. Of course, the problem can be extended with several additional requirements like

- **The existence of multiple depots.** The existence of an additional depot extends the problem to NP-hard complexity [77, 110].

- **A heterogeneous fleet.** A fleet consisting of multiple vehicle types. In this case special vehicle types have to be used for service and it is known to be NP-hard even without multiple depots. If the trips are restricted such that they could be serviced only by a subset of the vehicle types then a further extension, called vehicle type groups is inserted [65].

- **Time windows.** In this case variable trip departure and arrival times are considered. Even the simplest time windowed vehicle scheduling problem is NP-complete, since in case of time windows a vehicle scheduling problem with one vehicle and one depot is a traveling salesman problem with time windows. Both discrete and continous time windows are possible [26, 76].

- **Route constraints.** It is possible to force a special property on the routes of a feasible schedule. Typical route constraints are time restrictions on the vehicle routes. These are considered, for example, for fuel restrictions or maintenance intervals [59].

- **etc.**

2.2.3 Complexity of vehicle scheduling

If the vehicle scheduling problem has a single depot then it is relatively easy to solve in the sense that it could be formulated as a problem for which polynomial time algorithms exist. Models for solving the single depot case include the minimal decomposition model, the assignment model, the transportation model and the network flow model.

However, different variants, like the existence of multiple depots extend the problem to NP-hard complexity [15, 86]. In this case different locations for starting routes are possible. In addition, it is a frequent constraint that a vehicle has to return to its start depot at the end of its route. The multiple depot case is often modeled with single and multi-commodity models and set partitioning models. Note, that these models are also applicable to vehicle routing problems.

2.2.4 Popular models for vehicle scheduling

This section introduces the most popular methods, each of them with a short description. They are grouped based on the number of depots defined in the problem.

Single depot case

The VSP for a single depot is relatively "easy" to solve in the sense that it could be formulated as a problem for which polynomial time algorithms are known [19, 41, 101]. Perhaps the most popular models are the following:

- **Minimal decomposition models.** These models typically do not respect operational costs. Another drawback of these kind of models that no upper bound can be set on the fleet size.

- **Assignment models.** These models handle operational costs and formulates the problem as an assignment problem. Can be represented by bipartite graphs.
- **Transportation models.** In the literature some transportation models are often called quasi-assignment models and specialized assignment algorithms have been adapted to this problem. Can be represented by bipartite graphs.
- **Network flow models.** In case of network flow models the solution can be obtained by solving a specific minimum cost flow problem.

Multiple depot case

In this case, different depots for starting bus routes are possible. As an additional restriction each vehicle has to return to its start depot at the end of its route. The multiple depot case is proven to be NP-hard [35, 58].

- **Single-commodity models.** The problem is modeled by a graph where the nodes represent the trips, depots and vehicles. The objective is to find a minimum cost set of elementary circuits, such that each node is covered by exactly one circuit, each circuit has exactly one depot/vehicle node and the number of circuits with a node belonging to a depot never exceeds the capacity of that depot [20].
- **Multi-commodity models.** In general, the multi-commodity models can be considered as the extensions of the network flow models of the single depot vehicle scheduling problems. There is an independent network for each depot and the multi-commodity models are based on the combination of these networks (which is a multigraph) [94].
- **Set partitioning models.** The main idea behind the set partitioning models is to enumerate all feasible routes for the vehicles and then choose a proper subset of these routes that satisfies all constraints. In contrast with other formulations

duty related constraints (like time or fuel restrictions) can be easily recognized [17].

The quality of the LP is an important factor for any solution method in obtaining the optimal integer solution. The single-commodity models provide weaker LP-bound than the connection-based multi-commodity flow formulation which has the same bound quality as the set partitioning model.

2.3 Vehicle routing

Since the transportation process involves all stages of the production and distribution systems and represents a relevant component (generally from 10% to 20%) of the final cost of the goods, the vehicle routing problem is an extremely important practical problem. The large number of real-world applications, both in North America and in Europe, have widely shown that the use of computerized procedures for the distribution process planning produces substantial savings (generally from 5% to 20%) in the global transportation costs. It is easy to see that the impact of these savings on the global economic system is significant [125].

2.3.1 General introduction to vehicle routing

The Vehicle Routing Problem calls for the determination of the optimal set of routes to be performed by a fleet of vehicles to serve a given set of customers, and it is one of the most important and studied combinatorial optimization problems. In most of the cases it is assumed that some goods must be distributed between depots and customers.

The distribution of goods is carried out by a set of vehicles, which are located in one or more depots and perform their movements by an appropriate road network. To solve the vehicle routing problem it is necessary to determine a set of routes, each performed by a single vehicle that starts and ends at its own depot, such that all customer demand is satisfied with respect to the operational constraints and the global transportation cost is minimized.

The road network is usually represented by a directed graph. The arcs of the graph represent the road sections while the nodes of the graph may represent depots, customers or road junctions. If every arc of the graph can be traversed in both directions then it is possible to represent the road network with an undirected graph. There is a cost associated with each arc which denotes the length or the travel time of the given road section. Note, that this parameter may depend on the vehicle type or on the period during the road section is traversed. Unlike the road network which is more or less the same in each application, the characteristics of the customers and the vehicle fleet are often determined by the given application.

2.3.2 Variants, modifications, extensions

There are many different variants of the vehicle routing problem and nearly all of them based on some real life application. In many cases it is assumed that some goods must be distributed between depots and customers. However, typical applications may also include waste collection, school bus routing, dial-a-ride systems and many more. Most of the applications can be modeled by the following variants of the vehicle routing problem:

- **Capacitated vehicle routing.** The capacity refers to the capacity of the cargo hold of vehicles. This is the simplest and most studied variant of vehicle routing.

- **Distance constrained vehicle routing.** A global constraint imposed on the fleet, usually on the distance or on the time.

- **Vehicle routing with time windows.** Each customer must be served in an associated time window.

- **Vehicle routing with backhauls.** The customer set is partitioned into two subsets. The elements of the first subset are the linehaul customers, each of them requiring a given quantity of product to be delivered. The second subset contains the backhaul customers, where a given quantity of inbound product must be picked up. All linehaul customers must be served before any backhaul customer is served.

- **Vehicle routing with pickup and delivery.** A set of transportation requests have to be satisfied. Each request is defined by a pickup point, the corresponding delivery point and a demand to be transported between these locations.

2.3.3 Complexity of vehicle routing

It is easy to see that the vehicle routing problem is a generalization of the well-known travelling salesman problem which is known to be NP-hard (Note that the travelling salesman problem is also a generalization, namely of the Hamiltionian cycle problem which is one of Karp's original 21 problems [73]). As the different variants of the problem become more and more convoluted, the complexity of the problem increases as well. However, this does not mean, that an algorithm, that can solve a complicated variant (like vehicle routing with time windows) performs well on other "simpler" (like capacitated vehicle routing) variants. On the contrary, practice shows that different variants need different solution methods and algorithms.

2.3.4 Popular methods for vehicle routing

Due to its practical importance, vehicle routing is an extensively studied branch of operations research. Exact and heuristic methods are also popular, for comprehensive surveys see e.g. [30, 52, 82, 105, 125].

Exact methods

A typical vehicle routing problem can be formulated as an integer programming problem or a mixed-integer programming problem, see e.g. [7, 22, 25, 48, 83, 81]. The proposed models take into account more and more characteristics of the real-world applications and the developed algorithms and implementations can find good solutions for even industrial size problems in reasonable time.

Methods based on heuristic

Real-life applications are often so convoluted that they cannot be solved by exact methods. Thus, heuristic methods are also very popular and frequently used to solve vehicle routing problems [51]. These methods include

- genetic algorithms [6, 108]
- tabu search [50, 124]
- ant-colony optimization [11, 134]
- simulated annealing [21, 100]
- many other heuristic methods [24, 106].

2.4 Reaction-pathway identification

Reaction-pathway identification plays a key role in the study of the kinetics of mechanisms of chemical or biochemical reactions [16]. For example, it is essential for understanding the effects of external conditions on the rates of catalytic reactions. Such understanding can be a basis for various applications, especially for designing novel industrial chemical processes.

2.4.1 General introduction to reaction-pathway identification

Any chemical reaction consists of elementary reactions. Given the reactants and the final products of the overall chemical reaction, the aim of reaction-pathway identification is to determine the most likely pathway i.e., interconnection between the elementary reactions that yield the given final products from the precursors. Although the knowledge of the rate, reversibility, equilibrium and the extent of the reaction facilitates the ultimate identification of the definite mechanism, the reaction pathway in itself contains no information about these parameters. Thus, the determination of a reaction pathway or mechanism consists of two phases for any given overall reaction.

In the first phase all of the feasible candidate mechanisms are identified and then in the second phase the ultimate pathway or mechanism is selected from the candidate mechanisms identified in the first phase. Those who are engaged in reaction-pathway identification often deal with both phases. Both of these phases have their own unique tasks and difficulties.

The first phase of reaction-pathway identification

Every reaction pathway leading from the precursors to the final products of the reaction is a network of the elementary reactions constituting the pathway. In such a network, each elementary reaction could contribute to the forward, reverse or no step to the network. These three possibilites result in 59048, i.e., $(3^{10} - 1)$ possible combinations even if the network consists of only ten elementary reactions. This may result in hundreds of plausible networks from which the feasible candidate pathways are to be identified.

Complexity

It is easy to see that this is indeed a mathematically and computationally challenging problem. Furthermore, due to the relationship (which will be detailed later) between the reaction-pathway identification problem and the process-network synthesis problem, the complexity of the reaction-pathway identification problem is NP-hard [16]. Due to the above, it is not a big surprise that the first phase of reaction-pathway identification has drawn the attraction of relatively few researchers.

The second phase of reaction-pathway identification

The second phase is esentially based on the advancement of modern precision sensors, high-speed computing methods and devices which resulted in the more accurate measurements of the experimental parameters, efficient spectroscopic determination, speedy simulation of mechanistic reaction-rate equations, reliable molecular dynamic and quantum mechanical calculations, and robust multi-steady state or stability analysis. The resultant information from these experiments and observations is collected in huge knowledge and data bases.

In the second phase, the candidate pathways or mechanisms are virtually selected from these knowledge and data bases by the researchers. Note, that in the light of new experimental and computational results, such pathways and mechanisms could be modified.

Interaction between the two phases of reaction-pathway identification

The latter approach has the definite possibility that a valid pathway or mechanism may be easily overlooked. It is hard to eliminate this possibility on the basis of experimental or computational results. To overcome this problem, all the valid candidate mechanisms should be rigorously identified in the first phase. Thus, the two phases of reaction-pathway identification should be executed successively and iteratively. Moreover, the phases should not only undertaken sequentially, but interactively as well.

It is not uncommon that the input elementary reactions to the first phase are extracted from the data and knowledge bases generated by the expirements and observations in the second phase. The second phase could detect a previously unknown active species contributing to the reaction under consideration which could result in the addition of one or more elementary reaction or reactions to the first phase. Thus, the synthesis of feasible candidate networks from all plausible elementary reactions is the most rational approach for accomplishing the first phase of reaction-pathway identification.

2.4.2 Existing methods

The first phase of reaction-pathway identification is solved by mathematical methods. These methods have different theoretical backgrounds, like

- **Linear algebra.** The earliest methods were based on linear algebra [115, 116] and this approach is still popular even today [123]. While these methods based on linear algebra, they do not use linear algebraic bases to characterize a system of chemical reactions but some more refined concepts, like direct paths [64, 117] or simplexes [122].

- **Convex analysis.** Convex analysis became very popular around 2000 when it was extensively used for the analysis of metabolic networks [102, 112]. Perhaps elementary flux modes [103, 114] and extreme pathways [104, 111, 113] are the best known approaches.

- **Graph theory.** While these methods are heavily based on graph theory [36, 37, 38], they also incorporate the elements of linear algebra and [31, 32] mathematical programming [85, 89].

- **Other methods.** Methods based on other theoretical backgrounds have been also used to accomplish the first phase, see e.g., [66, 92, 93]

Scientists engaged in the second phase usually have a background related to catalysis, biochemistry or combustion science. Due to the industrial and practical significance of the subject matter, their number is vast and increasing, see e.g. [18, 33, 34, 61, 62, 127, 128].

Chapter 3

The P-graph framework

This chapter overviews the main concepts and ideas behind the P-graph framework. The framework was introduced in the early nineties to provide a mathematically rigorous approach for complex optimization problems arising in chemical and allied industries. Nowadays, as a result of nearly two decades of continous development it is extended to solve various other combinatorial optimization problems often arising in industrial applications.

3.1 Motivation

As it has been shown in section 2.1, mathematical programming methods are frequently used to solve process-network synthesis problems. Moreover, these methods are considered to be exact in contrast to heuristic methods. A typical method applying mathematical programming usually has the following steps:

1. Generate the mathematical model from the problem definition and the initial data.

2. Solve the generated model by a solver.

3. Determine the optimal flowsheet from the solution of the mathematical programming model.

Note, that while the above procedure is considered to be exact, it does not exclude heuristics entirely. The first step, the generation of the mathematical model is often guided by intuition. While a heuristic method generates structures that can be deduced from its rules, a mathematical programming method determines structures embedded in the "superstructure". Potentially optimal structures might be excluded from consideration, if the heuristic rules or the initial superstructure is not sufficiently complete.

This was demonstrated in [79]. For the extremely simple case, for the production of pure products from one feed stream by simple and sharp separators, it has been always assumed, but never rigorously proved, that the optimal structure of a separation network cannot contain loops. While the assumption has probably been satisfied for such a simple case, it may be invalid for some separation problems of practical importance, such as the case of multiple feeds or multicomponent products. [79] described a numerical example where the optimal structure of a separation system contained a loop which demonstrated that recycling should be allowed in the minimum cost separation system.

The P-graph framework provides an in-depth study of the fundamental mathematical properties of the superstructure and introduces the concept of "maximal structure" which is analogous to the conventionally used term superstructure. The framework directly and systematically, i.e., algorithmically constitutes the objective function and the set of constraints from the input parameters. Based on mathematically rigorous definitions and theorems, it is formally proven that the global optimum is included in the generated mathematical model. Moreover, the framework judiciously exploits the structural features of the process to be synthesized which gives rise to efficient algorithms which are readily implementable on computers.

3.2 Combinatorial foundations

In process-network synthesis, a material is uniquely defined by its composition, i.e., by its components and their concentrations. The exact description may vary depending on the desired level of abstraction and precision.

Operating units are the functional units in a process network performing various operations. Any operating unit can be specified by the set of its input materials and the set of its output materials. If an operating unit is denoted by a pair (α,β), then α is the set of inputs to and β is the set of outputs from the operating unit. An operating unit is not defined unless its input and output materials are specified.

Let $\mathcal{M}$ be a given finite nonempty set of all materials to be taken into consideration in the process synthesis.

Definition 3.2.1 (Synthesis problem) *A synthesis problem is defined by the triplet* $(\mathcal{P}, \mathcal{R}, \mathcal{O})$*, where*

$$\mathcal{P}(\subset \mathcal{M})$$

is a set of final products,

$$\mathcal{R}(\subset \mathcal{M}, \mathcal{P} \cap \mathcal{R} \neq \emptyset)$$

is a set of raw materials, and

$$\mathcal{O}(\subseteq \wp(\mathcal{M}) \times \wp(\mathcal{M}))$$

is a set of operating units (and $\wp(\mathcal{M})$ *denotes the power set of the materials).*

The relationships among the sets $\mathcal{P}, \mathcal{R}, \mathcal{O}$ appears to be very trivial from the standpoint of chemical engineering. Still, it is essential to ensure mathematical rigor and to the flawless execution of the computer algorithms for process-network synthesis.

3.3 The P-graph representation

Conventional graphs are suitable for analyzing a process structure, however, such graphs are incapable of uniquely representing process structures in synthesis. The examples in [44] demonstrate that the two most commonly used graphs in chemical engineering, digraphs and signal-flow graphs are semantically not rich enough to faithfully represent a process structure. The former is not sufficient to uniquely represent individual materials and their relationship and the latter is not sufficient to uniquely represent individual operating units and their relationship.

3.3.1 Formal definition

To uniquely characterize a synthesis problem, a more sophisticated graph is required. Thus, [44] proposed a special directed bipartite graph, the process graph (or P-graph in short) to capture not only the syntactic but also the semantic contents of the process structure. A graph is bipartite if its vertices can be partitioned into two disjoint sets and no two vertices in the same set are adjacent. Thus a P-graph is defined formally as follows.

Definition 3.3.1 (P-graph) *Let $\mathscr{m}$ be a finite set and*

$$\mathscr{o} \subseteq \wp(\mathscr{m}) \times \wp(\mathscr{m}). \tag{3.3.1}$$

A P-graph is defined by a pair ($\mathscr{m}$, $\mathscr{o}$) where the vertices of the graph are the elements of

$$\mathscr{m} \cup \mathscr{o} \tag{3.3.2}$$

and the arcs of the graph are the elements of

$$\mathscr{A} = \mathscr{A}_1 \cup \mathscr{A}_2 \tag{3.3.3}$$

with

$$\mathscr{A}_1 = \{(x, y) : y = (\alpha, \beta) \in \mathscr{o} \textit{ and } x \in \alpha\} \tag{3.3.4}$$

and

$$\mathscr{A}_2 = \{(y, x) : y = (\alpha, \beta) \in \mathscr{o} \textit{ and } x \in \beta\}. \tag{3.3.5}$$

Definition 3.3.2 (Structure of a system) *Let $\mathscr{m} \subseteq \mathscr{M}$, $\mathscr{o} \subseteq \mathscr{O}$ and $\mathscr{o} \subseteq \wp(\mathscr{m}) \times \wp(\mathscr{m})$. The structure of this system is defined by P-graph $(\mathscr{m}, \mathscr{o})$.*

3.3.2 Illustrative example

A material in the process is symbolized by a circle, designating an M-type vertex while an operating unit is represented by a horizontal bar, designating an O-type vertex. For further classification see Fig. 3.1.

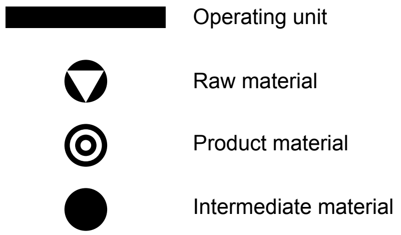


Figure 3.1: P-graph symbols

For illustration, let $\mathscr{P} = \{A\}$, $\mathscr{R} = \{E, G\}$ and $\mathscr{O} = \{(\{B\}, \{A, D\}), (\{C\}, \{A, H\}), (\{D, E\}, \{B\}), (\{E, F\}, \{C\}), (\{G, H\}, \{F\})\}$. The corresponding P-graph is depicted on Fig. 3.2.

Since the elements of o are the pairs of some subsets of m, o satisfies (3.3.1). The vertex set of this graph is $\mathscr{V} = \{A, B, C, D, E, F, G, H, (\{B\}, \{A, D\}), (\{C\}, \{A, H\}), (\{D, E\}, \{B\}), (\{E, F\}, \{C\}), (\{G, H\}, \{F\})\}$. With these notations each arc in $\mathscr{A}_1$ is from an M-type vertex to an O-type vertex (e.g. $(B, (\{B\}, \{A, D\})))$ and in A_2 from an O-type vertex to an M-type vertex (e.g. $((\{D, E\}, \{B\}), B))$.

The fundamental properties of the P-graph are mathematically rigorous and formally proven. More of the basic definitions and many detailed proofs can be found in [44].

3.4 Structural mappings

Both the combinatorial axioms and the combinatorial algorithms manipulating the P-graphs can be formulated more concisely and perspicuously through structural mappings. Given a P-graph (m,o), the following structural mappings are defined.

Mapping $\psi^-(o)$ yields the set of materials of a process structure, each of which is an inlet to at least one operating unit in set o:

$$\psi^-(o) = \bigcup_{(\alpha,\beta)\in o} \alpha \tag{3.4.1}$$

Mapping $\psi^+(o)$ is the pair of mapping $\psi^-(o)$ and thus yields the set of materials each

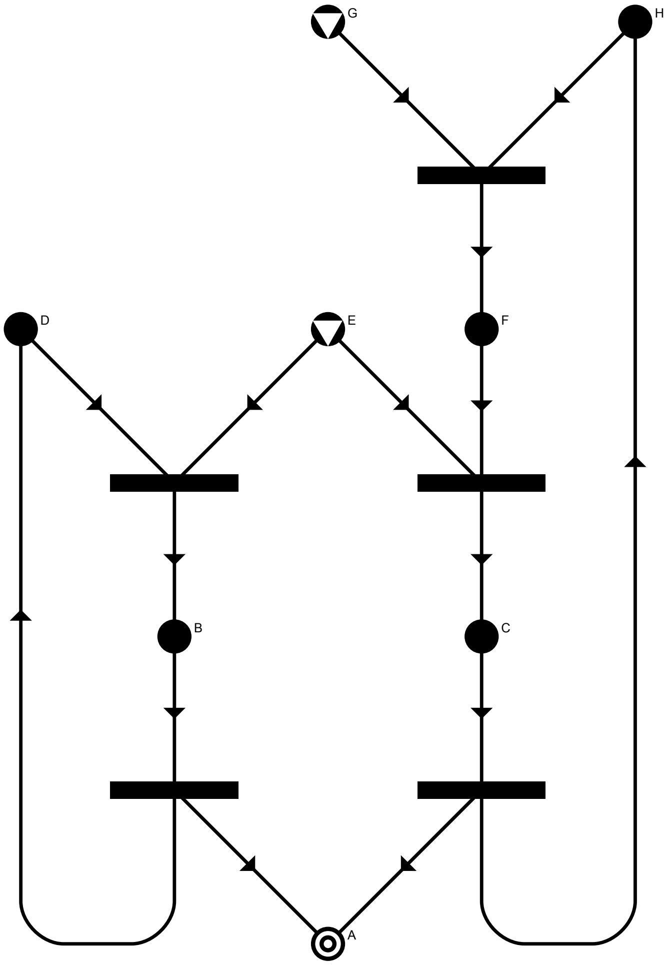


Figure 3.2: A sample P-graph

of which is an outlet from at least one operating unit in set o:

$$\psi^+(o) = \bigcup_{(\alpha,\beta)\in o} \beta \tag{3.4.2}$$

Mapping $\psi(o)$ is the union of mappings $\psi^-(o)$ and $\psi^+(o)$ yielding the set of those materials each of which is either an inlet to or an outlet from at least one operating unit found in set o:

$$\psi(o) = \psi^-(o) \cup \psi^+(o) \tag{3.4.3}$$

Similar structural mappings can be defined on the materials to yield analog sets of operating units. Mapping $\varphi^-(m)$ yields the set of operating units of a process structure, each of which produces at least one material in set m:

$$\varphi^-(m) = \{(\alpha,\beta) \in o : \beta \cap m \neq \emptyset\} \tag{3.4.4}$$

Similarly, $\varphi^+(m)$ yields the set of operating units each of which consumes at least one material in set m:

$$\varphi^+(m) = \{(\alpha,\beta) \in o : \alpha \cap m \neq \emptyset\}. \tag{3.4.5}$$

The union of mappings $\varphi^-(m)$ and $\varphi^+(m)$ is mapping $\varphi(m)$ yielding the set of operating units, each of which either produces or consumes at least one material in set m:

$$\varphi(m) = \varphi^+(m) \cup \varphi^-(m) \tag{3.4.6}$$

Now let m_1 and m_2 be two sets of materials, and o be a set of operating units. Then, relation $\Rightarrow_o$ is defined as follows:

$$m_1 \Rightarrow_o m_2 \text{ if and only if } m_2 = \psi^+(\varphi^+(m_1) \cap o). \tag{3.4.7}$$

Literally, the relation yields the set of materials produced by those operating units in set o that consume at least one material from set m_1. The reflexive, transitive closure of $\Rightarrow_o$ is denoted by $\Rightarrow_o^*$.

3.5 Axioms of combinatorially feasible process structures

Certain combinatorial properties are inherent in all feasible process structures. For example, a structure containing no linkage between a raw material and a final product is unlikely to represent a practical process. The unique features of feasible processes are formulated into a complete mathematical axiom system and thus the search for the optimal structure is restricted to the set of feasible structures.

3.5.1 Formal definition

The following set of axioms express the necessary combinatorial properties to which a P-graph (m, o) must conform to be a combinatorially feasible process structure or solution structure of process-network synthesis problem $(\mathcal{P}, \mathcal{R}, \mathcal{O})$:

Definition 3.5.1 (Solution-structure) *P-graph (m, o) is a solution-structure of synthesis problem $(\mathcal{P}, \mathcal{R}, \mathcal{O})$ if it satisfies the following axioms:*

(S1) Every final product is represented in the graph: $\mathcal{P} \subseteq m$.

(S2) A vertex of the material type has no input if and only if it represents a raw material: $(m \setminus \psi^+(o) = m \cap \mathcal{R})$.

(S3) Every vertex of the operating unit type represents an operating unit defined in this synthesis problem: $(o \subseteq \mathcal{O})$.

(S4) Every vertex of the operating unit type has at least one path leading to a vertex of the material type representing a final product: $\forall o \in o$ there exists a set m' such that $\psi^+(\{o\}) \Rightarrow_o^* m'$ and $m' \cap \mathcal{P} \neq \emptyset$.

(S5) If a vertex of the material-type belongs to the graph, it must represent an input to or an output from at least one operating unit represented in the graph: $m \subseteq \psi(o)$.

In other words, Axiom (S1) ensures that each product is produced by at least one operating unit in the process system; Axiom (S2) implies that no raw material should be generated by the process system under consideration; through Axiom (S3) only the plausible operating units are taken into account during synthesis; Axiom (S4) disallowes the existence of any operating unit not contributing to the generation of any product and finally Axiom (S5) ensures that only those materials that belong to at least one operating unit in the structure can belong to the structure.

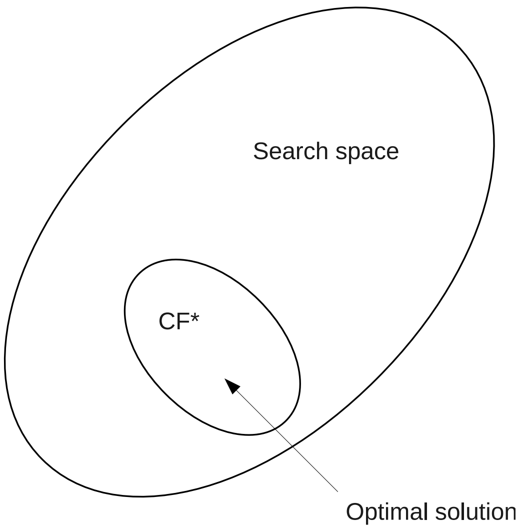
P-graph (m, o) is defined to be a combinatorial feasible or solution structure for a synthesis problem $(\mathcal{P}, \mathcal{R}, \mathcal{O})$ if and only if it satisfies Axioms (S1) through (S5).

3.5.2 Utilization

This axiom set acts as a filter to eliminate all combinatorially infeasible or invalid networks. In a super-structure generated by a traditional method the number of infeasible networks can be frequently vast even when the super-structure is completely and correctly constructed. For example, the commercial size process in [43, 47] has 35 operating units which gives rise to a complete super-structure containing (2^{35}-1) or roughly 34.36 billion possible networks.

None of the traditional methods of process-network synthesis allows the exclusion of the overwhelming majority of infeasible networks a priori. To find the optimal structure among the 34.36 billion possible networks a conventional mathematical programming has to search all of them. At this magnitude, if the efficiency of a conventional method is increased by 100%, it is the same as if the search space is reduced from 34.36 billion to 17.18 billion possible networks which is still an enormous number.

In contrast to the above, with the five axioms it is possible to reduce the search space to the combinatorially feasible solution structures. (See Fig. 3.3) In case of the example cited from [43, 47] this means that it is sufficient to consider 3465 combinatorially feasible networks instead of 34.36 billion possible networks. Note that the search space is reduced by roughly 99.99999% percent.



CF:Combinatorially Feasible Structures

Figure 3.3: Reduction of the search space

3.5.3 Maximal structure

Let $\mathcal{S}(\mathcal{P},\mathcal{R},\mathcal{O})$ be the set of all solution-structures for process-synthesis problem $(\mathcal{P},\mathcal{R},\mathcal{O})$. One of the basic properties of $\mathcal{S}(\mathcal{P},\mathcal{R},\mathcal{O})$ is that it is closed under union, i.e., the union of two solution-structures remain a solution structure. Formally, if:

$$\sigma_1 \in \mathcal{S}(\mathcal{P},\mathcal{R},\mathcal{O}) \text{ and } \sigma_2 \in \mathcal{S}(\mathcal{P},\mathcal{R},\mathcal{O}) \tag{3.5.1}$$

then

$$(\sigma_1 \cup \sigma_2) \in \mathcal{S}(\mathcal{P},\mathcal{R},\mathcal{O}). \tag{3.5.2}$$

Definition 3.5.2 (Maximal structure) *The union of all solution structures, denoted by $\mu(\mathcal{P},\mathcal{R},\mathcal{O})$ is defined to be the maximal structure, i.e.,*

$$\mu(\mathcal{P},\mathcal{R},\mathcal{O}) = \bigcup_{\sigma \in S(\mathcal{P},\mathcal{R},\mathcal{O})} \sigma \tag{3.5.3}$$

In the maximal structure, each arc or vertex belong to at least one solution-structure and each solution-structure is a subgraph of the maximal structure. The maximal structure has been extensively studied in [42] where many of its properties has been formally proven. Since $\mu(\mathcal{P}, \mathcal{R}, \mathcal{O})$ is closed under union, and its cardinality is finite, the maximal structure is a solution structure itself.

The maximal structure of the synthesis problem $(\mathcal{P}, \mathcal{R}, \mathcal{O})$ comprises all the combinatorially feasible structures capable of yielding the specified products from the given raw materials. Thus, the maximal structure contains the optimal process and the mathematical model of a process-network synthesis problem should be based on the maximal structure.

3.6 Combinatorial algorithms

Based on axioms (S1)-(S5) two important combinatorial algorithms can be formulated. Algorithm MSG generates the maximal structure of a given problem while algorithm SSG generates each combinatorially feasible solution structure exactly once.

3.6.1 Algorithm MSG

Algorithm MSG (Maximal Structure Generator) generates the maximal structure of the given $(\mathcal{P}, \mathcal{R}, \mathcal{O})$ process-network synthesis problem in polynomial time. In the reduction part algorithm MSG eliminates the materials and operating units which should not belong to the maximal structure. This is carried out stepwisely starting from the raw materials of the input structure validating that no node violates any of the five axioms. In this phase the elimination of a node often leads to the elimination of other nodes linked to it.

In the construction phase the remaining nodes are linked again stepwisely but this time starting from the products. The linking is done with respect to the axioms. Note, that some of the feasible linkages may be eliminated in the reduction phase and thus they must be reestablished in the construction phase.

Running algorithm MSG to determine the maximal structure of the process-network synthesis problem should be the first step of a solution procedure. Besides generating the mathematical programming model of the problem it can be used to determine whether the problem has a feasible solution, or not. Since the maximal structure is the union of all combinatorially feasible solution structures, the absence of the maximal structure implies that the problem has no feasible solution.

3.6.2 Algorithm SSG

Algorithm SSG (Solution Structure Generator) has been developed to generate the set of combinatorially feasible solution structures which is also of fundamental importance [43]. SSG generates each combinatorially feasible solution structure exactly once.

Algorithm SSG was originally introduced as a recursive algorithm which used decision mappings [47]. While recursion has a certain elegancy in expressing algorithms it is not the most efficient approach in case of a computer implementation. Thus, like many other recursive algorithm, SSG has a nonrecursive formulation, which is implemented in practice.

This nonrecursive formulation of SSG is essentially a branch and bound without a bound function and the enumeration of the combinatorially feasible solutions is the generation of the complete search tree. Note, that while the size of the set of combinatorially feasible solution structures is excessively large to have its elements enumerated in practice, SSG still constitutes a fundamental building block for a mathematical programming approach to process synthesis.

3.7 Parametric model of process-network synthesis

A mathematical programming problem consists of the constraints and the objective function. For a process-network synthesis problem, the constraints comprise the mass balances, the availability of raw materials and the lower bounds on the amounts of products to be manufactured to meet the demand.

3.7.1 Mass balance constraints

For each material m_j a lower bound $Lp(m_j)$ and an upper bound $Up(m_j)$ is defined on the gross production. If material m_j is a product then the lower bound $Lp(m_j)$ on the gross result must be greater than zero; it is defined as zero otherwise.

$$Lp(m_j) = \begin{cases} > 0 & \forall m_j \in \mathcal{P} \\ 0 & \textit{otherwise.} \end{cases} \tag{3.7.1}$$

If m_j is a raw material then the upper bound on the gross result is set to zero; otherwise, it must be greater than, or equal to $Lp(m_j)$:

$$Up(m_j) = \begin{cases} 0 & \forall m_j \in \mathcal{R} \\ \geq Lp(m_j) & \textit{otherwise.} \end{cases} \tag{3.7.2}$$

An upper bound on the gross consumption of materials is also defined; for a material m_j, it is denoted by $Uc(m_j)$. $Uc(m_j)$ must be greater than zero if m_j is a raw material; otherwise it is defined as zero.

$$Uc(m_j) = \begin{cases} > 0 & \forall m_j \in \mathcal{R} \\ 0 & \textit{otherwise.} \end{cases} \tag{3.7.3}$$

Every intermediate material must satisfy the mass balance constraints such that the amount produced from each intermediate material has to be greater than or equal to the amount consumed from that material.

3.7.2 Objective function

The objective function represents the cost of the network, which is to be minimized. The cost of the network consists of the costs of the operating units and raw materials.

The annual cost of an operating unit is considered to be the sum of its operating cost and its annualized investment cost:

$$\text{annual cost} = \text{operating cost} + \frac{\text{investment cost}}{\text{payout period}} \tag{3.7.4}$$

Since the optimization model is expected to provide the optimal loads of operating units beside the optimal process structure, the cost is given as the function of the mass load, e.g., by a linear function with a fixed charge

$$cf(o_i) + cp(o_i)x_i \tag{3.7.5}$$

where $cf(o_i)$ is the fixed charge, $cp(o_i)$ is the proportionality constant, and x_i is the load of the operating unit, which typically varies between 0 and 1, i.e., 0-100%. If both the investment and operating costs are given by functions, then the cost function is the combination of them. The parameters of the linear cost function with fixed charge are the sums of the parameters $cf^{op}(o_i)$ and $cp^{op}(o_i)$ of the operating cost and the parameters $\frac{cf^{inv}(o_i)}{\text{payout period}}$ and $\frac{cp^{inv}(o_i)}{\text{payout period}}$ of the annualized investment cost:

$$cf(o_i) = \frac{cf^{inv}(o_i)}{\text{payout period}} + cf^{op}(o_i) \tag{3.7.6}$$

$$cp(o_i) = \frac{cp^{inv}(o_i)}{\text{payout period}} + cp^{op}(o_i) \tag{3.7.7}$$

The price of raw material $m_j \in \mathcal{R}$ is denoted by $cm(m_j)$.

3.7.3 Further notations

The relation between material m_j and operating unit o_i is expressed by a_{ij} which denotes the difference between the production and consumption rate of m_j by o_i:

$$\alpha_i = \{m_j \in \mathcal{M} : a_{ji} < 0\} \tag{3.7.8}$$

and

$$\beta_i = \{m_j \in \mathcal{M} : a_{ji} > 0\}. \tag{3.7.9}$$

Now denote the set of operating units in the optimal structure by $\mathcal{o}^* \subseteq \mathcal{O}$ and the set of materials by $\mathcal{m}^* \subseteq \mathcal{M}$ where

$$\mathcal{m}^* = \bigcup_{(\alpha_i,\beta_i)\in \mathcal{o}^*} \alpha_i \cup \beta_i. \tag{3.7.10}$$

Furthermore, denote the vector of the optimal loads of the operating units for the problem by $\mathbf{x}^* = [x_1, x_2, ..., x_n]^T$ and the objective value by

$$z^* = \sum_{(\alpha_i,\beta_i)=o_i\in \mathcal{o}^*} (cf(o_i) + x_i^*(cp(o_i) - \sum_{m_j\in\alpha\cup\beta} a_{ij}cm(m_j))). \tag{3.7.11}$$

3.7.4 The complete parametric model

The aim of the parametric process-network synthesis problem is to determine the optimal solution i.e., the network $(\mathscr{m}^*, \mathscr{o}^*, \mathbf{x}^*, z^*)$ satisfying the conditions in (3.7.12)-(3.7.16) where z^* is minimal.

$$\forall o_i = (\alpha_i, \beta_i) \in \mathscr{o}^* : m_j \in \alpha_i \iff a_{ji} < 0, m_j \in \beta_i \iff a_{ji} > 0 \tag{3.7.12}$$

$$\forall m_j \in \mathscr{m}^* \cap \mathscr{R} : -Uc(m_j) \leq \sum_{o_i \in \mathrm{o}^*} a_{ji} x_i^* \leq 0 \tag{3.7.13}$$

$$\forall m_j \in \mathscr{m}^* \cap \mathscr{P} : Lp(m_j) \leq \sum_{o_i \in \mathscr{o}^*} a_{ji} x_i^* \leq Up(m_j) \tag{3.7.14}$$

$$\forall m_j \in \mathscr{m}^* \setminus \mathscr{R} \setminus \mathscr{P} : 0 \leq \sum_{o_i \in \mathscr{o}^*} a_{ji} x_i^* \leq Up(m_j) \tag{3.7.15}$$

$$0 < x_i^* \leq u_i \iff o_i \in \mathscr{o}^* \tag{3.7.16}$$

3.8 The MILP model of process-network synthesis

The P-graph framework algorithmically generates a mixed-integer linear programming model for the process-network synthesis problem. Furthermore, it is mathematically proven that the generated MILP model contains the global optimum. Although this is an integral part of the framework, most of the publications strongly emphasizes the combinatorial innovations of the framework. Thus, the model was published surprisingly late in [10, 13].

3.8.1 Binary variables

In contrast to the combinatorial algorithms, the structural decisions in the MILP formulation are represented by y_i binary variables instead of sets or graphs. $y_i = 1$ represents the inclusion of o_i to a structure, while $y_i = 0$ represents the exclusion of o_i from a structure. y_i's are often called as existence variables in the literature.

Since the load of an operating unit can only differ from 0 if it is included to the structure, this relation have to be expressed by a linear constraint in the MILP model. In the parametric model of the process-network synthesis an upper bound u_i

is defined for the load of each operating unit o_i. u_i is typically equal to 1, except when o_i can have multiple instances or available in multiple capacities in the structure, and these alternatives can be expressed by a single cost function. If o_i is included into the structure, i.e., $y_i = 1$, then the corresponding continuous load variable x_i can be any real value in the range of 0 to u_i. Thus,

$$x_i \leq y_i u_i. \tag{3.8.1}$$

3.8.2 The complete MILP model

The aim is to determine the $(\mathbf{x}^*, \mathbf{y}^*, z^*)$ optimal solution of the MILP problem algorithmically generated from the parametric process-network synthesis problem, satisfying the conditions in (3.8.2) through (3.8.6) where z^* is minimal.

$$\forall m_j \in \mathcal{R} : -Uc(m_j) \leq \sum_{o_i \in O} a_{ji} x_i^* \leq 0 \tag{3.8.2}$$

$$\forall m_j \in \mathcal{P} : Lp(mj) \leq \sum_{o_i \in O} a_{ji} x_i^* \leq Up(m_j) \tag{3.8.3}$$

$$\forall m_j \in \mathcal{M} \setminus \mathcal{R} \setminus \mathcal{P} : 0 \leq \sum_{o_i \in O} a_{ji} x_i^* \leq Up(m_j) \tag{3.8.4}$$

$$0 < x_i^* \leq y_i^* u_i \tag{3.8.5}$$

$$z^* = \sum_{o_i \in O} (y_i^* cf(o_i) + x_i^* (cp(o_i) - \sum_{m_j \in (M)} a_{ij} cm(m_j))) \tag{3.8.6}$$

Eqns (3.8.2)-(3.8.4) are basically the (mass balance) constraints imposed on the raw, product and intermediate materials, respectively. (3.8.5) represents the individual bounds on the capacities of the operating units. (3.8.6) is the cost of the network, which is to be minimized.

3.8.3 Solving the MILP model

Since SSG is a branch and bound without a bound function, it is possible to generate each of the combinatorial feasible structures, solve the corresponding mathematical programming models and then select the one(s) with the minimum objective value.

This is basically a complete enumeration and it is considered to be an ineffective approach, but still, this is a valid option which is actually implemented in the software executing P-graph framework algorithms.

However, the P-graph framework has more effective solution algorithms and perhaps the most notable among them is ABB (Accelerated Branch-and-Bound) [46]. As its name suggest, ABB is based on the famous branch and bound principle. It has been developed specifically for process-network synthesis problems. Using clever enhancements, like neutral extensions, it is capable to solve process-network synthesis problems in a reasonable time. Like each branch and bound algorithm, ABB is well parallelizable, which is a definite advantage considering the computer architectures available today. With parallel branch and bound methods it is possible to achieve super linear speedup for some problem instances [121].

3.9 Relationship between the MILP model and the combinatorial axioms

Perhaps it is worth noting, that each of the combinatorial axioms is reflected by the MILP model:

(S1) Every final product is represented in the graph. This is analog to eqn (3.8.3).

(S2) A vertex of the material type has no input if and only if it represents a raw material. This can be derived from eqn (3.8.2).

(S3) Every vertex of the operating unit type represents an operating unit defined in this synthesis problem. Trivially, eqns (3.8.2)-(3.8.4) consider only those operating units which are defined in the problem.

(S4) Every vertex of the operating unit type has at least one path leading to a vertex of the material type representing a final product. This can be deduced from the mass balance constraints, i.e., from the combination of eqns (3.8.2)-(3.8.4).

(S5) If a vertex of the material-type belongs to the graph, it must represent an input to or an output from at least one operating unit represented in the graph. Like above, this is ensured by the mass balance constraints expressed by eqns (3.8.2)-(3.8.4).

3.10 Extensions and adaptations

In the past decades, the P-graph framework has been successfully extended to solve various different practical problems. For example, the framework has been extended to

- Simultaneously synthesize a process and its fault-tolerant control system [60]
- Handle waste properly giving rise to an optimal process with integrated in-plant waste treatment [45]
- Simultaneously synthesize a process and its heat exchanger network [98]

Note, that the framework is not limited to problems related to chemical and allied industries. It has been adapted to

- Building-evacuation-route planning [49]
- Supply chain management [14]
- Sustainable development [75]
- The combination of the above two [129]

It is also possible to combine the framework with discrete modeling tools, like Petri-nets [130] for the optimization and verification of IT systems. The framework has been also adapted to solve the reaction-pathway identification problem [16]. This is one of the main topics of this thesis, and it is covered in detail in Chapter 6.

3.11 Related publication

Refereed Journal Paper

1. B. Bertok, M. Barany and F. Friedler. Generating and Analyzing Mathematical Programming Models of Conceptual Process Design by P-graph Software, Industrial & Engineering Chemistry Research, 52:166-171, 2013. (IF=2.235)

Chapter 4

Modeling vehicle scheduling problems as process-network synthesis problems

Vehicle scheduling problems arise frequently in industrial applications perhaps most notably in public transportation. Vehicle scheduling is similar to vehicle routing in the sense that in both cases a demand should be satisfied with a fleet of available vehicles. However, a major difference is that in case of vehicle scheduling the routes or more like the trips are given by the problem definition “only” the vehicles must be assigned to them. While vehicle scheduling may seem much simpler than vehicle routing in case of multiple depots the complexity of a VSP becomes NP-hard.

Vehicle scheduling problems are often solved by mathematical programming methods. However, as in the case of PNS the underlying mathematical models may be constructed based on intuition and earlier experience and thus they may not contain the global optimum. Here it will be shown that the VSP can be modeled as a PNS problem by the P-graph framework resulting in a MILP model which contains the global optimum.

4.1 Problem definition

Vehicle scheduling is the main topic of many papers. There are many variations with many different notations but the aim is usually the following [19] : Given a set of timetabled trips with fixed travel (departure and arrival) times and start and end locations as well as traveling times between all pairs of end stations, the objective is to find an assignment of trips to vehicles such that

- Each trip is completed exactly once
- Each vehicle performs a feasible sequence of trips and
- The overall costs are minimized

4.1.1 Notations

Here, the notations proposed in [10] are used for formalization. Thus,

- $\mathcal{T}$ denotes the set of trips to be performed
- $\mathcal{V}$ denotes the set of available vehicles
- $T_i \in \mathcal{T}$ denotes a trip to be performed
- $V_i \in \mathcal{V}$ denotes a vehicle which could perform a tour
- $\mathcal{A}(T_i)$ denotes the set of vehicles which could perform T_i
- $\mathcal{H}(V_i)$ denotes the set of trips completed by V_i
- $t_s(T_i)$ denotes the starting time of a trip T_i
- $t_e(T_i)$ denotes the ending time of a trip T_i
- $l_s(T_i)$ denotes the start location of T_i
- $l_e(T_i)$ denotes the end location of T_i
- $l_d(V_i)$ denotes the depot of V_i

- $c(V_i)$ denotes the cost associated with V_i
- $e(V_i)$ denotes the CO_2 emission of V_i
- $v_{max}(V_i)$ denotes the maximum speed of V_i
- d denotes the distance between a pair of locations

Note that some of the above parameters like the CO_2 emission of V_i are not necessary to formulate a vehicle scheduling problem. However, their incorporation into the model makes it possible handle more sophisticated, more realistic applications. These additional parameters can be simply ignored to formulate a conventional vehicle scheduling problem.

4.1.2 Formal definition

Based on the above notations, the vehicle scheduling problem can be formalized in the following way. Given a set of timetabled trips with fixed travel (departure and arrival) times and start and end locations as well as traveling times between all pairs of end stations, the objective is to find an assignment of trips to vehicles such that

- Each trip is completed exactly once

$$\forall T_i \in \mathcal{T} : |T_i| = 1, \exists \mathcal{H}(V_j), T_i \in \mathcal{H}(V_j), T_i \notin \bigcup_{i=1, i \neq j}^{n} \mathcal{H}(V_i) \tag{4.1.1}$$

- Each vehicle performs a feasible sequence of trips

$$\begin{aligned} \forall V_i : T_i, T_j \in \mathcal{H}(V_i) \Rightarrow V_i \in \mathcal{A}(T_i) \cap A(T_j), t_e(T_i) \le t_s(T_j), \\ \frac{d(l_e(T_i), l_s(T_j))}{t_s(T_j) - t_e(T_i)} \le v_{max}(V_i) \end{aligned} \tag{4.1.2}$$

- The cost is minimal

$$\begin{aligned} & \sum_{V_i \in \mathcal{V}} (\sum_{\mathcal{H}(V_i)} (c(V_i) d(l_s(T_i), l_e(T_i))) \\ + & \sum_{\substack{\mathcal{H}(V_i) \\ min\{t_e(T_i) \le t_s(T_j) : t_s(T_j) - t_e(T_i)\}}} (c(V_i) d(l_e(T_i), l_s(T_j))) \\ + & \quad c(V_i) d(l_d(V_i), min\{T_i \in \mathcal{H}(V_i) : t_s(T_i)\})) \end{aligned} \tag{4.1.3}$$

4.2 Modeling vehicle scheduling as a synthesis problem

This section introduces the concepts of the P-graph model of the vehicle scheduling problem. These concepts are illustrated by an example and then some important properties of the model and the solution yielded by the model are proven.

4.2.1 Analogy between the concepts of vehicle scheduling and process-network synthesis

In a process-network synthesis problem, a network of operating units transforms the given raw materials into the desired products and during this procedure a set of intermediate materials is created. In a vehicle scheduling problem, a set of trips have to be completed by utilizing the available resources and this is carried out by traveling between a set of places.

It is not hard to see the analogy between the concepts of vehicle scheduling and process-network synthesis. The trips can be thought as products, which have to be produced by a process system. The vehicles performing the trips are considered as resources which have to be utilized in order to complete these trips and as such they are analogous to the raw materials of a process system.

In order to complete the trips, the vehicles have to travel between a sequence of places, which is conceptually similar to the sequence of intermediate materials leading to a product in a process system. A travel has a starting location and a destination similarly to an operating unit, which has a set of inputs and a set of outputs. Thus, it is possible to create a rough mapping between the concepts of vehicle scheduling and the concepts of process synthesis (see Table 4.1).

4.2.2 The P-graph model of vehicle scheduling

The P-graph model of vehicle scheduling is based on the above anology. As a consequence, the P-graph model of vehicle scheduling will operate with materials and operating units.

Table 4.1: Mapping the concepts of VSP to PNS

VSP	PNS
trips	products
resources	raw materials
locations	intermediate materials
traveling	operating units

Materials

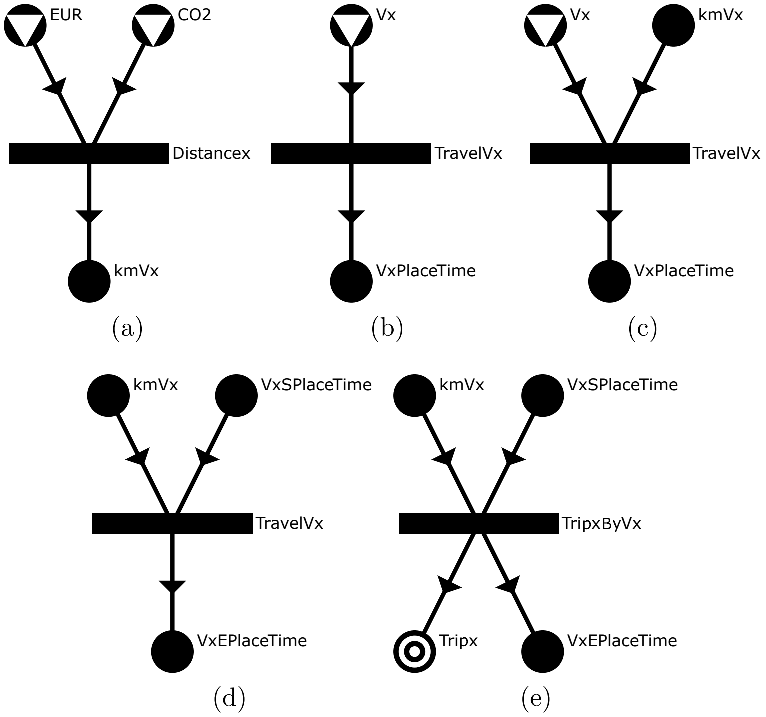
A trip is modeled as a product. The position of a vehicle at a given time is modeled by an intermediate material (e.g. Vehicle1 is in Budapest at 8:00). If a vehicle moves between two locations, it is modeled by an operating unit. A vehicle in its initial position, i.e, in its depot, is modeled by a raw material.

Note that while the CO_2 emission is the byproduct of the traveling, for modeling purposes it is also considered as a resource. This renders it possible to impose an upper bound on the raw material modeling the emission. For similar reasons, the available funds are also considered as a resource and thus it is possible to impose an upper bound on the cost. The cost and the emission are proportional to the completed distance, which is modeled by an intermediate material which is produced from the cost and the emission.

Operating units

The model has three types of operating units. The first type produces an intermediate material from the raw materials representing the cost and the emission. The produced material represents the distance associated with a vehicle V_x. This is interpreted in the following way: each completed kilometer costs a given amount of Euro and during each completed kilometer a given amount of CO_2 is emitted. See Figure 4.1a for illustration.

The second type of operating unit models a movement of a particular vehicle but this movement does not complete a trip. Actually, this covers several use cases. In the simplest case the depot is located at the exact same place where the tour begins.



(a) (b) (c) (d) (e)

Figure 4.1: Operating units used in vehicle scheduling

This is illustrated by Figure 4.1b.

In the second case the vehicle must leave the depot and travel to a place where it can begin a trip. This is illustrated by Figure 4.1c.

In the third case the vehicle travels between two trips, i.e., it has completed one trip but it has to travel to another place to begin the next. This is illustrated by Figure 4.1d.

Finally, the third type of operating unit models a movement of a particular vehicle and this movement completes a trip. This is very similar to the previous case illustrated by Figure 4.1d. See Figure 4.1e for illustration.

The proper combination of these materials and operating units yield the P-graph model of the VSP. In the following, the concept is clarified further by an illustrative example.

Table 4.2: Trips to be completed

$T_i \in \mathcal{T}$	$t_s(T_i)$	$l_s(T_i)$	$t_e(T_i)$	$l_e(T_i)$	$\mathcal{A}(P_i)$
T_1	7:00	Tihany	7:40	Almádi	V_1, V_2
T_2	8:00	Veszprém	9:20	Fehérvár	V_1, V_3
T_3	8:40	Veszprém	9:40	Tihany	V_1, V_2

Table 4.3: The fleet of vehicles

$V_i \in \mathcal{V}$	$l_d(V_i)$	$c(V_i)$	$e(V_i)$	$v_{max}(V_i)$
V_1	Tihany	0.6	375	90
V_2	Fehérvár	0.5	400	90
V_3	Kenese	0.4	300	90

4.2.3 Illustrative example

In this example a predefined set of trips have to be completed by the available vehicles. The properties of the trips are listed in Table 4.2. This table lists the start and end times of a trip, the start and end locations of a trip and the vehicles which could complete the given trip. For example, T_1 starts at 7:00 in Tihany, ends at 7:40 in Almádi and could be completed by V_1 or V_2.

Table 4.3 lists the properties of the available vehicles. These are the location of the depot, the cost, the CO_2 emission and the maximal speed of the vehicle. In this example, the cost is given in Euro, the maximal speed in km/h and the CO_2 emission in g. The cost and emission are proportional to the completed distance. The problem data is visualized by Figure 4.2.

Table 4.2 gives rise to Table 4.4. The trips are represented by product materials and the bounds are set to ensure that each trip is completed exactly once.

Table 4.5 is derived from Table 4.3. There is exactly one from each vehicle and thus the upper bounds on these raw materials are set to 1. The costs of the vehicles are proportional to the covered distance without any associated fix cost. On the other hand the raw materials modeling the cost and the emission are unconstrained.

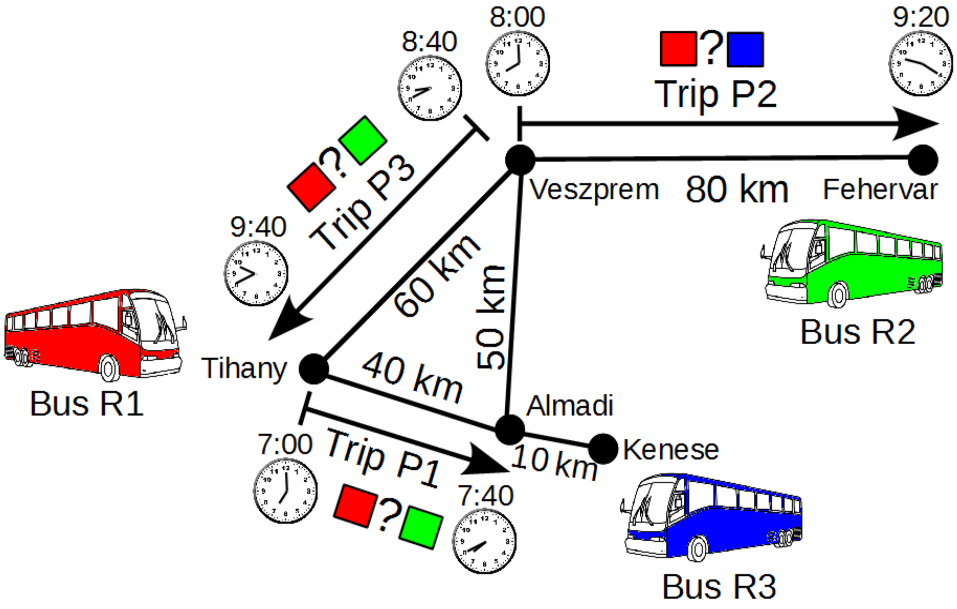


Figure 4.2: The illustrative example

Table 4.4: Products of the synthesis problem

Products ($m_j \in \mathcal{P}$)	Lower bounds ($Lp(m_j)$)	Upper bounds ($Up(m_j)$)
T_1	1	1
T_2	1	1
T_3	1	1

Table 4.5: Raw materials of the synthesis problem

Raw materials ($m_j \in \mathcal{R}$)	Upper bounds ($Uc(m_j)$)	Prices ($cm(m_j)$)
V_1	1	0
V_2	1	0
V_3	1	0
EUR	∞	1
CO_2	∞	0

Note that only the raw material modeling the cost has an associated price. Since the material representing the completed distance is produced from the raw materials representing the cost and the emission it is ensured that the overall cost and emission are proportional to the completed distance.

Table 4.6 lists the operating units involved in the P-graph model of the vehicle scheduling problem. A *Distance* type of operating unit produces the intermediate material representing the completed distances of a vehicle. For example, $Distance_1$ is interpreted in the following way: each completed kilometer by V_1 costs 0.6 EUR and 375 g CO_2.

A $Travel$ type of operating unit models a movement which does not complete a trip. For example, V_2 is initially located in its depot, in Fehérvár. It can complete T_1 or T_3 and thus it must travel to Tihany or Veszprém to complete one of the two trips. This is modeled by $Travel_1$ and $Travel_2$.

$Travel_3$ models the case when the depot is located at the start point of a trip. In this case after the vehicle leaves the depot it can begin the trip immediately. Note that $Travel_4$ and $Travel_5$ yield to two different trips. $Travel_6$ is analog to $Travel_1$ and $Travel_2$. In case of $Travel_7$ and $Travel_8$ the vehicle has already completed a trip and it travels to the start point of another trip.

The rest of the operating units model movements which complete trips. For example, T_1ByV_1 models that V_1 travels from Tihany to Almádi and thus it completes T_1.

The maximal structure of the illustrative example is shown in Figure 4.3. Note that the return of a vehicle to its respective depot could be easily enforced after it

Table 4.6: Operating units of the synthesis problem

$o_i \in \mathcal{O}$	α_i	β_i	$lb(o_i)$	$ub(o_i)$
$Distance_1$	0.6 EUR, 375 CO_2	kmV_1	0	∞
$Distance_2$	0.5 EUR, 400 CO_2	kmV_2	0	∞
$Distance_3$	0.4 EUR, 300 CO_2	kmV_3	0	∞
$Travel_1$	V_2, 80 kmV_2	$V_2Veszprem0840$	0	∞
$Travel_2$	V_2, 140 kmV_2	$V_2Tihany0700$	0	∞
$Travel_3$	V_1	$V_1Tihany0700$	0	∞
$Travel_4$	V_1, 60 kmV_1	$V_1Veszprem0840$	0	∞
$Travel_5$	V_1, 60 kmV_1	$V_1Veszprem0800$	0	∞
$Travel_6$	V_3, 60 kmV_3	$V_3Veszprem0800$	0	∞
$Travel_7$	$V_2Almadi0740$, 50 kmV_2	$V_2Veszprem0840$	0	∞
$Travel_8$	$V_1Almadi0740$, 50 kmV_1	$V_1Veszprem0840$	0	∞
T_1ByV_1	$V_1Tihany0700$, 40 kmV_1	$V_1Almadi0740$, T_1	1	1
T_1ByV_2	$V_2Tihany0700$, 40 kmV_2	$V_2Almadi0740$, T_1	1	1
T_2ByV_1	$V_1Veszprem0800$, 80 kmV_1	$V_1Fehervar0920$, T_2	1	1
T_2ByV_3	$V_3Veszprem0800$, 80 kmV_3	$V_3Fehervar0920$, T_2	1	1
T_3ByV_1	$V_1Veszprem0840$, 60 kmV_1	$V_1Tihany0940$, T_3	1	1
T_3ByV_2	$V_2Veszprem0840$, 60 kmV_2	$V_2Tihany0940$, T_3	1	1

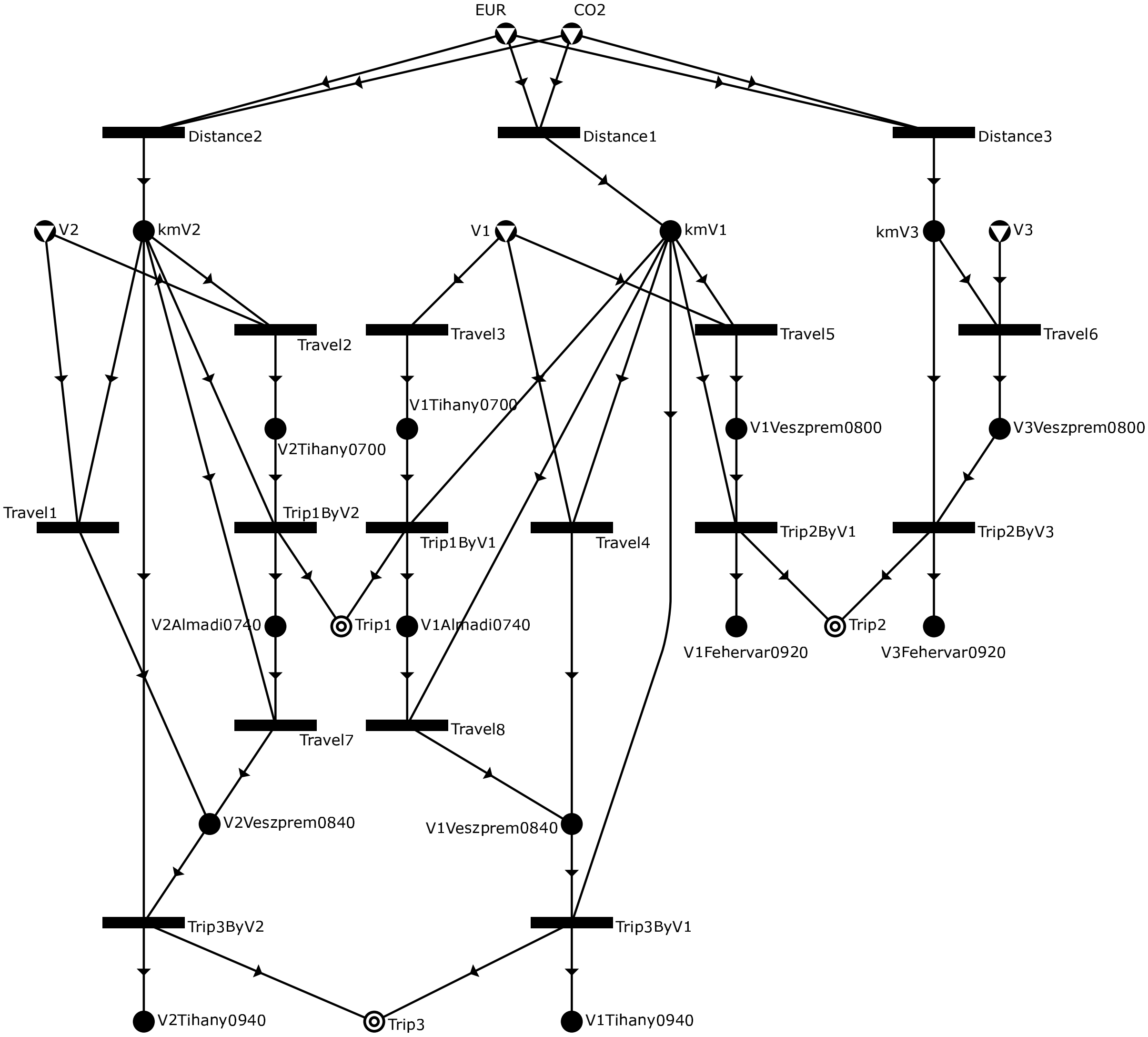


Figure 4.3: The maximal structure of the illustrative example

has completed a feasible sequence of trips. However, to keep the example simple and perspicuous this case is omitted here.

4.3 Algorithmic model generation

[10] introduced an algorithm which generates the P-graph model of the vehicle scheduling problem directly from the input data. A simplified, more readable version of this algorithm is shown by Figure 4.4. In [10] the flow rates, bounds and costs were set directly in the pseudo code. Here, to improve readibility, these are replaced by the functions $SetDistanceProperties$, $SetTravelProperties$ and $SetTripProperties$ which do nothing but set the proper bounds, flow rates and the costs of the given operating units.

In the first step the algorithm initializes the sets $\mathcal{P}, \mathcal{R}, \mathcal{O}$. The set of products is basically equivalent to the set of trips to be completed. The set of raw materials contains the available vehicles and two other materials, one modeling the cost (EUR) and the other modeling the emission (CO_2). At this point there is no operating unit defined and thus $\mathcal{O}$ is empty. n denotes a simple counter which is used to index operating units.

The second step iterates over the set of vehicles. Here, the upper bound and the cost is set for each vehicle. Furthermore, for each a vehicle, a $Distance$ type of operating unit is generated and added to the model. The function $SetDistanceProperties$ is used to set the cost, bound and the flow rate for its input parameter. To keep the pseudo code readable these simple assignment operations are not detailed here.

The third step iterates over the set of products. After the bounds and cost are set for a trip, the third step iterates over the set of vehicles which could complete the trip. This leads to step 3.1 which constructs the operating unit modeling the movement when the vehicle leaves its depot. The function $SetTravelProperties$ is analog to the function $SetDistanceProperties$.

Step 3.2 constructs the operating unit which completes the trip. The function $SetTripProperties$ is analog to the previously introduced functions.

Step 3.3 searches for vehicles which could complete two consecutive trips. If a vehicle can complete a trip and it can arrive to the start location of another trip in time, then this movement is modeled by an operating unit which is constructed here.

Theorem 4.3.1 *The algorithm generates the P-graph model of a vehicle scheduling problem in polynomial time.*

Proof The functions called by the algorithm just set the properties of the operating units. The properties of the materials and operating units are set in $O(1)$ time. A new element is also added to a set in $O(1)$ time. The complexity of the computation of the intersection of two sets depends on the implementation. It is possible to implement sets such a way that their intersections can be easily computed by a simple logical and operation. Thus, it can be computed in $O(1)$ time. Basically, each of the atomic operations can be computed in $O(1)$ time.

The first step of the algorithm iterates over the set of vehicles in $O(|\mathcal{V}|)$ time. The third step iterates over the step of products. In step 3.1 the algorithm iterates over the set of those vehicles which could complete the given tour. In the worst case scenario each vehicle could complete the given tour and this leads to $O(|\mathcal{V}|)$ time. Step 3.3 searches for trips which can be completed after the actual trip. In the worst case scenario each vehicle could complete both of the trips, i.e., the intersection becomes $\mathcal{V}$ and this leads to $O(|\mathcal{V}|)$ time. Thus, the overall complexity is $O(|\mathcal{V}| + |\mathcal{P}|(|\mathcal{V}| + |\mathcal{P}||\mathcal{V}|))$.

□

4.4 The validity of the model

To prove the validity of the solution and the model it has to be shown that a solution of the vehicle scheduling problem is also a solution to the synthesis problem modeling the vehicle scheduling problem. On the other hand, it also has to be shown that the synthesis problem has no such solution which is not a solution to the vehicle scheduling problem.

```
Input VSP problem: $\mathcal{T}$, $\mathcal{V}$, $\mathcal{A}$, $\mathbf{t}_s$, $\mathbf{t}_e$, $\mathbf{l}_s$, $\mathbf{l}_e$, $\mathbf{l}_d$, $\mathbf{D}$, $\mathbf{A}$, $\mathbf{c}_t$, $\mathbf{e}_t$, $\mathbf{v}_{max}$
Output Parametric PNS: $(\mathcal{P}, \mathcal{R}, \mathcal{O})$

1.step     $\mathcal{P} := \mathcal{T}$, $\mathcal{R} := \mathcal{V} \cup \{EUR, CO_2\}$, $\mathcal{O} := \emptyset$, $n := 1$;
2.step     for all $V_k \in \mathcal{V}$ do
               $Uc(V_k) := 1; cm(V_k) := 0;$
               $Distance_k := (\{EUR, CO_2\}, \{kmV_k\});$
               SetDistanceProperties($Distance_k$);
               $\mathcal{O} := \mathcal{O} \cup \{Distance_k\};$
           end for
3.step     for all $T_i \in \mathcal{P}$ do
               $Lp(T_i) := 1, Up(T_i) := 1;$
               for all $V_k \in \mathcal{A}(T_i)$ do
3.1 step           if $l_d(V_k) = l_s(T_i)$ do
                       $Travel_n$:=$(\{V_k\}, \{V_k l_s(T_i) t_s(T_i)\});$
                       SetTravelProperties($Travel_n$);
                   else
                       $Travel_n$:=$(\{V_k, kmV_k\}, \{V_k l_s(T_i) t_s(T_i)\});$
                       SetTravelProperties($Travel_n$);
                   end if
                   $\mathcal{O} := \mathcal{O} \cup \{Travel_n\};$
                   $n := n + 1;$
3.2 step           $T_i byV_k := (\{V_k l_s(T_i) t_s(T_i), kmV_k\}, \{V_k l_e(T_i) t_e(T_i), T_i\});$
                   SetTripProperties($T_i byV_k$);
                   $\mathcal{O} := \mathcal{O} \cup \{T_i byV_k\};$
               end for
3.3 step       for all $T_j \in \mathcal{P}$ do
                   for all $V_k \in \mathcal{A}(T_i) \cap \mathcal{A}(T_j)$ do
                       if $t_e(T_i) \leq t_s(T_j)$ and $\frac{d(l_e(T_i), l_s(T_j))}{t_s(T_j) - t_e(T_i)} \leq v_{max}(V_k)$ do
                           $Travel_n := (\{V_k l_e(T_i) t_e(T_i), kmV_k\}, \{V_k l_s(T_i) t_s(T_i)\});$
                           SetTravelProperties($Travel_n$);
                           $\mathcal{O} := \mathcal{O} \cup \{Travel_n\};$
                           $n := n + 1;$
                       end if
                   end for
               end for
           end for
```

Figure 4.4: Algorithm to generate the P-graph model of a VSP

```
SetDistanceProperties($o_k$, $\mathbf{c_t}$, $\mathbf{e_t}$)
begin
    $lb(o_k) := 0, ub(o_k) := \infty$;
    $a_{EUR,o_k} := -c_t(V_k)$, $a_{CO_2,o_k} := -e_t(V_k)$;
    $a_{kmV_k,o_k} := 1$;
end
```

Figure 4.5: The SetDistanceProperties subroutine

```
SetTravelPropertiesI($o_n$, $\mathbf{c_t}$, $\mathbf{e_t}$)
begin
    $lb(o_n) := 0, ub(o_n) := \infty$;
    $a_{V_i,o_i} := -1$, $a_{V_i l_s(T_i) t_s(T_i),o_i} := 1$;
end
```

Figure 4.6: The SetTravelProperties subroutine

It is also important to prove that the global optimum is embedded in the generated mathematical model. It has to be shown that a P-graph representing a solution structure for a vehicle scheduling problem conforms to Axioms (S1)-(S5).

Theorem 4.4.1 *The solution of a vehicle scheduling problem is also solution to the synthesis problem modeling the vehicle scheduling problem.*

Proof In order to prove this theorem, it has to be shown that a solution satisfying eqns (4.1.1)-(4.1.2) also satisfies eqns (3.8.2)-(3.8.4). Eqn (4.1.1) says that each trip has to be completed exactly once. Since each trip is a product, this is equivalent to eqn (3.8.3) if $Lp(m_j) = 1 = Up(mj)$ and thus each product is produced exactly once.

Eqn (4.1.2) says that each vehicle must perform a feasible sequence of trips. In order to complete a trip, a vehicle has to travel from one place to another. This means, that the operating unit modeling the travel must produce the material modeling the destination from the material modeling the starting location. The material modeling the starting location is either the destination for another travel and thus it is produced from another intermediate material, or it is produced by an operating unit which models that the vehicle leaves the depot. In other words, the materials and operating units modeling the sequence of feasible trips satisfy the mass balance constraints of the synthesis problem defined by eqn (3.8.2) and eqn (3.8.4).

□

Theorem 4.4.2 *There is no such solution to the synthesis problem modeling the vehicle scheduling problem which is not a solution to the vehicle scheduling problem.*

In order to prove that a solution to the synthesis problem is not a solution to the vehicle scheduling problem it has to be shown that the solution satisfies eqns (3.8.2)-(3.8.4) but violates eqns (4.1.1)-(4.1.2).

If a solution violates eqn (4.1.1), i.e., a trip is completed more than once or it is not completed at all, eqn (3.8.3) is also violated, since either $Lp(m_j)$ or $Up(mj)$ is violated. If a solution violates eqn (4.1.2) then a vehicle does not perform a feasible sequence of trips. In an infeasible sequence of trips the operating units modeling the movements of the vehicles are not connected properly and this leads to the violation of the mass balance constraints defined by eqn (3.8.2) and eqn (3.8.4). Thus, if a solution is not a solution to the vehicle scheduling problem it cannot be solution to the synthesis problem.

□

Theorem 4.4.3 *A P-graph ($\mathfrak{m}$,$\mathfrak{o}$) representing a solution structure for a vehicle scheduling problem conforms to Axioms (S1)-(S5) and thus it is a combinatorially feasible solution structure for the process-network synthesis problem ($\mathcal{P}$,$\mathcal{R}$,$\mathcal{O}$).*

Proof In case of a vehicle scheduling problem the main aim is to complete a set of trips. In the model each trip is represented by a product material. To complete all of the trips all of the products representing them has to be included in the solution structure and thus Axiom S1 (and eqn (3.8.3)) is satisfied.

In the model the raw materials represent the available resources (like EUR,CO_2 or the available vehicles). None of these materials are produced but every other material is produced from them and thus Axiom S2 (and eqn (3.8.2)) is satisfied.

The algorithm generates each operating unit systematically from the problem definition and therefore there will be no operating unit in the synthesis problem which is not given in the problem definition. Hence, Axiom S3 is satisfied (and eqns (3.8.2)-(3.8.4) consider only those operating units which are defined by the problem).

An operating unit which completes a trip has a direct connection with a product representing a trip. Each operating unit modeling a movement of a vehicle is connected to an operating unit which completes a trip. Basically, a sequence where these two types of operating units alternate can be interpreted as a route of a vehicle where it completes trips and travels between trips.

An operating unit which produces the distance is connected to each operating unit which represents a movement. Thus, each operating unit included in the solution structure has at least one path leading to a material representing a final product and Axiom S4 is satisfied (and thus the mass balance constraints expressed by eqns (3.8.2)-(3.8.4) are also satisfied.).

The fifth axiom says that there should be no isolated vertex in the solution structure which represents a material. This is ensured by the conversion algorithm since each material is added to the model through an operating unit (except EUR and CO_2 but they are also connected to the proper operating units). An isolated vertex would mean that there is a place which could not be approached by a vehicle, or a trip which could not be completed by a vehicle, etc (and this would also violate the mass balance constraints imposed on the system by eqns (3.8.2)-(3.8.4)). □

Corollary 4.4.4 *The mathematical programming model generated by the P-graph framework contains the global optimum of the vehicle scheduling problem.*

4.5 Relationship with other models

Vehicle scheduling problems with multiple depots are usually modeled by commodity flow and set partitioning formulations. In practical problems, these formulations can provide several million integer variables.

4.5.1 Commodity flow models

The problem is modeled by a complete directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where the set of vertices $\mathcal{V} = \{1, 2, \ldots, m+n\}$ is partitioned into two subsets. Subset $\mathcal{W} = \{1, \ldots, m\}$ contains a vertex k for each depot D_k and subset $\mathcal{N} = \{m+1, \ldots, m+n\}$ contains a vertex $m + j$ for each trip T_j. An arc (i, j) with $i, j \in \mathcal{N}$ corresponds to a transition between trips T_{i-m} and T_{j-m} while an arc (i, j) with $i \in \mathcal{W}$ corresponds to the start of a vehicle duty (similarly, an arc (i, j) with $j \in \mathcal{W}$ corresponds to the end of a vehicle duty). Arcs representing infeasible transitions can be removed from the graph. In these integer programs a binary variable x_{ij} is 1 if arc $(i, j) \in \mathcal{A}$ is used in the optimal solution, and it is 0 otherwise [90].

In multi-commodity formulations, an independent network is built for each depot. Then the models are based on the multigraph generated from the combination of these networks.

4.5.2 Set partitioning models

Set partitioning models enumerate all feasible routes for the vehicles and choose a subset of these routes which fulfills all restrictions. These models are derived from multi-commodity models and they use the same underlying graph structure. The resultant integer programming problem has only a few constraints (in fact a constraint for each trip to be covered) but a large number of binary variables since every feasible path through the network for each depot is represented by a binary variable in the model.

Table 4.7: Trips to be completed

$T_i \in \mathcal{T}$	$t_s(T_i)$	$l_s(T_i)$	$t_e(T_i)$	$l_e(T_i)$	$\mathcal{A}(P_i)$
T_1	10:30	Tatabánya	11:25	Fehérvár	$V_1, V_2, V_3, V_4, V_5, V_6$
T_2	10:00	Veszprém	11:40	Érd	$V_1, V_2, V_3, V_4, V_5, V_6$
T_3	11:00	Keszthely	13:40	Dunaújváros	$V_1, V_2, V_3, V_4, V_5, V_6$
T_4	14:30	Seregélyes	15:35	Budapest	$V_1, V_2, V_3, V_4, V_5, V_6$
T_5	8:00	Budapest	9:15	Fehérvár	$V_1, V_2, V_3, V_4, V_5, V_6$
T_6	13:50	Tatabánya	16:25	Keszthely	$V_1, V_2, V_3, V_4, V_5, V_6$

4.5.3 The P-graph model

In contrast with the above models, the P-graph model of vehicle scheduling does not formalize the problem as a pure integer program. The binary variables are associated with only those operating units which produce trips, and thus the result is a mixed-integer programming problem. A binary variable is 1 if the associated vehicle completes the trip and 0 otherwise.

4.6 Application

The P-graph model of vehicle scheduling has been already applied to solve some real-life problems. The following example is taken from [9] where the P-graph framework was compared with a heuristic method. In this example the aim is to create a feasible timetable with minimal cost, i.e., assign the buses to the trips such that the overall cost is minimized. The trips to be completed are listed in Table 4.7 and the available vehicle fleet in Table 4.8.

The heuristic method assigns the trips to the buses in consecutive order. The selection rule is simple: the method assigns the next trip to the bus which can complete it such that the increase in the overall cost is minimal with respect to the already scheduled trips. While this seems to be reasonable rule, the method cannot reschedule those trips which are already scheduled and thus it stucks in a local optimum.

The P-graph framework is able to generate alternate solutions (see [97] for details)

Table 4.8: The fleet of vehicles

$V_i \in \mathcal{V}$	$l_d(V_i)$	$c(V_i)$	$e(V_i)$	$v_{max}(V_i)$
V_1	Veszprém	0.8	300	60
V_2	Dunaújváros	0.8	300	60
V_3	Tatabánya	1.2	400	60
V_4	Budapest	1.2	405	60
V_5	Keszthely	1.5	375	60
V_6	Budapest	1.5	375	60

Table 4.9: Alternate solutions

Solution	Assignments	Total mileage	Cost	CO_2 emission
	T_1| T_2| T_3| T_4| T_5| T_6|			
#1	R_3| R_1| R_2| R_4| R_4| R_1|	977	888	313
#2	R_4| R_1| R_2| R_4| R_4| R_1|	977	888	313
...	...	...	...	...
#4	R_6| R_1| R_2| R_6| R_6| R_1|	977	967	309
...	...	...	...	...
#186	R_3| R_1| R_2| R_1| R_4| R_3|	1232	1212	412
...	...	...	...	...

for a given problem. Table 4.9 highlights some of the alternate solutions of the above example. While solution #1-#2 are the most cost efficient, the least emission is achieved through solution #4. The best solution generated by the heuristic method in terms of cost efficiency, which is the 186th solution of the P-graph framework is far worse than the optimal solution yielded by the P-graph framework.

This shows that while the problem seemed to be relatively easy, the optimum could not be determined by a heuristic method even with a reasonable selection rule. The P-graph model of vehicle scheduling was also used by the 5starz travel agency to plan their timetables.

4.7 Summary

In this chapter I have proposed a method and an algorithm to generate the P-graph model of a vehicle scheduling problem. Moreover, I have proven the validity of generated the model. The benefit is obvious since the mathematical programming model generated from the P-graph model contains the global optimum. However, the proposed method also highlights through the CO_2 emission that it is easy to incorporate additional factors into the model which lays the foundations for the multi-objective optimization by the P-graph method.

4.8 Related publications

Refereed Journal Papers

1. M. Barany, B. Bertok, Z. Kovacs, F. Friedler and L.T. Fan, Optimization Software for Solving Vehicle Assignment Problems to Minimize Cost and Environmental Impact of Transportation, Chemical Engineering Transactions, 21:499-504, 2010.

2. M. Barany, B. Bertok, Z. Kovacs, F. Friedler and L.T. Fan, Solving Vehicle Assignment Problems by Process-network Synthesis to Minimize Cost and Environmental Impact of Transportation, Clean Technologies and Environmental Policy, 13:637-642, 2011. (IF=1.753)

International conference presentation

1. M. Barany, B. Bertok, Z. Kovacs and F. Friedler, Optimization software for solving vehicle assignment problems to minimize cost and environmental impact of transportation, presented at the PRES 2010 (11th Conference on Process Integration, Modelling and Optimisation for Energy Saving and Pollution Reduction), Praha, Czech Republic, August 28-September 1, 2010.

Chapter 5

Modeling vehicle routing problems as process-network synthesis problems

It is needless to emphasize the practical importance of vehicle routing problems. While the problem has many different variations to properly represent a wide range of real-life situations basically all of these variations are derived from the CVRP (Capacitated Vehicle Routing Problem). Perhaps the most noted extensions are the VRPB (Vehicle Routing with Backhaul), VRPTW (Vehicle Routing with Time Windows) and VRPPD (Vehicle Routing with Pickup and Delivery). The relationships among them and the importance of the CVRP is highlighted by Figure 5.1.

Vehicle routing problems are often solved by mathematical programming methods. As it has been previously mentioned these models are often based on intuition and thus they may not contain the global optimum of the problem. In this chapter it will be shown that the CVRP can be modeled as a PNS problem by the P-graph framework. Since it is proven that the P-graph model of the PNS problem contains the global optimum of the problem this implies that the model contains the global optimum of the CVRP as well.

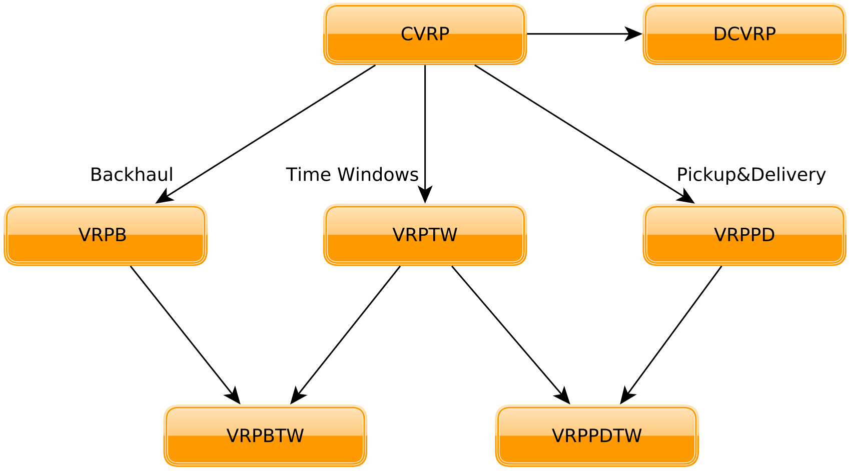


Figure 5.1: Different type of vehicle routing problems

5.1 Problem definition

Like vehicle scheduling, capacitated vehicle routing is the main topic of many papers and thus there are many variations and many different notations. This section is based on the excellent overview provided in [125] and it introduces the basic concepts of the CVRP. The aim of these problems is usually the following.

Given a set of customers with demands and a fleet of vehicles, the objective of a CVRP is to minimize the total cost of serving all customers. The total cost is a weighted function of the number of routes and their lenght (or travel time). The customers corresponding to deliveries and demands are known in advance, are deterministic and cannot be split. The vehicles are identical and are based at a single central depot. If a vehicle performs a tour then it must start it and finish it at the depot. Besides this, only capacity restrictions are imposed for the vehicles.

The CVRP generalizes the well-known Traveling Salesman Problem (TSP) and is known to be NP-hard. The TSP is a specific case of the CVRP when the number of vehicles is exactly one, and the capacity of this vehicle is great enough to satisfy all of the customer demands. In this case the aim is simply to determine a minimum-cost simple circuit visiting all the vertices of G (or in other words, a Hamiltionian circuit).

5.1.1 Basic concepts

This section introduces the basic concepts of the CVRP. Note that the representation of the road network is more or less the same in each application but there might be significant differences in the characteristics of the customers and the vehicle fleet in different applications.

The road network

For a formal description, let $G = (\mathcal{V}, \mathcal{A})$ be a complete graph where $\mathcal{V} = \{v_0, v_1, \ldots, v_n\}$ denotes the set of vertices and $\mathcal{A}$ denotes the set of arcs. Vertices $v_1, v_2, \ldots, v_n$ correspond to customers, while the depot corresponds to v_0 (in some cases the depot corresponds to a special vertex denoted by v_{n+1}). Each customer v_i has a nonnegative demand $d(v_i)$ which has to be satisfied. The demand of the depot is $d(v_0) = 0$.

There is a nonnegative $c(v_i, v_j)$ cost associated with each arc $(v_i, v_j) \in \mathcal{A}$ which represents the travel cost between v_i and v_j. The existence of loops is not permitted in general and this is ensured by defining $c(v_i, v_i) = \infty$ for all $v_i \in \mathcal{V}$. If G is directed then the cost matrix is asymmetric giving rise to asymmetric CVRP. Otherwise the problem is symmetric CVRP and $c(v_i, v_j) = c(v_j, v_i)$ for each $(v_i, v_j) \in \mathcal{A}$. In this case the arc set could be replaced by a set of undirected edges denoted by $\mathcal{E}$.

The cost matrix

In real-life applications the cost matrix often satisfies the triangle inequality, i.e.,

$$c(v_i, v_k) + c(v_k, v_j) \geq c(v_i, v_j) \qquad \text{for all } v_i, v_j, v_k \in \mathcal{V}. \tag{5.1.1}$$

This means that the simplest way between two points i, j is the direct link. The existence of the triangle inequality may be a requirement by the algorithms for CVRP and could be obtained by adding a suitable large positive quantity to the cost of each arc. However, this may distort the metric to such an extent that it may produce very bad lower and upper bounds with respect to those corresponding to the original costs. The cost matrix satisfies the triangle inequality if the cost of each arc of the graph is equal to the cost of the shortest path between its endpoints.

If the cost $c(v_i, v_j)$ is defined as the Euclidean distance between the two points corresponding to vertices i and j for each arc $(v_i, v_j) \in \mathcal{A}$ then the cost matrix is symmetric and satisfies the triangle inequality. It is a common mistake to round the real valued Euclidean arc costs to the nearest integers, since this may lead to the violation of the triangle inequality. It can be simply avoided by rounding all of the costs up.

The fleet of vehicles

A set of identical vehicles, each with capacity C is available at the depot. To ensure feasibility it is assumed that $d(v_i) \leq C$ for each $i = 1, 2, \ldots, n$. Each vehicle may perform at most one route and it is assumed that the number of vehicles is not smaller than the minimum number of vehicles needed to serve all customers. The latter can be determined by solving the Bin Packing Problem associated with the CVRP. In the associated BPP the minimum number of bins have to be determined (each of them having a capacity C) which are sufficient to load all n items (each of them having a nonnegative weight $d(v_i)$, $i = 1, 2, \ldots, n$).

Define the number of vehicles by l. In the CVRP a collection of exactly l simple circuits has to be found with minimum cost. Each circuit corresponds to a vehicle route and the cost is defined as the sum of the costs of the arcs belonging to the circuits. Moreover, in a valid solution each circuit visits the depot, each vertex representing a customer is visited by exactly one circuit and the sum of the demands of the vertices visited by a circuit does not exceed the vehicle capacity.

In some CVRP instances, the number of available vehicles may be greater than the minimum number of vehicles necessary to serve all customers. This renders it possible to leave some of the vehicles unused. In this case a fixed cost may be associated with the use of a vehicle and the objective function could be extended with an additional objective requiring the minimization of the number of vehicles used. In other cases, the capacities of the available vehicles are different while sometimes routes containing only one customer are not allowed.

The Distance-Constrained Vehicle Routing Problem

If the capacity constraint is replaced by a maximum length (or time) constraint for each route, the CVRP becomes DVRP. In particular, a nonnegative length $t(v_i, v_j)$ is associated with each arc (v_i, v_j) and the total length of the arcs of each route cannot exceed the maximum route length. If the vehicles are different, the maximum route lengths may differ.

In another approach, the arc lengths may represent travel times. In this case it is possible to introduce a service time $s(v_i)$ for each customer v_i representing a time period for which the vehicle must stop at its location. The service times could be added to the travel times of the arcs, i.e,

$$t(v_i, v_j) = t'(v_i, v_j) + \frac{s(v_i)}{2} + \frac{s(v_j)}{2} \qquad \forall (v_i, v_j) \in \mathcal{A} \tag{5.1.2}$$

where $t'(v_i, v_j)$ is the original travel time of arc (v_i, v_j). In general, the cost and length matrices are the same, i.e.,

$$t(v_i, v_j) = c_(v_i, v_j) \qquad \forall (v_i, v_j) \in \mathcal{A}$$

The Distance-Constrained Capacitated Vehicle Routing Problem

If the vehicle capacity and the maximum distance is constrained, the CVRP becomes DCVRP.

5.1.2 Formal definition

Let v_x and v_y be two different locations, i.e., $v_x, v_y \in \mathcal{V}$, $v_x \neq v_y$. Denote the arc, i.e., the road section leading from v_x to v_y by an ordered pair $(v_x, v_y) \in \mathcal{A}$. The set of vehicles is denoted by $\mathcal{B} = \{b_1, b_2, \ldots, b_l\}$. $\mathcal{H}_k(b_j) \subseteq \mathcal{V} \times \mathcal{V}$ denotes the kth set of road sections traveled by vehicle b_j.

Definition 5.1.1 (Tour) *The kth set of road sections traveled by vehicle V_j is called a tour if there exists such an ordering of the road sections in $\mathcal{H}_k(b_j)$ that*

$$\mathcal{H}_k(b_j) = \{(v_{x1}, v_{y1}), (v_{x2}, v_{y2}), \ldots, (v_{xq}, v_{yq})\}$$

where $v_{yi} = v_{xi+1}$ for all $i = 1, \ldots, q-1$ and $v_{yq} = v_{x1}$.

$\mathcal{H}(b_j)$ denotes the set of all of the tours performed by b_j, i.e.,

$$\mathcal{H}(b_j) = \mathcal{H}_1(b_j) \cup \mathcal{H}_2(b_j) \cup \ldots \cup \mathcal{H}_{r(b_j)}(b_j)$$

where $r(b_j)$ is the number of tours performed by b_j and $\mathcal{H}$ denotes the complete set of tours performed by the fleet, i.e.,

$$\mathcal{H} = \bigcup_{j=1}^{l} \mathcal{H}(b_j).$$

Using these notations, we can formalize the CVRP in the following way. Given n customers with demands and l vehicles, the objective of a CVRP is to find a set of feasible tours such that

- Each customer is served (each customer demand is satisfied)

$$\forall v_i \in \mathcal{V} : d(v_i) > 0 \Rightarrow \exists v_x \quad (v_x, v_i) \in \mathcal{H} \tag{5.1.3}$$

- Each customer is served by exactly one vehicle (each vertex is visited by exactly one circuit)

$$\forall v_i \in \mathcal{V} : |\{(v_x, v_y) \in \mathcal{H} : v_y = v_i\}| = 1 \tag{5.1.4}$$

- Each tour is started and finished at the depot (each circuit visits the depot)

$$\begin{aligned} \forall b_j \in \mathcal{B}, \forall k \in \{1, 2, \ldots, r(b_j)\} : \mathcal{H}_k(b_j) = \{(v_{x1}, v_{y1}), (v_{x2}, v_{y2}), \ldots, (v_{xq}, v_{yq})\} \\ \Rightarrow v_{x1} = v_{yq} = v_0 \end{aligned} \tag{5.1.5}$$

- The sum of the demands of the customers visited by a tour does not exceed the vehicle capacity

$$\forall b_j \in \mathcal{B}, \forall k \in \{1, 2, \ldots, r(b_j)\} : \sum_{(v_x, v_y) \in \mathcal{H}_k(b_j)} d(v_y) \leq C(b_j) \tag{5.1.6}$$

- The cost is minimal

$$\sum_{(v_x, v_y) \in \mathcal{H}} c(v_x, v_y) \tag{5.1.7}$$

Note that the first two constraints could be expressed by one single constraint. It is separated into two different constraints because in some practical situations it could be preferable if more than one vehicle can serve a customer. In this case, the second constraint can be ignored.

Table 5.1: Mapping the concepts of CVRP to PNS

CVRP	PNS
goods in the depot	raw materials
goods at the customer	products
locations	intermediate materials
traveling (transportation)	operating units

5.2 Modeling capacitated vehicle routing as a synthesis problem

This section introduces the concepts of the P-graph model of the capacitated vehicle routing problem. These concepts are illustrated by examples and then some important properties of the model and the solution yielded by the model are proven.

5.2.1 Analogy between the concepts of capacitated vehicle routing and process-network synthesis

Like in the case of vehicle scheduling, it is possible to find an analogy between the concepts of capacitated vehicle routing and the concepts of process synthesis. The goal of vehicle routing is to satisfy customer demands by distributing the available goods from the depot. This is similar to a process system, where a set of products have to be manufactured from the available raw materials.

In order to satisfy customer demands, the goods have to travel between a sequence of places (or to be exact, they have to be transported) starting from the depot until they reach their respective destinations. This is similar to the sequence of intermediate materials which leads from the raw materials to the products in a process system. A travel has a starting location and a destination similarly to an operating unit, which has a set of inputs and a set of outputs. This discussion is summarized by Table 5.1.

5.2.2 The P-graph model of capacitated vehicle routing

Vehicle routing problems were first modeled by the P-graph framework in [2, 67]. These conference presentations demonstrated by examples, without any comprehensive description, that it is possible to define a CVRP as a PNS. The first detailed description was given in [8]. This section expands the concepts of the above analogy and introduces the complete P-graph model of capacitated vehicle routing.

Modeling locations, customers and goods

The representation of locations are closely related to the representation of customers and goods. It is modeled by an intermediate material if a vehicle is at a given place at a given time. Similarly, an intermediate material models if a good is at a given place at a given time.

It is easy to clarify this with a simple example. Given three customers (A, B, C), two vehicles $(V1, V2)$ and a good to be transported $(G1)$ the following set of materials is defined: $\mathcal{M} = \{V1A, V1B, V1C, V2A, V2B, V2C,$ $G1AV1, G1BV1,$ $G1CV1,$ $G1AV2,$ $G1BV2, G1CV2\}$, where

- $V1A$: represents that $V1$ is at customer A
- $V1B$: represents that $V1$ is at customer B
- $\vdots$
- $V1C$: represents that $V1$ is at customer C
- $G1AV1$: represents that $G1$ is at A on $V1$
- $G1BV1$: represents that $G1$ is at B on $V1$
- $\vdots$
- $G1CV2$: represents that $G1$ is at C on $V2$

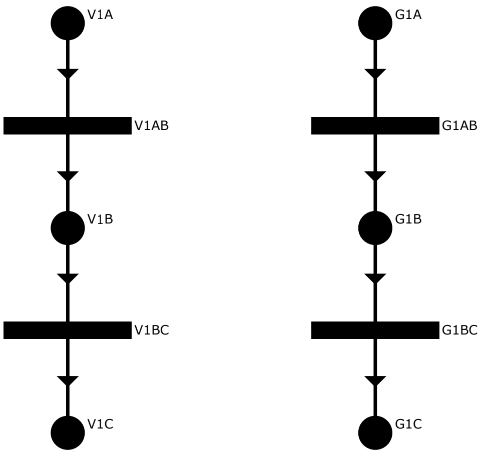


Figure 5.2: Movements of vehicles and goods

Modeling the movements

The movements are modeled by operating units. Using material set $\mathcal{M} = \{V1A, V1B, V1C, G1A, G1B, G1C\}$, the following set of operating units can be defined: $\mathcal{O} = \{(\{V1A\}, \{V1B\}), (\{V1B\}, \{V1C\}), (\{G1A\}, \{G1B\}), (\{G1B\}, \{G1C\})\} = \{V1AB, V1BC, G1AB, G1BC\}$ where

- $V1AB$ represents that $V1$ travels from A to B
- $V1BC$ represents that $V1$ travels from B to C
- $G1AB$ represents that $G1$ is transported from A to B
- $G1BC$ represents that $G1$ is transported from B to C

These operating units can be linked by their inputs and their outputs and thus it is possible to form complex routes. For example, if $V1$ advances from A to B and then it heads from B to C then $V1AB$ and $V1BC$ becomes connected through $V1B$. Similarly, if $G1$ moves from A to B then from B to C then $G1AB$ and $G1BC$ becomes connected. See Figure 5.2 for illustration.

Note that these are just the movements of the vehicles and the goods, until this point there was no actual transportation performed. In a real tour, $V1$ somewhere

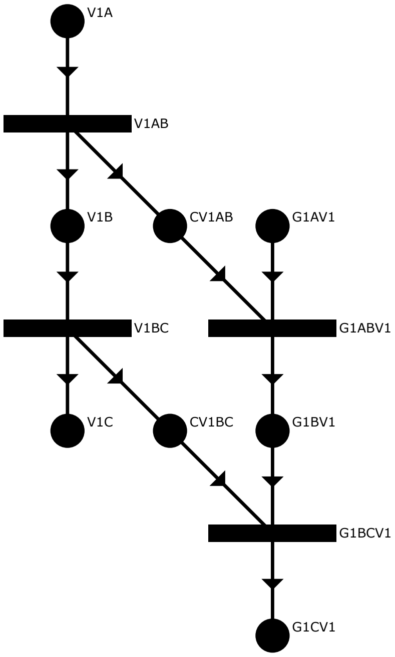


Figure 5.3: $V1$ transports $G1$ from A to B and then transports it further to C

unloads $G1$ and heads back to the depot without any cargo. This implies that the movements of the vehicles and the goods must be separated.

If an operating unit is representing a movement of a vehicle between two places, then the lower bound and the upper bound on its capacity must set to 1 (i.e. $1 = lb(o_i) \leq x_i \leq ub(o_i) = 1$). This ensures that the vehicle completely travels that section, or does not travels it at all.

Modeling the transportation

To model the transportation of good $G1$ by $V1$ between places A and B, the operating unit modeling the movement of $V1$ between A and B have to be connected to the operating unit modeling the movement of $G1$ between A and B. A new material, $CV1AB$ is used for this purpose, which expresses the transportation capacity of $V1$ between A and B. See Figure 5.3 for illustration.

$CV1AB$ is interpreted in the following way. If the capacity of a vehicle is 10000

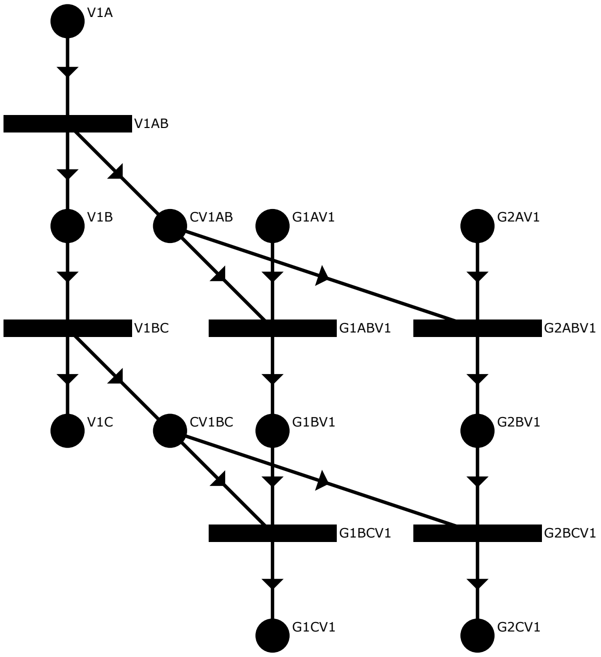


Figure 5.4: Transportation of two goods by a single vehicle

units then a travel between customers A and B creates 10000 units of transportation capacity between A and B, i.e., $V1AB$ creates 10000 $CV1AB$ (the flow rate is 10000). On the other hand, the ratio of $CV1AB$ and $G1AV1$ is 1:1, meaning that the transportation of 1 unit of $G1$ requires 1 unit from the capacity of $V1$ (the flow rates are 1-1).

If a vehicle transports more than one good, then its transportation capacity is split among the goods. See Figure 5.4 for illustration.

Modeling the loading operation

A vehicle must be loaded with goods in the depot in order to satisfy customer demands. In contrast with the traditional CVRP where the loading operation is performed in a dedicated depot, the P-graph framework uses a more general loading operation which can be performed nearly anywhere. However, if every loading operation is performed at a dedicated place then that place essentially becomes the depot. Figure 5.6 illustrates a loading operation.

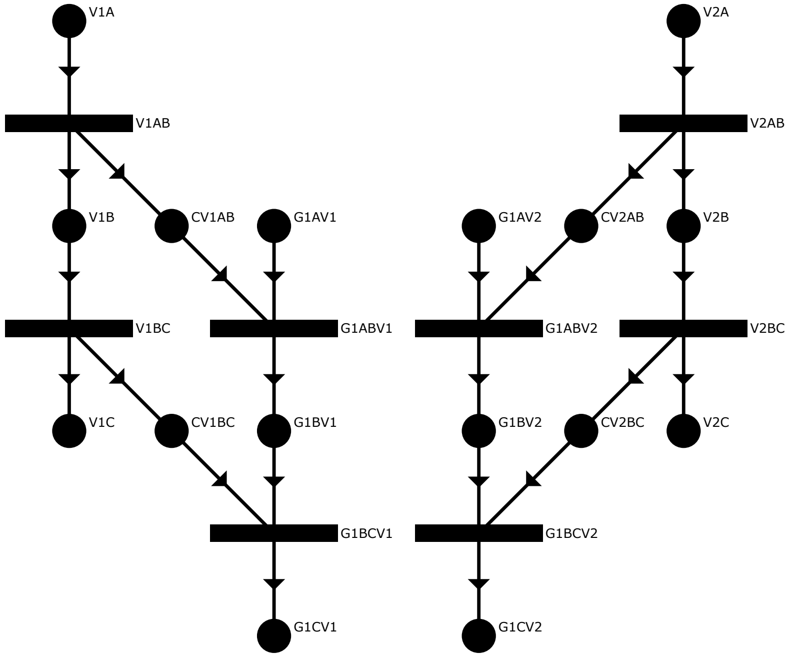


Figure 5.5: Transportation of a single good by two vehicles

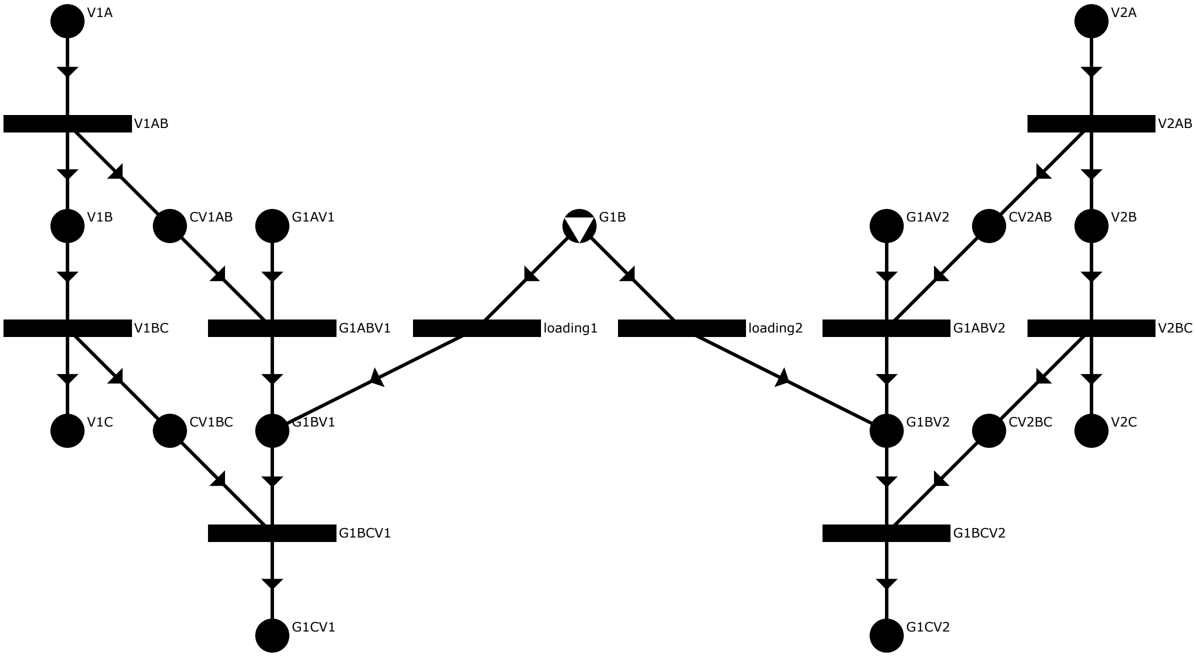


Figure 5.6: $G1$ is loaded onto $V1$ and $V2$

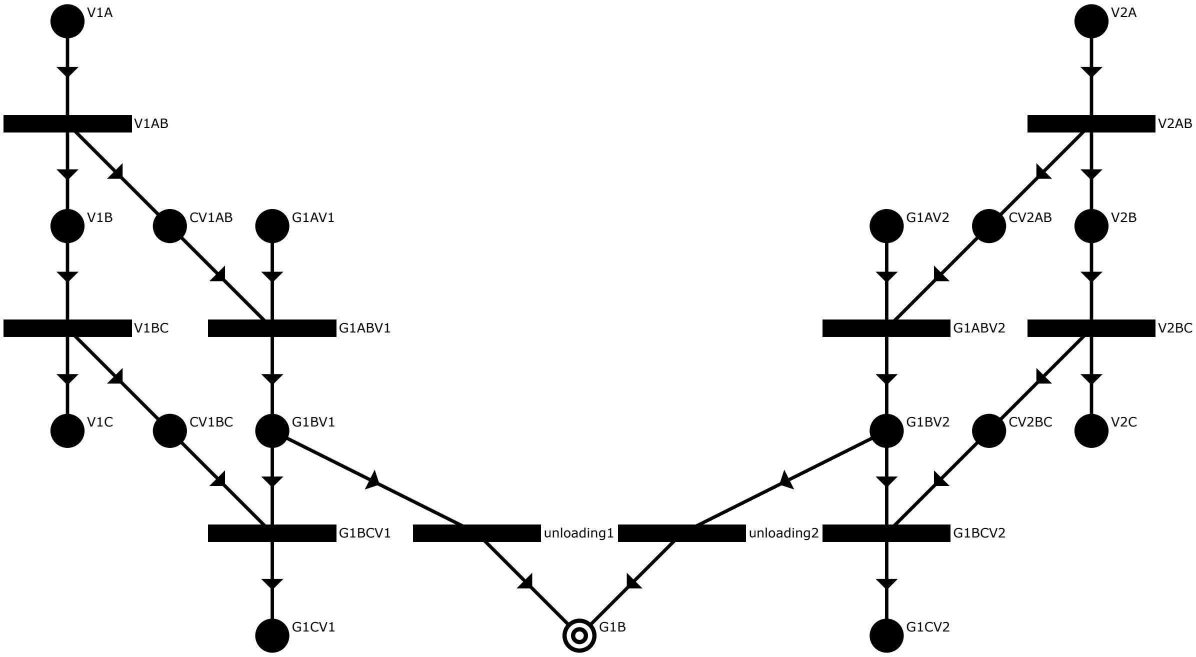


Figure 5.7: $V1$ and $V2$ unloads $G1$

The loading operation involves a raw material, which models a good that should be delivered to the customers. During the loading this material is converted into a material which models that the good is on the vehicle. Note, that in this example place B can be interpreted as the depot since both vehicle is loaded there.

It is possible to impose an upper bound on the availability of a raw material modeling a good which is to be distributed among the customers, but in practice this is seldom necessary. In practical applications, the dimensions of the goods and the capacity of the vehicles may differ. For example, there is a specified number of containers in the depot but the capacity of the vehicle is given in kilograms. In this case, the conversion is indicated by the flow rates of the operating unit modeling the loading operation. This conversion might be reversed during the unloading.

Modeling of the unloading operation

The unloading operation is considered to be the opposite of the loading operation and it is also modeled by operating units. See Figure 5.7 for illustration.

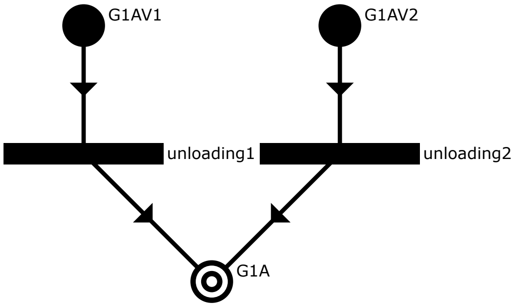


Figure 5.8: An unloading operation

Figure 5.7 illustrates that $V1$ and $V2$ delivers $G1$ at B. The interpretation is the following: the goods on the vehicles are converted into goods at the customers and thus the customer demands are satisfied.

The customer demand is represented by a material, in this example by the product $G1B$. The lower bound of $G1B$ defines the minimum required quantity from $G1$ at B. If this demand cannot be satisfied the problem is generally considered to be infeasible.

On Figure 5.7 a customer demand is satisfied by multiple vehicles. This violates the classical definition of the VRP but it could address some practical situations.

Mutually exclusive operating units

The concept of mutually exclusive operating units can be implemented in several ways. The simpler, but less elegant method is to add more materials and operating units to the model. Figure 5.8 illustrates an unloading operation: G could be delivered by both $V1$ and $V2$ in A.

In order to illustrate how the mutual exclusion works, this small example is expanded with the raw material UA, two intermediate materials, $V1U$ and $V2U$ and two operating units, $(\{UA\}, \{V1U\})$ and $(\{UA\}, \{V1U\})$. See Figure 5.9 for the extended example.

If the lower bounds on the capacity of these newly added operating units is set to one and the upper bound on the availibilty of UA is also set to one, then these operating units cannot operate simultaneously and thus the customer demand cannot

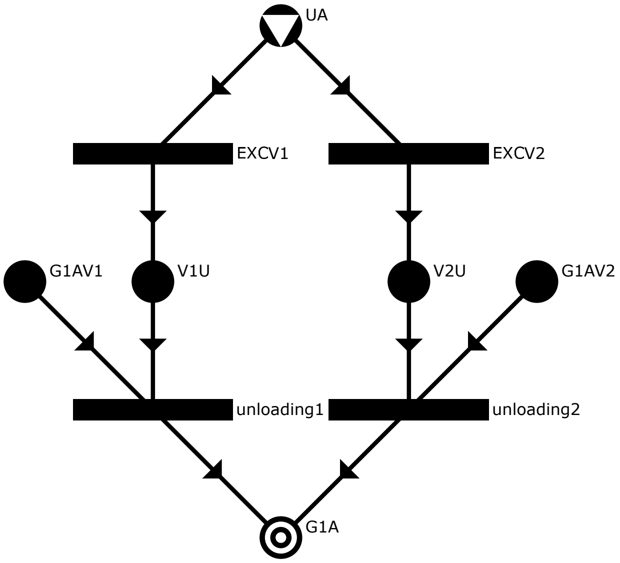


Figure 5.9: An unloading operation with mutual exclusion

be satisfied by more than one vehicle.

Although this is a completely valid solution, the addition of operating units and materials may be a little tedious. Moreover, if the model is expanded with additional operating units the number of binary variables increase. An approach that handles mutually exclusive operating units algorihmically would be more efficient and more elegant.

In the algorithmic approach, for each operating unit a set of operating units is defined. If the operating unit is included in the solution structure then the operating units in the set are excluded from the solution immediately. Thus, the elements of the set are said to be mutually exclusive with the operating unit. Formally, let (m, o) be a solution structure and let o'' be an operating unit. If the set of mutually exclusive operating units is denoted by Υ:

$$\Upsilon = \{o' | o' \in O \text{ and } o'' \in (m, o) \Rightarrow o' \notin (m, o)\} \tag{5.2.1}$$

Note that a set of mutually exclusive operating units can be defined not just for a single operating unit, but for a set of operating units too.

The mutually exclusive sets will be taken into consideration by algorithms of the

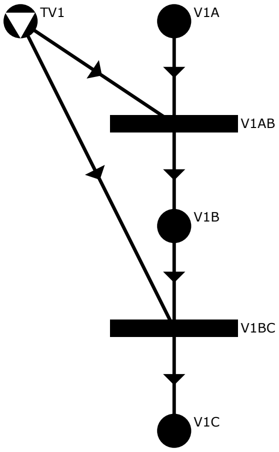


Figure 5.10: Modeling time constraints

P-graph framework like SSG and ABB (i.e., where decisions must be made about the inclusion and the exclusion of the operating units). Whenever an algorithm includes an operating unit into the structure, it excludes the mutually exclusive operating units from the structure. In a way, the mutual exclusion can be thought as the extension of decision mappings. The detailed description of mutual exclusions and their application by the P-graph algorithms can be found in [70].

Modeling the distance constraints

The distances from the cost matrix are utilized in the following way. The total distance which is covered by a vehicle is modeled by a raw material. This raw material is feeded as an input to every operating unit modeling a movement of the respective vehicle. The input flow rate of this material is set to be the distance between the two places connected by the operating unit (which is essentially an entry in the cost matrix). See Figure 5.10 for illustration.

Material $TV1$ represents the total distance which can be covered by vehicle $V1$. It is feeded to $V1AB$ and $V1BC$ which represent the movements of $V1$.

The travel cost is computed from this material also. If a price is associated with the raw material then the travel cost is proportional to the consumption of the material. Actually, this could be modeled through the proportional cost of the operating unit, but perhaps it is more elegant to set a price on the raw material.

Note, that the raw material does not represent necessarily the distance, it may represent travel time, or any other practical thing which can be related to the movement of the vehicle. This can be used, for example, to impose a CO_2 emission quota on the fleet. Since multiple resources can be assigned to a vehicle simultaneously, it is possible to cover a wide range of sophisticated practical applications.

5.3 Algorithmic model generation

[8] introduced an algorithm which generates the P-graph model of the capacitated vehicle routing problem directly from the input data. Besides this algorithm, a heuristic approach is also presented here which can be used to reduce the number of binary variables during the model generation.

5.3.1 Model generation

The input of the algorithm is a CVRP problem which is defined by the following parameters: a $(\mathcal{V}, \mathcal{A})$ weighted complete graph, which represents the underlying road-network; a vector $\mathbf{d}$ which represents the demands of the customers; a nonnegative integer N_v which represents the number of vehicles, i.e, the size of the fleet; and finally a nonnegative integer C_v which represents the capacity of the vehicles. Note, that $(\mathcal{V}, \mathcal{A})$ is essentially a different representation of the cost matrix.

The dimension of $\mathbf{d}$ is equal to the cardinality of $\mathcal{V}$. For simplicity, it is assumed that only one good is involved in the example. Thus, each entry of $\mathbf{d}$ represents the demand of customer i from the good. In case of multiple goods vector $\mathbf{d}$ becomes a matrix $\mathbf{D}$. The actual implementation is based on the latter case, but the pseudo code is simpler with the former.

If the capacities of the vehicles are not the same then C_v becomes a vector $\mathbf{C_v}$ which has a dimension of N_v. This is a more general case which deviates from the standard CVRP but it is much more practical and thus it is supported in the computer implementation.

```
Input CVRP problem: $(\mathcal{V}, \mathcal{A})$, $\mathbf{d}$, $N_v$, $C_v$
Output Parametric PNS: $(\mathcal{P}, \mathcal{R}, \mathcal{O})$

0. step   $\mathcal{V} \neq \emptyset$, $\mathcal{A} \neq \emptyset$, $N_v > 0$, $C_v > 0$, $\mathcal{P} := \emptyset$, $\mathcal{R} := \emptyset$, $\mathcal{O} := \emptyset$, $l = 0$;
1. step   for all $v_i \in \mathcal{V}$ do
              if $d(v_i) > 0$ do
                  $\mathcal{P} \cup \{GV_i\}$;
                  $Lp(GV_i) := d(v_i)$;
                  $\mathcal{R} \cup \{GV_0\}$;
              end if
          end for
2. step   while $l < N_v$ do
              $\mathcal{R} \cup \{TK_l\}$;
              set $cm(TK_l)$;
2.1 step      for all $v_i \in \mathcal{V}$ do
                  if $v_0$ do
                      $\mathcal{O} \cup \{\{GV_0\}, \{GV_0K_l\}\}$;
                  else if $d(v_i) > 0$ do
                      $\mathcal{O} \cup \{\{GV_iK_l\}, \{GV_i\}\}$;
                  end if
              end for
2.2 step      for all $(v_i, v_j) \in \mathcal{A}$ do
                  $\mathcal{O} \cup \{\{K_lI, TK_l\}, \{K_lJ, CK_lIJ\}\}$;
                  $\mathcal{O} \cup \{\{GV_iK_l, CK_lIJ\}, \{G_jK_l\}\}$;
                  $ub(\{\{K_lI, TK_l\}, \{K_lJ, CK_lIJ\}\}) := 1$;
                  $lb(\{\{K_lI, TK_l\}, \{K_lJ, CK_lIJ\}\}) := 1$;
                  set rates of materials
              end for
              $l := l + 1$;
          end while
```

Figure 5.11: Algorithm to generate the P-graph model of a VRP

At the beginning of the algorithm it is practical to validate the input data and initialize the output data. This is not an important part of the pseudo code but a vital part of the implementation thus it is indicated as the 0. step.

In the first step the algorithm identifies the raw materials and the products by iterating over the vertices of $\mathcal{V}$. The products are derived directly from the demands and the raw materials from the products since it is assumed that there is enough product to satisfy all demand. Essentially, if a customer has a nonnegative demand from a good then a product material is added to the model which represents the given good at the given place. In the same time, a raw material is also added to the model. The depot is an integral part of the problem definition thus it becomes the place, which is indicated in the name of each raw material representing a good.

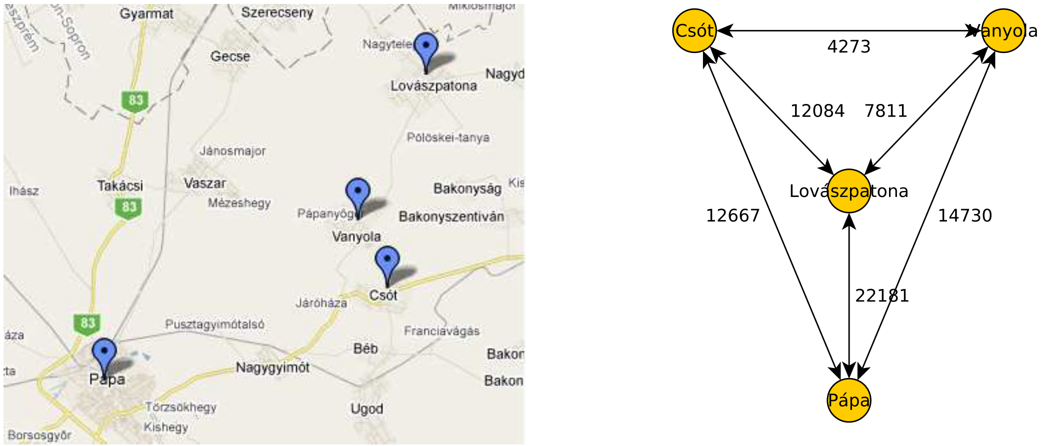
As it has been mentioned above, for the sake of simplicty the pseudo code involves only one good. Thus, the product modeling the good at place v_i is denoted by GV_i and the demand of customer v_i by $d(v_i)$.

The operating units are added to the model in the second step. In the beginning of each iteration, the raw material representing the distance covered by vehicle k is added to the model. It is crucial to set a price for this material since it is the main component of the overall cost. In case of a DCVRP problem an upper bound should be also set on this material. If other constraints are imposed on the vehicle then the materials representing them should be added similarly.

After this material is added the algorithm iterates over set $\mathcal{V}$ again. During a similar iteration in the first step the raw materials and the products were identified and added to the model. This iteration complements the model with the operating units modeling the corresponding loading and unloading operations.

The next loop iterates over set $\mathcal{A}$. An arc connects two places thus it defines an operating unit modeling a movement between the two endpoints. This also defines an operating unit which represents a transportation between the two places. The lower bound and the upper bound are set to one and the input flow rate of the capacity is equal to the weight of the arc (i. e., the distance between the two places).

After this iteration is finished, the algorithm proceeds with the next vehicle. If all of the vehicles have been processed, the algorithm terminates.



(a) Positions on the map

(b) Graph representation

Figure 5.12: A small example with a depot and three customers

Theorem 5.3.1 *The algorithm generates the P-graph model of a vehicle routing problem in polynomial time.*

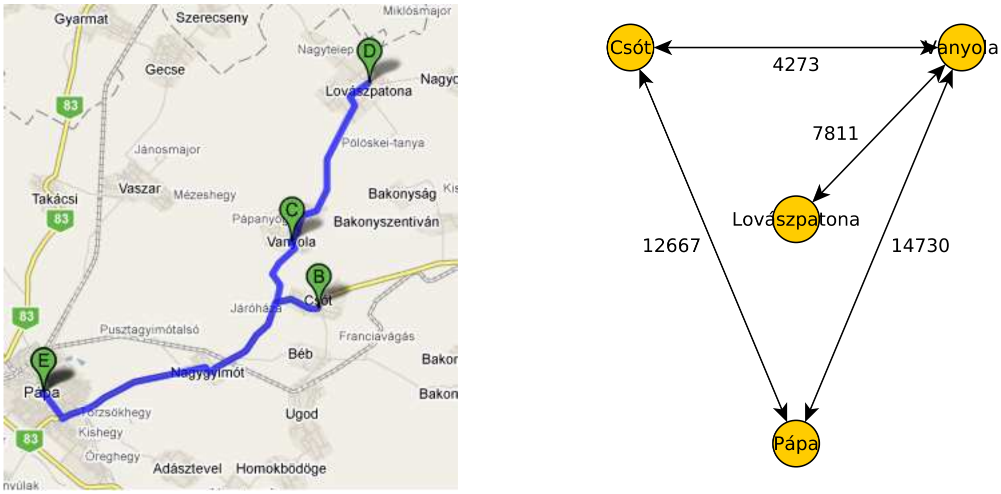
Proof The "atomic" operations of the algorithm are very simple. Both the time of an assignment and the addition of an element to a set is $O(1)$. The first step iterates over the set of vertices in $O(|\mathcal{V}|)$ time. Step 2.1 iterates over the set of vertices again in $O(|\mathcal{V}|)$ time and step 2.2 iterates over the set of edges in $O(|\mathcal{A}|)$ time. However, this is repeated N_v times. Thus, the algorithm runs in $O(|\mathcal{V}|+N_v(|\mathcal{V}|+|\mathcal{A}|))$ time. □

5.3.2 Model simplification

In case of a MILP the number of binary variables is a major factor in the computational time. If the number of binary variables is lowered then the computational time decreases. In the P-graph model of the CVRP the number of binary variables can be reduced by the following heuristic.

The triangle inequality

Figure 5.12 illustrates a small example with a depot (Pápa) and three customers (Vanyola, Csót, Lovászpatona). The weights of the edges of the complete graph in



(a) Positions on the map

(b) Graph representation

Figure 5.13: The only way to Lovászpatona goes through Vanyola

Figure 5.12b express the distances between the places in meters. A closer look on the map reveals that Lovászpatona can be reached only through Vanyola. This can be exploited in the following way.

Given three vertices, A,B and C and three edges AB, BC and AC connecting these vertices the triangle inequality says that

$$AB + BC \leq AC \tag{5.3.1}$$

In Euclidean geometry the triangle inequality is a theorem about distances. In the above case it can be interpreted as the trip from A to C is not longer through B than the direct trip from A to C and thus AC might be redundant. To minimize numerical errors it is subservient to introduce a scalable ϵ and transform eqn 5.3.1 into

$$AB + BC \leq AC(1 + \epsilon) \tag{5.3.2}$$

Using this equation it is easy to see that the edges connecting Pápa and Csót to Lovászpatona are redundant and they can be removed from the complete graph since Lovászpatona can be reached only through Vanyola. The resultant graph is shown on Figure 5.13b.

Table 5.2: Test instances

Problem	Vertices	Edges	Materials	Operating units
Route 303	4	12	38	56
Route 305	6	30	84	132
Route 307	7	42	97	182

Table 5.3: Decrease in the number of operating units

Problem	Redundant	Nonredundant	Decrease (#)	Decrease (%)
Route 303	56	40	16	29
Route 305	132	124	8	6
Route 307	182	134	48	26

Computational results

[8] contains computational results which are based on a real-life R&D project and some of them are shown here to illustrate the performance improvement achieved by the model simplificaton. All of these test instances use two vehicles to satisfy customer demands (Note, that this doubles the number of operating units).

The characteristics of the test instances are shown in Table 5.2, which indicates the number of vertices and edges of the complete graph and the number of materials and operating units in the P-graph representation. The number of vertices includes the depot.

Table 5.3 shows the decrease in the number of operating units. The first coloumn shows the original number of operating units in a given instance and the second coloumn shows the number of operating units after the removal of the redundant ones with the triangle inequality.

Table 5.4 shows the improvement in performance. These computational times belong to the PNS solver, which implements the algorithms of the P-graph framework. It is not indicated in a different table, but the nonredundant graphs yield the same optimum values as the redundant ones.

Table 5.4: Improvement in performance - PNS Solver

Problem	Redundant	Nonredundant	Improvement	Improvement (%)
Route 303	0.604 s	0.217 s	0.387 s	64
Route 305	2 min 40.642 s	2 min 3.702 s	36.948 s	23
Route 307	3 min 45.802 s	1 min 2.659 s	2 min 43.143 s	72

Table 5.5: Improvement in performance - CBC

Problem	Redundant	Nonredundant	Improvement	Improvement (%)
Route 303	1.39 s	0.56 s	0.83 s	59
Route 305	39.53 s	31.08 s	8.45 s	21
Route 307	68.65 s	23.57 s	45.08 s	66

To separate the benchmark results of the PNS solver and the underlying mathematical model, [8] used a software to obtain the MILP model in a conventional file format which can be read by general purpose mathematical solvers like CBC or CPLEX. Table 5.5 shows the computational results of CBC while the results of CPLEX are displayed in Table 5.6. These two mathematical solvers yielded the same optimums and the same routes as the PNS solver.

It can be seen from the benchmark results that the triangle inequality yields a notable improvement in the performance and this improvement is independent from the implementation of the PNS solver. However, it is considered only as a heuristic since it is possible to construct such problem instances where the utilization of the triangle inequality does not lead to the optimal solution.

Table 5.6: Improvement in performance - CPLEX

Problem	Redundant	Nonredundant	Improvement	Improvement (%)
Route 303	0.05 s	0.03 s	0.02 s	40
Route 305	4.39 s	2.89 s	1.5 s	34
Route 307	5.09 s	0.86 s	4.23 s	83

5.4 The validity of the model

To prove the validity of the solution and the model it has to be shown that a solution of the capacitated vehicle routing problem is also a solution to the synthesis problem modeling the capacitated vehicle routing problem. On the other hand, it also has to be shown that the synthesis problem has no such solution which is not a solution to the capacitated vehicle routing problem.

It is also important to prove that the global optimum is embedded in the generated mathematical model. Thus, it has to be shown that a P-graph representing a solution structure for a capacitated vehicle routing problem conforms to Axioms (S1)-(S5).

Theorem 5.4.1 *The solution of a capacitated vehicle routing problem is also solution to the synthesis problem modeling the capacitated vehicle routing problem.*

Proof In order to prove this theorem it has to be shown that a solution satisfying eqns (5.1.3)-(5.1.6) also satisfies eqns (3.8.2)-(3.8.4). Eqn (5.1.3) says that all customer demand have to be satisfied. In the synthesis model of the capacitated vehicle routing problem each customer demand is modeled by a raw material (at the depot) and a product material (at the customer). Thus, the customer demands are satisfied if the product materials are produced by the process system, i.e., eqn (3.8.3) is satisfied.

Eqn (5.1.4) says that each customer is served by exactly one vehicle. In the synthesis model this is enforced by the concept of mutually exclusive operating units. If they are implemented by additional operating units, they must satisfy the mass balance constraints imposed on the raw materials by eqn (3.8.2).

Eqn (5.1.5) says that each tour is started and finished at the depot. In the synthesis model there is no dedicated depot but if all of the vehicles are loaded at a given place then that place becomes the depot. However, this place is modeled by an intermediate material. This means that besides it is consumed by the process system, it is also produced. Since each place is modeled by intermediate materials, starting from the place modeling the depot, there exists a sequence of intermediate materials that leads back to the depot. If a vehicle travels through such a sequence it also satisfies the mass balance constraints imposed on the intermediate materials by eqn (3.8.4).

Eqn (5.1.6) says that the sum of the demands of the customers visited by a tour does not exceed the vehicle capacity. In the synthesis model, the capacity of the vehicle is modeled by an intermediate material. If the capacity is not exceeded then the mass balance constraints imposed on the intermediate materials by eqn (3.8.4) are satisfied.

□

Theorem 5.4.2 *There is no such solution to the synthesis problem modeling the capacitated vehicle routing probem which is not a solution to the capacitated vehicle routing problem.*

In order to prove that a solution to the synthesis problem is not a solution to the capacitated vehicle routing problem it has to be shown that the solution satisfies eqns (3.8.2)-(3.8.4) but violates eqns (5.1.3)-(5.1.6).

If a solution violates eqn (5.1.3), i.e., there is at least one customer who is not served, eqn (3.8.3) is also violated since there is a product which is not produced properly by the process system.

If eqn (5.1.4) is violated, i.e., a customer is served by more than one vehicle, it means that the mutually exclusive operating units consume more raw materials than what is available and thus eqn (3.8.2) is also violated.

The violation of eqn (5.1.4) means that a tour is not started or finished at the depot. However, this also violates eqn (3.8.4).

If the sum of the demands of the customers visited by a tour exceed the vehicle capacity, i.e., eqn (5.1.6) is violated, then the mass balance constraints imposed on the intermediate materials by eqn (3.8.4) are also violated.

□

Theorem 5.4.3 *A P-graph (m,o) representing a solution structure for a capacitated vehicle routing problem conforms to Axioms (S1)-(S5) and thus it is a combinatorially feasible solution structure for the process-network synthesis problem $(\mathcal{P},\mathcal{R},\mathcal{O})$.*

Proof In case of a CVRP the main aim is to satisfy given customer demands. In the model each demand is represented by a product material. To satisfy all of the customer demands all of the products representing them has to be included in the solution structure and thus Axiom S1 (and eqn (3.8.3)) is satisfied.

In the model the raw materials represent the available goods and resources. None of these materials are produced and thus Axiom S2 (and eqn (3.8.2)) is satisfied.

The algorithm generates each operating unit systematically from the problem definition and therefore there will be no operating unit in the synthesis problem which is not given in the problem definition. Hence, Axiom S3 is satisfied (and eqns (3.8.2)-(3.8.4) consider only those operating units which are defined by the problem).

An operating unit which models an unloading has a direct connection with a product. In a solution, each operating unit modeling a transportation is part of a sequence of operating units which lead from a loading operation to an unloading operation. Thus, there exists a path from each of these operating units which leads to a product. Each operating unit which represents the corresponding movement of a vehicle is connected to the operating units modeling the transportations through the material representing the capacity. Thus, there exists a path which leads to a product from each of these operating units as well.

The above sequence of operating units should be interpreted as a route of a vehicle. The vehicle loads the necessary goods (through the operating unit representing the loading) and then heads for the customer (through the operating units representing the movements) with the given goods (through the operating units representing the transportation) to satisfy demands (with the operating unit modeling the unloading operation). Since in a solution each operating unit is connected to the product somehow it can be stated that Axiom S4 (and the mass balance constraints expressed by eqns (3.8.2)-(3.8.4)) is satisfied.

The fifth axiom is ensured by the conversion algorithm since each material is added to the model through an operating unit (except raw materials and the products but they are also connected to the proper operating units). Thus, there is no isolated vertex in the solution structure which represents a material. An isolated vertex would mean that there is a place which could not be approached by a vehicle, or a demand which could not be satisfied by a vehicle, etc. (and it would violate the mass balance constraints imposed on the system by eqns (3.8.2)-(3.8.4)). □

Corollary 5.4.4 *The mathematical programming model generated by the P-graph framework contains the global optimum of the capacitated vehicle routing problem.*

5.5 Relationship with other models

Those familiar with VRP models may recognize some similarities between the P-graph model and two commonly used formalizations. In this section, some of these similarities and differences are discussed.

5.5.1 Vehicle flow models

Two-index vehicle flow models

Basic versions of the VRP are often modeled by two-index vehicle flow models. Such formulations use $O(n^2)$ binary variables to indicate if a vehicle traverses an arc in the optimal solution. A variable x_{ij} takes value 1 if $(i, j) \in \mathscr{A}$ belongs to the optimal solution and takes 0 otherwise.

Two-index vehicle flow models can be used only when the cost of the solution can be expressed as the sum of the costs associated with the traversed arcs. Moreover, it is not known, which vehicle traverses an arc in the solution.

Three-index vehicle flow models

Three-index vehicle flow models overcome some of the drawback of the two-index models by explicitly indicating the vehicle that traverses an arc. These models use $O(n^2l)$ binary variables x and $O(nl)$ binary variables y. A variable x_{ijk} counts the number of times arc $(i, j) \in \mathscr{A}$ is traversed by vehicle k in the optimal solution and variable y_{ik} takes value 1 if customer i is served by vehicle k in the optimal solution and takes 0 otherwise.

Three-index vehicle flow models generalize the two-index models and they have been extensively used to model more constrained versions of the VRP. Their main drawback is the increased number of binary variables.

5.5.2 Commodity flow models

The commodity flow models extend the two-index vehicle flow formulations by a new set of continous variables associated with the arcs, representing the amounts of demand flowing among them. The formulation extends the graph by a copy of the depot denoted by vertex $n + 1$. For any route of a feasible solution, two directed paths are defined by the flow variables, one from vertex 0 to $n + 1$, whose variables represent the vehicle load and another from vertex $n + 1$ to vertex 0, whose variables represent the residual capacity on the vehicle.

5.5.3 The P-graph model

In the P-graph model a binary variable is associated with an operating unit only if the operating unit has a fix cost or a lower bound. Thus, like the three-index vehicle flow models, the P-graph model uses $O(n^2l)$ binary variables (associated with the operating units modeling the movements of the vehicles) but there is also a set of continous variables (associated with the operating units modeling the movements of the goods) which is similar to the commodity flow models.

Note, that if the mutually exclusive operating units are implemented with additional operating units, the model is extended with a new set of binary variables (exactly $n * l$, where n is the number of places and l is the number of vehicles), which is essentially equivalent to the set of y variables used in the three-index vehicle flow models. However, if the mutually exclusive operating units are handled algorithmically by the framework, then this set of binary variables can be omitted.

A two-index vehicle flow formalization may require less binary variable to model the CVRP, but in the P-graph model it is known which vehicle traverses which road section and this is essential to generate alternative solutions.

Also note, that in contrast with the commodity flow models, in the P-graph model there is no copy of the depot and thus the flow of the goods is defined between the customers and the depot.

5.6 Summary

In this chapter I have introduced an algorithm which generates the P-graph model of a capacitated vehicle routing problem. I have also proven the validity of the generated mathematical model and I have shown that it contains the global optimum.

The P-graph model of the CVRP provides a good basis for further developments. With the current implementation of the loading and unloading operations it is straightforward to model pickup and delivery problems. As a recent result, the P-graph framework was extended with time constraints [71]. The inclusion of these constraints into the model may yield the P-graph model of the VRPTW problem.

The initial benchmark results show that the model is more competitive than the solver which implements the algorithms of the P-graph framework. This provides a good basis for future solver development. The benchmark results of CPLEX and CBC show that the potential is there which could be exploited by adjusting the PNS solver to the characteristics of the CVRP. This could be integrated with the already existing graphical editor of the P-graph framework to yield a complete software framework for the CVRP which has a competitive solver, an easy to use graphical editor and strong theoretical background.

5.7 Related publications

1. M. Bárány, témavezető: Dr. Bertók Botond. Keretrendszer fuvarszervezés és analóg feladatok megoldására a P-gráf módszertan alkalmazásával, Intézményi Tudományos Diákköri Konferencia, 2009. Pannon Egyetem (I. prize)

2. M. Bárány, témavezető: Dr. Bertók Botond. Keretrendszer fuvarszervezés és analóg feladatok megoldására a P-gráf módszertan alkalmazásával, XXX. Országos Tudományos Diákköri Konferencia, 2011. (II. prize)

Chapter 6

New results in reaction-pathway identification

A reaction-pathway identification procedure has two distinct phases. The first phase enumerates exhaustively the feasible candidate pathways, and the second phase identifies the ultimate feasible pathway or pathways among them. Probably the most efficient way to execute the first phase is to algorithmically generate the networks of feasible candidate pathways from a predefined set of plausible elementary reactions.

Although the first phase of reaction-pathway identification is not as popular as the second, still, several methods have been proposed to solve it. This chapter explores the relationship between the terms direct mechanism, extreme pathway and structurally minimal pathway. While these methods have different theoretical backgrounds, applying linear algebra, convex analysis and graph theory respectively, all of them consider any chemical reaction system as a network of elementary reactions.

6.1 Direct mechanisms

This chapter briefly introduces the linear algebraic method developed by Happel and Sellers [63]. Their work is based on that they viewed a chemical system as a network of elementary reactions linked to one another by common reactants. To determine the possible network configurations of a chemical system, they have reinterpreted this task mathematically and were looking for solution of the following problem: Given a finite list of elementary reactions, determine all the possible combinations of the given reactions that form a specified overall reaction.

6.1.1 Theoretical background

To answer this question from a mathematical viewpoint, it is necessary to define two vector spaces, an s-dimensional space of chemical reaction mechanisms and a q-dimensional space of chemical reactions. These two vector spaces are related to each other since each mechanism $\mathbf{m}$ gives rise to a unique reaction $f^R(\mathbf{m})$, wherein f^R is a function transforming the mechanism into the reaction. f^R is linear: the reactions in a chemical reaction system are additive, and thus, the reaction associated with combined mechanisms $\mathbf{m}_1$+$\mathbf{m}_2$ is $f^R(\mathbf{m}_1)$+$f^R(\mathbf{m}_2)$.

Steps

The simplest kind of mechanism ideally consists of a one-step molecular interaction and is termed step. Any mechanism is a combination of such steps. Each of the steps produces one of the elementary reactions forming a basis for the space of all reactions involved in the chemical reaction system. For example, let $\mathbf{s}_1$ be a step which yields a_3 from a_1 and a_2 and let $\mathbf{s}_2$ be a step which transforms a_3 into a_4. Then $f^R(\mathbf{s}_1)$ is the vector $-a_1 - a_2 + a_3$, $f^R(\mathbf{s}_2)$ is the vector $-a_3 + a_4$ and $f^R(\mathbf{s}_1 + \mathbf{s}_2)$ is $-a_1 - a_2 + a_4$.

σ denotes the rate of reaction. If a step $\mathbf{s}$ is repeated σ times, then the linear equation becomes $f^R(\sigma\mathbf{s}) = \sigma f^R(\mathbf{s})$. σ can be a negative value, expressing the possibility of a reverse reaction.

The stoichiometric matrix

Denote the species contained in the chemical reaction system by $a_1, a_2, ..., a_a$. The elementary reactions among these species are denoted by the **r** vectors in eqn (6.1.1)

$$\begin{array}{lcccccccc} \mathbf{r}_1 & = & \gamma_{11}a_1 & + & \gamma_{12}a_2 & + & \ldots & + & \gamma_{1a}a_a \\ \mathbf{r}_2 & = & \gamma_{21}a_1 & + & \gamma_{22}a_2 & + & \ldots & + & \gamma_{2a}a_a \\ \vdots & & \vdots & & & & & & \vdots \\ \mathbf{r}_s & = & \gamma_{s1}a_1 & + & \gamma_{s2}a_2 & + & \ldots & + & \gamma_{sa}a_a \end{array} \tag{6.1.1}$$

where the γ's are stoichiometric coefficients. Usually, each elementary reaction has one or two positive coefficients, one or two negative coefficients and the remainder are equal to zero. An elementary reaction may have more nonzero coefficients but it is assumed, that it has at least one negative and at least one positive coefficient.

The elementary reactions in eqn (6.1.1) may be linearly dependent. The maximum number of linearly independent reaction vectors in a linearly independent subset is denoted by q. This subset provides a basis for a q-dimensional vector space, termed the reaction space. In other words, q is the rank of the $s \times a$ matrix of stoichiometric coefficients in eqn (6.1.2).

$$\begin{bmatrix} \gamma_{11} & \gamma_{12} & \cdots & \gamma_{1a} \\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2a} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{s1} & \cdots & \cdots & \gamma_{sa} \end{bmatrix} \tag{6.1.2}$$

Mechanisms and reactions

Step $\mathbf{s}_i$ denotes the molecular interaction which produces $\mathbf{r}_i$ or $f^R(\mathbf{s}_i)$. Let mechanism **m** be any linear combination of steps in the form

$$\mathbf{m} = \sigma_1\mathbf{s}_1 + \sigma_2\mathbf{s}_2 + \ldots + \sigma_s\mathbf{s}_s \tag{6.1.3}$$

where the coefficients, σ_i, are real numbers signifying the rate of occurence of $\mathbf{s}_i$. The s-dimensional vector space comprising the set of all such mechanisms is called the mechanism space. The reaction, **r**, corresponding to the mechanism, **m**, can be obtained by applying the linear function, f^R, to the above equation, thereby resulting in

$$\mathbf{r} = \sigma_1\mathbf{r}_1 + \sigma_2\mathbf{r}_2 + \ldots + \sigma_s\mathbf{r}_s \tag{6.1.4}$$

The equations of eqn (6.1.1) can be substituted into this expression, leading to the following explicit linear combination:

$$\mathbf{r} = f^R(\mathbf{m}) = (\sum_{i=1}^{s}\sigma_i\gamma_{i1})a_1 + (\sum_{i=1}^{s}\sigma_i\gamma_{i2})a_2 + \ldots + (\sum_{i=1}^{s}\sigma_i\gamma_{ia})a_a \tag{6.1.5}$$

which can be expressed in matrix form also:

$$\mathbf{r} = f^R(\mathbf{m}) = [\sigma_1\ \sigma_2\ \ldots\ \sigma_s]\begin{bmatrix}\gamma_{11} & \gamma_{12} & \cdots & \gamma_{1a}\\ \gamma_{21} & \gamma_{22} & \cdots & \gamma_{2a}\\ \vdots & \vdots & \ddots & \vdots\\ \gamma_{s1} & \cdots & \cdots & \gamma_{sa}\end{bmatrix}\begin{bmatrix}a_1\\ a_2\\ \vdots\\ a_a\end{bmatrix} \tag{6.1.6}$$

The steady-state of a chemical reaction system

Species in a chemical reaction system can be grouped into two classes. One comprises terminal species including starting reactants and final products. The other comprises intermediates that do not belong to the terminal species. In a steady-state mechanism, the concentrations of all intermediates are presumed to be constant, thus implying that the net rate of production of every intermediate is zero.

Denote the intermediate species by $a_1, a_2, \ldots, a_I$, and the terminal species, by $a_{I+1}, a_{I+2}, \ldots, a_{I+t}$ where $I + t = a$. The first i coefficients in the right-hand side of eqn (6.1.5) will be zero. Horiuti has introduced [68] a characterization for a steady-state mechanism as one whose coefficients $\mu_1, \mu_2, \ldots, \mu_s$ satisfy the I linear equations:

$$[\mu_1\ \mu_2\ \ldots\ \mu_s]\begin{bmatrix}\gamma_{11} & \cdots & \gamma_{1I}\\ \vdots & \ddots & \vdots\\ \gamma_{s1} & \cdots & \gamma_{sI}\end{bmatrix} = [0\ 0\ \ldots\ 0] \tag{6.1.7}$$

If the rank of the $s \times I$ matrix in the above equation is denoted by h, the dimension of the space of all steady-state mechanisms, p, is equal to $s - h$ and the dimension r of the space of all reactions which they produce equals $q - h$. The reactions in the

r-dimensional space are overall reactions, and as such, they involve terminal species only.

Characterizing a steady-state system by linear algebraic bases

The values of s, h, r, q and p and the relations among them can be determined by simply considering them as dimensions of vector spaces and resorting to well-known linear algebraic concepts like basis and the linear independence of vectors. In linear algebra, a basis for a vector space is a sequence of vectors that form a set that is linearly independent and spans the space.

Since the dimension of a space is equal to the number of elements in the basis, every steady-state mechanism can uniquely be expressed in terms of p steady-state mechanisms. While this approach uniquely represents each steady-state mechanism, it does not provide a valid classification from a chemical point of view since the choice of basis is arbitrary and is not dictated, in general, by any consideration of chemistry.

Characterizing a steady-state system by direct mechanisms

There exists, however, a unique collection of mechanisms in every chemical reaction system, called direct mechanisms, which is fundamental constitutent of any mechanism. They are also known as "direct paths" [95] or "cycle-free mechanisms" [115, 116]. Let $\mathbf{m}$ be a mechanism and $\mathbf{r}$ be the reaction, which it produces. Mechanism $\mathbf{m}$ is defined as direct if it is minimal in the sense that, if one step is omitted then there is no mechanism for $\mathbf{r}$, which can be formed from any linear combination of the remaining steps.

In every chemical reaction system, the set of all direct mechanisms contains within a basis for the vector space of all mechanisms of the given system. Usually, there are more direct mechanisms than basis elements and thus there may exist linear dependence relations among direct mechanisms, but even then, they will differ chemically. Note, that while a linear algebraic basis for a system may be ambigous, the set of direct mechanisms is a uniquely defined attribute of the system.

Table 6.1: List of candidate elementary reactions

(1) $H_2 + \ell \rightleftharpoons H_2\ell$

(2) $H_2\ell + \ell \rightleftharpoons H\ell + H\ell$

(3) $N_2 + \ell \rightleftharpoons N_2\ell$

(4) $N_2\ell + \ell \rightleftharpoons N\ell + N\ell$

(5) $N_2\ell + H_2\ell \rightleftharpoons N_2H_2\ell + \ell$

(6) $N_2H_2\ell + \ell \rightleftharpoons NH\ell + NH\ell$

(7) $N\ell + H\ell \rightleftharpoons NH\ell + N\ell$

(8) $NH\ell + H\ell \rightleftharpoons NH_2\ell + \ell$

(9) $NH\ell + H_2\ell \rightleftharpoons NH_3\ell + \ell$

(10) $NH_2\ell + H\ell \rightleftharpoons NH_3\ell + \ell$

(11) $NH_3\ell \rightleftharpoons NH_3 + \ell$

Table 6.2: List of the identifiers of the species

Species	H_2	ℓ	$H_2\ell$	$H\ell$	N_2	$N_2\ell$	$N\ell$	$N_2H_2\ell$	$NH\ell$	$NH_2\ell$	$NH_3\ell$	NH_3
Notation	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}

6.1.2 Illustrative example

For those not familiar with reaction-pathway identification, the modeling procedure is illustrated through the exhaustively studied ammonia synthesis reaction. The overall reaction produces ammonia from hydrogen and nitrogen, i.e.,

$$N_2 + 3H_2 \rightleftharpoons 2NH_3 \tag{6.1.8}$$

The set of plausible elementary reactions is listed in 6.1 and taken from [32]. Tables 6.2 and 6.3 list the species and steps involved, respectively, as denoted by Happel and Sellers. The stoichiometric matrix in eqn (6.1.9) is constructed directly from Table 6.3.

Table 6.3: Table of stoichiometric coefficients γ_{ij}'s defining the steps

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}
s_1	-1	-1	1									
s_2			-1	2								
s_3		-1			-1	1						
s_4		-1				-1	2					
s_5		1	-1			-1		1				
s_6		-1						-1	2			
s_7		1		-1			-1		1			
s_8		1		-1					-1	1		
s_9		1	-1						-1		1	
s_{10}		1		-1						-1	1	
s_{11}		1									-1	1

$$\boldsymbol{\gamma} = \begin{bmatrix} -1 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix} \tag{6.1.9}$$

The set of direct mechanisms can be obtained by the proper solution algorithm. The resultant set of direct mechanisms is shown in Table 6.4. The coefficient matrix in eqn (6.1.10) is obtained directly from Table 6.4.

Table 6.4: List of direct mechanisms for the ammonia synthesis

	s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}
m_1	3	1	1	1			2		2		2
m_2	3	3	1	1			2	2		2	2
m_3	3		1	1			2	1	3	1	3
m_4	3		1		1	1			2		2
m_5	3	2	1		1	1		2		2	2
m_6	3		1	2	3	3	4	2		2	2

$$\boldsymbol{\sigma} = \begin{bmatrix} 3 & 1 & 1 & 1 & 0 & 0 & 2 & 0 & 2 & 0 & 2 \\ 3 & 3 & 1 & 1 & 0 & 0 & 2 & 2 & 0 & 2 & 2 \\ 3 & 0 & 1 & 1 & 0 & 0 & 2 & 1 & 3 & 1 & 3 \\ 3 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 2 & 0 & 2 \\ 3 & 2 & 1 & 1 & 1 & 1 & 0 & 2 & 0 & 2 & 2 \\ 3 & 0 & 1 & 2 & 3 & 3 & 4 & 2 & 0 & 2 & 2 \end{bmatrix} \tag{6.1.10}$$

6.2 Structurally minimal pathways

Reaction-pathway identification can be defined as a class of process-network synthesis problems where each species consists of a finite number of chemical elements in a fixed ratio, and these chemical elements are conserved throghout the process. The products of an elementary reaction comprise exactly the same chemical elements as the starting reactants, and the products of an overall reaction contain exactly the same components as the starting reactants. In the following, it will be shown, that a reaction-pathway identification problem can be interpreted as a process-network synthesis problem, and thus, it can be solved by the P-graph framework [16, 31].

6.2.1 Theoretical background

In this discussion, the reaction-pathway identification is defined by the quadruple $(\mathbf{E}, \mathcal{O}, \mathcal{M}, \mathcal{Q})$ where

- $\mathcal{Q} = \{q_1, q_2, \ldots, q_h\}$ is the finite ordered set of the components of the species

- $\mathcal{M} = \{a_1, a_2, \ldots, a_l\}$ is the finite ordered set of species each of which is defined by an $\mathbf{a}_j = [a_{1,j}, a_{2,j}, \ldots, a_{h,j}]^T \in (\mathbb{R}_0^+)^h$ vector of nonnegative numbers with $a_{k,j}$ denoting the quantity of component q_k in the species $\mathbf{a}_j$ $(k = 1, 2, \ldots, h)$
- $\mathbf{E}$ is an l-dimension vector of real numbers denoting the overall reaction, i.e., $\mathbf{E} = [E_1, E_2, \ldots, E_l]^T \in \mathbb{R}^l$, where E_j signifies the difference of the production and consumption rate of the species $\mathbf{a}_j$ $(j = 1, 2, \ldots, l)$ by the overall reaction. Species $\mathbf{a}_j$ is a starting reactant of the overall reaction if and only if $E_j < 0$, and it is a final product of the overall reaction if and only if $E_j > 0$
- $\mathcal{O} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_n\}$ is the finite ordered set of elementary reactions where every reaction $\mathbf{e}_i$ is represented by an l-dimensional vector of real numbers, $\mathbf{e}_i = [e_{1,i}, e_{2,i}, \ldots, e_{l,i}]^T \in \mathbb{R}^l$, where $e_{j,i}$ indicates the difference of the rate of consumption and production of the species, $\mathbf{a}_j$ $(j = 1, 2, \ldots, l)$, by elementary reaction i

Presumably,

$$\mathcal{Q} \cap \mathcal{M} = \mathcal{M} \cap \mathcal{O} = \mathcal{O} \cap \mathcal{Q} = \emptyset \text{ and } \mathbf{E} \notin \mathcal{Q} \cup \mathcal{M} \cup \mathcal{O}. \tag{6.2.1}$$

The P-graph representation

In case of reaction-pathway identification, the graphical representation of a P-graph is altered. Elementary reactions are depicted by horizontal bars, while chemical and active species by circles. If a chemical species is a reactant to an elementary reaction, the vertex representing the species is linked to the vertex representing the elementary reaction by an arc. Similarly, if a chemical species is a product from an elementary reaction, then the vertex representing the elementary reaction is linked by an arc to the vertex representing the species.

The set of chemical or active species is denoted by $m \subseteq \mathcal{M}$; and the set of elementary reactions under consideration, by $o \subseteq \mathcal{O}$, where

$$m = \{a_j \in (\mathbb{R}_0^+)^h : j = 1, 2, \ldots, l'\} \qquad l' \leq l \tag{6.2.2}$$

and

$$\mathscr{o} = \{\mathbf{e}_i \in \mathbb{R}^l : i = 1, 2, \ldots, n'\} \qquad n' \leq n \tag{6.2.3}$$

An $(\mathscr{m}, \mathscr{o})$ P-graph representing a reaction network can be defined formally as follows: The set of vertices is $\mathscr{m} \cup \mathscr{o}$, and the set of arcs is $\mathscr{A}_1 \cup \mathscr{A}_2$, where

$$\mathscr{A}_1 = \{(a_j, \mathbf{e}_i) : a_j \in \mathscr{m}, \mathbf{e}_i \in \mathscr{o}, \mathbf{e}_{j,i} < 0\} \tag{6.2.4}$$

and

$$A_2 = \{(\mathbf{e}_i, a_j) : \mathbf{e}_i \in \mathscr{o}, a_j \in \mathscr{m}, \mathbf{e}_{j,i} > 0\} \tag{6.2.5}$$

An $(\mathscr{m}', \mathscr{o}')$ P-graph is a subgraph of the $(\mathscr{m}'', \mathscr{o}'')$ P-graph, i.e., $(\mathscr{m}', \mathscr{o}') \subseteq (\mathscr{m}'', \mathscr{o}'')$, if $\mathscr{m}' \subseteq \mathscr{m}''$ and $\mathscr{o}' \subseteq \mathscr{o}''$. The union of P-graphs $(\mathscr{m}', \mathscr{o}')$ and $(\mathscr{m}'', \mathscr{o}'')$ is defined to be the P-graph $(\mathscr{m}' \cup \mathscr{m}'', \mathscr{o}' \cup \mathscr{o}'')$.

Axioms of combinatorially feasible pathways

The P-graph representing a reaction pathway leading from the starting reactants to the final products defined by the overall reaction is *combinatorially feasible*, if it satisfies the following axioms:

(T1) Every final product is represented in the network.

(T2) Every starting reactant is represented in the network.

(T3) Each reaction step represented in the network is defined a priori.

(T4) Every active species represented in the network has at least one path leading to the final product of the overall reaction.

(T5) Every chemical or active species represented in the network must be a reactant for or a product from at least one reaction step represented in the network.

(T6) A reactant of any elementary reaction represented in the reaction network is a starting reactant, if it is not produced by any reaction step represented in the network.

(T7) The network includes at most either the forward or the reverse step of each elementary reaction represented in the network.

It is easy to see the similarities between Axioms (T1)-(T7) and Axioms (S1)-(S5) from Section 3.5. (Note, that with the proper terminology, (S1) is equal to (T1), (S3) to (T3) and (S5) to (T5).)

Axioms of feasible pathways

Not every combinatorially feasible structure constitutes a *feasible pathway.* A feasible pathway is a pathway satisfying the following six axioms;

(R1) Every final product is totally produced by the reaction steps represented in the pathway.

(R2) Every starting reactant is totally consumed by the reaction steps represented in the pathway.

(R3) Every active intermediate produced by any reaction step represented in the pathway is totally consumed by one or more reaction steps in the pathway, and every active intermediate consumed by any reaction step represented in the pathway is totally produced by one or more reaction steps in the pathway.

(R4) All reaction steps represented in the pathway are defined a priori.

(R5) The network representing the pathway is acyclic.

(R6) At least one elementary-reaction step represented in the pathway activates a starting reactant.

Feasible pathways must satisfy stronger constraints then combinatorially feasible pathways, since Axioms (T1)-(T7) are the relaxations of Axioms (R1)-(R6). Consequently, every feasible pathway is a combinatorially feasible pathway, but not every combinatorially feasible pathway is a feasible pathway.

RPIMSG and RPISSG

Axioms (T1)-(T7) give rise to algorithms RPIMSG and RPISSG, which in the context of reaction-pathway identification do exactly the same as algorithms MSG and SSG. Like MSG, RPIMSG reduces the search space by excluding those elementary reactions which do not contribute to any combinatorially feasible solution structure. The generated structure is maximal in the sense that it contains every combinatorially feasible solution structure but Axiom (T7) is not closed under union and thus the maximal structure may not be a solution structure itself. Analog to SSG, RPISSG yields the set of all combinatorially feasible reaction networks from the maximal structure of reaction networks.

RPIPBT

The feasible pathways defined by Axioms (R1)-(R6) are generated by algorithm RPIPBT (Pathway-Back-Tracking). RPIPBT is a branch and bound like algorithm which similarly to RPISSG, generates combinatorially feasible pathways but unlike RPISSG, it also tests these pathways whether they satisfy Axioms (R1)-(R3) and (R5) or not.

(R5) says that a feasible pathway must not contain cycles. This is verified by the algorithm by a linear program, where the constraints reflect the condition that the null vector can be expressed by a nonnegative linear combination of the elementary reactions included in the structure. Axioms (R1)-(R3) are verified by another linear program, which tests whether the overall reaction can be expressed by a linear combination of the included and the “undecided” (at the current node of the search tree it is not yet decided whether they are included in the structure or not) elementary reactions, where the former have strictly positive and the latter have nonnegative coefficients.

Structurally minimal pathways

A P-graph is termed a *structurally minimal pathway* or “*independent pathway*” if it represents a feasible pathway and none of its proper subgraphs can represent a feasible pathway. For a more exact definition of structurally minimal pathways, it is necessary to formalize Axioms (R1)-(R5) ((R6) relaxed).

A set o of elementary reactions satisfies Axioms (R1)-(R3) if and only if there exists a positive coefficient λ_i for each elementary reaction $\mathbf{r}_i \in o$ such that

$$\exists \boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_n]^T : \sum_{\mathbf{r}_i \in o} \lambda_i \mathbf{r}_i = \mathbf{E}, \quad \mathbf{r}_i \in o \Longleftrightarrow \lambda_i > 0 \tag{6.2.6}$$

The above equation implies that the system must be at steady-state. Axiom (R4) can be formulated as

$$o \subseteq O \tag{6.2.7}$$

and finally, Axiom (R5) can be formulated as

$$\nexists o' : o' \subseteq o, o' \neq \emptyset, \exists \boldsymbol{\lambda}' = [\lambda'_1, \lambda'_2, \ldots, \lambda'_n]^T : \sum_{\mathbf{r}_i \in o} \mathbf{r}_i \lambda'_i = 0, \quad \mathbf{r}_i \in o' \Longleftrightarrow \lambda'_i > 0 \tag{6.2.8}$$

From a mathematical point of view, an (m, o) P-graph is termed structurally minimal pathway, if it satisfies the following statement: The overall reaction can be expressed as a positive linear combination of the elementary reactions included in the structure and (m, o) is minimal in the sense that it has no proper subgraphs satisfying this criterion. With a slight modification, RPIPBT is able to generate the complete set of structurally minimal pathways.

6.2.2 Illustrative example

The modeling procedure is illustrated through the previously introduced ammonia synthesis. Two chemical elements and an active site are identified in the system. Set Q contains two elements and the notation of the active site ℓ;

$$Q = \{q_1, q_2, q_3\} = \{\dot{N}, \dot{H}, \dot{\ell}\} \tag{6.2.9}$$

Set $\mathcal{M}$ contains twelve species;

$$\mathcal{M} = \{a_1, a_2, \ldots, a_{12}\} \tag{6.2.10}$$

In an equivalent yet more detailed formulation the species are represented in vectorial form where the components of the vectors denote the quantities of the chemical elements in the species; as expressed below.

$$\mathcal{M} = \{[0,2,0]^T, [0,0,1]^T, \ldots, [1,3,0]^T\} \tag{6.2.11}$$

Often the set of species are given by simply listing the species contained in the chemical reaction system omitting the notation $a_1, a_2, \ldots,\ a_{12}$, as follows:

$$\mathcal{M} = \{H_2, \ell, \ldots, NH_3\} \tag{6.2.12}$$

The following equation shows how the set of elementary reactions are given in the terminology of structurally minimal pathways. Note, that unlike in the case of steps, it is necessary to define the elementary reactions in both directions;

$$\mathcal{O} = \{e_{1\rightarrow}, e_{2\rightarrow}, \ldots, e_{11\rightarrow}, e_{1\leftarrow}, e_{2\leftarrow}, \ldots, e_{11\leftarrow}\} \tag{6.2.13}$$

A detailed form of this expression is given below where each elementary reaction step is expressed by the vector of its stoichiometric coefficients.

$$\begin{aligned} \mathcal{O} = \{e_{1\rightarrow} &= [-1,-1,1,0,0,0,0,0,0,0,0,0]^T, \\ e_{2\rightarrow} &= [0,0,-1,2,0,0,0,0,0,0,0,0]^T, \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad \ldots \\ e_{11\rightarrow} &= [0,1,0,0,0,0,0,0,0,0,-1,1]^T, \\ e_{1\leftarrow} &= [1,1,-1,0,0,0,0,0,0,0,0,0]^T, \\ e_{2\leftarrow} &= [0,0,1,-2,0,0,0,0,0,0,0,0]^T, \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad \ldots \\ e_{11\leftarrow} &= [0,-1,0,0,0,0,0,0,0,0,1,-1]^T\} \end{aligned} \tag{6.2.14}$$

Similarly, the following equation shows the vector representing the overall reaction;

$$\mathbf{E} = [-3,0,0,0,-2,0,0,0,0,0,0,0,2]^T \tag{6.2.15}$$

Finally listed below is the set $\mathcal{V}$ of structurally minimal pathways (m_1, o_1), (m_2, o_2), $\ldots$,(m_6, o_6) determined by algorithm RPIPBT.

$$
\begin{aligned}
\mathcal{V} = \{(m_1 = N_2, H_2, NH_3, \ell, H_2\ell, H\ell, N_2\ell, N\ell, NH\ell, NH_3\ell,& \\
o_1 = \{e_{1\rightarrow}, e_{2\rightarrow}, e_{3\rightarrow}, e_{4\rightarrow}, e_{7\rightarrow}, e_{9\rightarrow}, e_{11\rightarrow}\}),& \\
(m_2 = N_2, H_2, NH_3, \ell, H_2\ell, H\ell, N_2\ell, N\ell, NH\ell, NH_2\ell, NH_3\ell,& \\
o_2 = \{e_{1\rightarrow}, e_{2\rightarrow}, e_{3\rightarrow}, e_{4\rightarrow}, e_{7\rightarrow}, e_{8\rightarrow} e_{10\rightarrow}, e_{11\rightarrow}\}),& \\
(m_3 = N_2, H_2, NH_3, \ell, H_2\ell, N_2\ell, N_2H_2\ell, NH\ell, NH_3\ell,& \\
o_3 = \{e_{1\rightarrow}, e_{3\rightarrow}, e_{4\rightarrow}, e_{7\rightarrow}, e_{8\rightarrow}, e_{9\rightarrow}, e_{10\rightarrow}, e_{11\rightarrow}\},& \\
(m_4 = N_2, H_2, NH_3, \ell, H_2\ell, N_2\ell, N_2H_2\ell, NH\ell, NH_3\ell,& \\
o_4 = \{e_{1\rightarrow}, e_{3\rightarrow}, e_{5\rightarrow}, e_{6\rightarrow}, e_{9\rightarrow}, e_{11\rightarrow}\}),& \\
(m_5 = N_2, H_2, NH_3, \ell, H_2\ell, H\ell, N_2\ell, N_2H_2\ell, NH\ell, NH_2\ell, NH_3\ell,& \\
o_5 = \{e_{1\rightarrow}, e_{2\rightarrow}, e_{3\rightarrow}, e_{5\rightarrow}, e_{6\rightarrow}, e_{8\rightarrow}, e_{10\rightarrow} e_{11\rightarrow}\}),& \\
(m_6 = N_2, H_2, NH_3, \ell, H_2\ell, H\ell, N_2\ell, N\ell, N_2H_2\ell, NH\ell, NH_2\ell, NH_3\ell,& \\
o_6 = \{e_{1\rightarrow}, e_{3\rightarrow}, e_{4\rightarrow}, e_{5\rightarrow}, e_{6\rightarrow}, e_{7\rightarrow}, e_{8\rightarrow}, e_{10\rightarrow}, e_{11\rightarrow}\})\}&
\end{aligned}
\tag{6.2.16}
$$

6.3 Extreme pathways

Metabolic pathway is a central paradigm in biology. There have been earlier attempts in characterizing complex metabolic networks (see e.g. [23]) and the interest in metabolic pathway analysis was increased further by the human genom project. However, the genome-scale metabolic networks reconstructed from annotation of genome sequences demanded new network-based definitions of pathways to facilitate analysis of their capabilities and functions [102]. This has given rise to two popular concepts termed Elementary Flux Modes [114] and Extreme Pathways [111].

Both Elementary Flux Modes and Extreme Pathways apply convex analysis to generate a set of unique pathways of the metabolic system. Due to their similarities the relationship among them has been analysed and it has been found that the set of elementary flux modes is a superset of the set of extreme pathways. For further information about the relationship of extreme pathways and elementary flux modes see [103]. Here, the focus will be on extreme pathways.

6.3.1 Theoretical background

A cellular metabolic reaction network is a collection of enzymatic reactions and transport processes that serve to replenish and drain the relative amounts of certain metabolites. A system boundary can be drawn around all these types of physically occurring reactions, which constitute internal fluxes operating inside the network [111].

Internal and exchange fluxes

Some metabolites may be fully enclosed by these boundaries while others may enter or exit the system. The latter necessiates the introduction of exchange fluxes which can be thought as the inputs/outputs of the system. In general, the reactions are classified as internal or exchange reactions, based on whether they cross the system boundary, or not. Reversible internal reactions are considered as two reactions in opposite directions, thus internal fluxes are nonnegative. This is only for mathematical purposes and does not influence the interpretation of the biochemical network.

Exchange reactions may be reversible but a metabolite cannot participate in more than one exchange reaction. The activity of an exchange reaction is considered to be positive if the metabolite is exiting and negative if the metabolite is entering the system.

The stoichiometric matrix

Given m metabolites and n reactions, the system is represented by the stoichiometric matrix $\mathbf{S} \in \mathbb{R}^{m \times n}$. Internal fluxes are denoted by v_i for $i = 1, ..., n_I$ and exchange fluxes by b_i for $i = 1, ..., n_E$ where $n = n_I + n_E$. Note, that n_E can never be greater then m, since there can be at most one exchange flux per metabolite. As usual, an S_{ij} element of $\mathbf{S}$ represents the stoichiometric coefficient of metabolite i in reaction j, and v_j represents the flux through reaction j.

For practical purposes, the internal reactions are represented by the first (n_I) series of coloumns of $\mathbf{S}$ and the external reactions are represented by the rest (n_E coloumns). Vector $\mathbf{v} \in \mathbb{R}^n$ represents the relative fluxes through the reactions in the metabolic network and it is constructed in a similar manner like **S**.

Representation of the metabolic system

The pathway structure to be determined should be an invariant property of the network along with stoichiometry. Thus, it is reasonable to analyse the system under steady-state, where the material balances are

$$\mathbf{S}\mathbf{v} = \mathbf{0} \tag{6.3.1}$$

The equation system in eqn (6.3.1) is usually underdetermined since the number of reactions is typically greater than the number of metabolites. The null space corresponds to the set of all solutions and a set of basis vectors can be selected to describe the null space in eqn (6.3.1), where each basis vector corresponds to a steady-state pathway [112]. For a complete analysis, further constraints have to be introduced. Each internal flux must be nonnegative:

$$v_i \geq 0, \forall i \tag{6.3.2}$$

The constraint on an exchange flux b_j depends on the status of the corresponding metabolite. Based on the direction of the exchange flux, the lower bound lb_j and the upper bound ub_j are set to 0, $-\infty$ or ∞. If b_i is bidirectional, i.e., both a source and a sink is present for the metabolite, then lb_j is set to $-\infty$ and ub_j to ∞ leaving the exchange flux unconstrained. Formally:

$$lb_j \leq b_j \leq ub_j \tag{6.3.3}$$

Convex analysis

Through eqns (6.3.1)-(6.3.3) a metabolic system in steady-state is described as a system of linear equalites/inequalities which limits the use of traditional concepts of linear algebra and gives rise to convex analysis. The solution set for this system can be described geometrically as convex polyhedral cone emanating from the origin of the n-dimensional space. Within this cone lie all the possible steady-state solutions; it is called as the steady-state flux cone. Since the flux space represents the capabilities of the given metabolic network, it clearly defines what a network can and cannot do.

In convex analysis, the edges of the cone are half-lines emanating from the origin and are called extreme rays. These rays are generate the cone and are systematically independent since they cannot be decomposed into a non-trivial convex combination of any other vectors residing in the cone. In contrast to the basis concept of linear algebra, this minimal generating set is unique.

Extreme pathways

In the context of metabolic systems, the edges of the cone are termed extreme pathways [111] as each edge corresponds to a particular pathway which satisfies eqns (6.3.1)-(6.3.3). Every point within the cone (C) can be written as a convex combination of extreme pathways. By denoting the extreme pathways by $\mathbf{p}_i$ and the total number of extreme pathways to generate C by k, we have

$$C = \{\mathbf{v} : \mathbf{v} = \sum_{i=1}^{k} w_i \mathbf{p}_i, \quad w_i \geq 0, \quad \forall i\} \tag{6.3.4}$$

where w_i denotes the weight of the given pathway in a particular flux distribution. The set of extreme pathways is similar to a coordinate system which can be used to describe a position in the space.

Properties of extreme pathways

While every flux distribution $\mathbf{v}$ can be expressed as a non-negative linear combination of the extreme pathways, the decomposition of a steady-state flux vector into the corresponding extreme pathways is not necessarily unique (in contrast with linear algebra, where such a decomposition is unique even though the basis itself is non-unique). The unique representation of every point in the solution space can be guaranteed only by a basis of the solution space. However, the set of extreme pathways form a basis only if the number of pathways equals to the dimension of the null space.

A set of $\{\mathbf{p}_1, \mathbf{p}_2, \ldots \mathbf{p}_k\}$ extreme pathways is systematically independent if no pathway can be written as a non-trivial non-negative linear combination of the other pathways. Thus, a set of pathways may be systematically independent while simultaneously being a linearly dependent set. Since the number of pathways forming the edges of the steady-state flux cone is typically greater than the dimension of the null space, the set of extreme pathways is linearly dependent. However, when the functional aspects of a metabolic system is investigated, systemic independence should take priority over linear independence as it is a unique property of any system and its structural capabilities.

Classification of extreme pathways

Extreme pathways can be classified according to coefficients of the exchange fluxes: there are pathways for which all of the coefficients of the exchange fluxes are zero, and there are pathways in which there are non-zero coefficients for a set of exchange fluxes. Moreover, the pathways can be classified even further, if the primary and currency metabolites are distinguished in the metabolic system. In this case, a third class of pathways can be introduced for which all of the exchange fluxes for the primary metabolites equal zero, while non-zero values exist for the exchange fluxes of some of the currency metabolites [111]:

- **Type-I pathways.** The exchange fluxes of the primary metabolites are active. These pathways are major contributors to the decomposition of almost any steady-state flux distribution and thus are of major interest.

- **Type-II pathways.** Only the exchange fluxes on the currency metabolites are active. These pathways correspond to true futile cycles existing within the network which serve to dissipate energy or reductive power.

- **Type-III pathways.** All of the exchange fluxes are inactive. These pathways represent cycles that have no net overall effect on the functional capabilities on the network. In most of the cases, these pathways denote the result of the decomposition of a reversible reaction into two irreversible reactions.

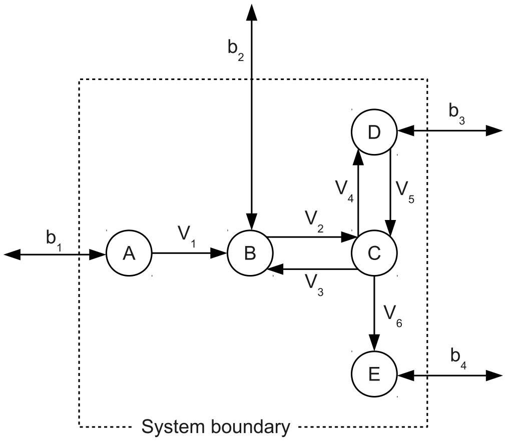


Figure 6.1: A sample metabolic network

6.3.2 Illustrative example

The modeling procedure is illustrated by an example taken from [111]. Fig. 6.1 depicts a simple biochemical network consisting of 5 metabolites, 6 internal and 4 exchange fluxes. All four exchange fluxes are unconstrained.

This network can formulated with respect to eqns (6.3.1)-(6.3.3) as

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.3.5}$$

with

$$v_j \geq 0, \quad j = 1 \ldots 6 \qquad -\infty \leq b_j \leq \infty, \quad j = 1, \ldots, 4. \tag{6.3.6}$$

The first six coloumns of **S** represent the internal reactions and the last four coloumns represent the internal reactions of the system. After executing a proper solution algorithm (see e.g. [111]), the following set of extreme pathways is obtained:

$$\mathbf{P} = \left[\begin{array}{ccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ \hline -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{array}\right] \begin{array}{l} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{array} \tag{6.3.7}$$

All seven pathways are presented as coloumns in the pathway matrix.

6.4 Equivalence of direct mechanisms and structurally minimal pathways

While direct mechanisms and structurally minimal pathways have a different theoretical background, there are also definite similarities between the two concepts. Essentially, both of them consider a chemical reaction system as a network comprising elementary chemical reactions linked to one another through shared reactants. A closer look reveals further similarities, like the existence of a steady-state constraint.

At this point the question may arise: is there any connection between direct mechanisms and structurally minimal pathways and if there is, what is it exactly? This question is valid indeed, and it will be shown that while these two concepts have different theoretical backgrounds (one rooted in linear algebra and the other in graph theory), they are actually equivalent.

First, it has to be shown that the overall reaction produced by a direct mechanism can be expressed by exactly one linear combination of the elementary reactions produced by the steps constituting the direct mechanism.

Theorem 6.4.1 *Let* $\mathbf{m}$ *be a direct mechanism yielding the overall reaction* $\mathbf{E}$*. Then, the overall reaction can be expressed by exactly one linear combination of the elementary reaction vectors,* $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n$*, generated by the steps* $\mathbf{s}_1$*,* $\mathbf{s}_2$*,* $\ldots$*,* $\mathbf{s}_n$ *constituting the direct mechanism,* $\mathbf{m}$*.*

Proof Let $\mathbf{m}$ be a direct mechanism yielding overall reaction $\mathbf{E}$. Let $\mathbf{s}_1$, $\mathbf{s}_2$, $\ldots$, $\mathbf{s}_n$ be the steps constituting the direct mechanism, $\mathbf{m}$, and let $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n$ be elementary reactions generated by the steps, $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n$. Then, $\mathbf{E}$ can be expressed as a linear combination of $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n$ with coefficients $\lambda_1, \lambda_2, \ldots, \lambda_n$:

$$\lambda_1\mathbf{r}_1 + \lambda_2\mathbf{r}_2 + \ldots + \lambda_n\mathbf{r}_n = \mathbf{E} \tag{6.4.1}$$

Suppose that the overall reaction can be written as another linear combination of these vectors with coefficients $\lambda_1^*, \lambda_2^*, \ldots, \lambda_n^*$, as

$$\lambda_1^*\mathbf{r}_1 + \lambda_2^*\mathbf{r}_2 + ... + \lambda_n^*\mathbf{r}_n = \mathbf{E} \tag{6.4.2}$$

and that there exists at least one such index k for which λ_k differs from λ_k^*, i.e.,

$$\exists k, \quad \lambda_k - \lambda_k^* \neq 0 \tag{6.4.3}$$

For $i = 1, 2, \ldots, n$, let ε_i denote the difference, $\lambda_i - \lambda_i^*$. Then, the linear combination of the elementary reaction vectors, $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n$, with coefficients $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$ is the null vector, i.e.,

$$\varepsilon_1\mathbf{r}_1 + \varepsilon_2\mathbf{r}_2 + \ldots + \varepsilon_n\mathbf{r}_n = \mathbf{E} - \mathbf{E} = 0 \tag{6.4.4}$$

This implies that the elementary reaction vectors are not linearly independent. Hence, one of the elementary reactions, $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_n$, can be expressed as a linear combination of the others. For example, for r_k where λ_k differs from λ_k^* (and thus ε_k is nonzero):

$$\varepsilon_k\mathbf{r}_k = -\sum_{\substack{i=1 \\ i\neq k}}^{n} \varepsilon_i\mathbf{r}_i \tag{6.4.5}$$

The above expression signifies that mechanism $\mathbf{m}$ is not a direct mechanism: The overall reaction $\mathbf{E}$ could be expressed as a linear combination of the reaction vectors $\mathbf{r}_1$, $\mathbf{r}_2$, $\ldots$, $\mathbf{r}_{k-1}$, $\mathbf{r}_{k+1}$, $\ldots$, $\mathbf{r}_n$ omitting reaction $\mathbf{r}_k$ as follows:

$$\lambda_1\mathbf{r}_1 + \lambda_2\mathbf{r}_2 + \ldots - \frac{\lambda_k}{\varepsilon_k}\sum_{\substack{i=1 \\ i\neq k}}^{n} \varepsilon_i\mathbf{r}_i + \ldots + \lambda_n\mathbf{r}_n = \mathbf{E} \tag{6.4.6}$$

This contradiction proves the theorem. □

Corollary 6.4.2 *To define a direct mechanism, it is sufficient to define the set of elementary reactions constituting the direct mechanism.*

This corollary implies that it is unnecessary to define the coefficients for the elementary reactions to uniquely define a direct mechanism; it is sufficient to define the set of elementary reactions constituting the direct mechanism. Corollary 6.4.2 provides the link between direct mechanisms and structurally minimal pathways. The combination of Corollary 6.4.2 and Theorem 6.4.3 prove that the direct mechanisms and the structurally minimal pathways of a chemical system are identical.

Theorem 6.4.3 *There is no proper subset of elementary reactions forming a structurally minimal pathway that can constitute a direct mechanism.*

Proof Suppose that for a vector $\mathbf{r}$ a given set of vectors $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_k, \mathbf{r}_{k+1}, \ldots, \mathbf{r}_n\}$ is minimal in the sense that the vector, $\mathbf{r}$, cannot be written as a linear combination of vectors of any proper subset of $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_k, \mathbf{r}_{k+1}, \ldots, \mathbf{r}_n\}$ with positive coefficients, i.e.,

$$\mathbf{r} = \sum_{i=1}^{n} \lambda_i \mathbf{r}_i, \qquad \forall \lambda_i > 0 \tag{6.4.7}$$

$\mathbf{r}$ is minimal in the sense that there is no proper subset of $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_k, \mathbf{r}_{k+1}, \ldots, \mathbf{r}_n\}$ satisfying eqn (6.2.8). Now suppose that by relaxing the positivity constraint and enabling negative coefficients, $\mathbf{r}$ can be written as a linear combination of the vectors in $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_k\}$, namely

$$\mathbf{r} = \sum_{i=1}^{k} \lambda_i^* \mathbf{r}_i \text{ where } k < n \tag{6.4.8}$$

Let δ be

$$\delta = \min_{\lambda_i^* < 0} \left\{ \frac{\lambda_i}{|\lambda_i^*|} \right\} \tag{6.4.9}$$

This δ is positive, since if $\lambda_i \geq 0$; then,

$$\frac{\lambda_i}{|\lambda_i^*|} \geq 0 \tag{6.4.10}$$

Multiplying both sides of eqn (6.4.8) by δ gives

$$\delta \mathbf{r} = \sum_{i=1}^{k} \delta \lambda_i^* \mathbf{r}_i \tag{6.4.11}$$

The sum of eqns (6.4.7) and (6.4.11) yields

$$\mathbf{r}(1+\delta) = \sum_{i=1}^{k} (\lambda_i + \delta \lambda_i^*) \mathbf{r}_i + \sum_{i=k+1}^{n} \lambda_i \mathbf{r}_i \tag{6.4.12}$$

In light of the definition of δ, in the first sum, where the minimum is attained, one of the coefficients $(\lambda_i + \delta\lambda_i^*)$ will be zero. By denoting the index of this coefficient by j, we have

$$\delta = \frac{\lambda_j}{|\lambda_j^*|} \text{ and } \lambda_j + \frac{\lambda_j}{|\lambda_j^*|}\lambda_j^* = 0 \tag{6.4.13}$$

The other coefficients will be nonnegative: it is trivial that

$$\lambda_i + \frac{\lambda_j}{|\lambda_j^*|}\lambda_i^* \geq 0 \text{ where } \lambda_i^* \geq 0 \tag{6.4.14}$$

and

$$\lambda_i + \frac{\lambda_i}{|\lambda_i^*|}\lambda_i^* = 0 \text{ where } \lambda_i^* < 0 \tag{6.4.15}$$

Since δ is defined where the minimum is attained, it is clear, that

$$0 < \delta = \frac{\lambda_j}{|\lambda_j^*|} \leq \frac{\lambda_i}{|\lambda_i^*|} \qquad \forall i, i \neq j \tag{6.4.16}$$

thereby yielding

$$\lambda_i + \frac{\lambda_j}{|\lambda_j^*|}\lambda_i^* \geq 0 \qquad \forall i, i \neq j \tag{6.4.17}$$

Thus, the vector, $\mathbf{r}$, can be written as a linear combination of the vectors of this proper subset with positive coefficients; moreover, by dividing both sides with $(1+\delta)$, a proper subset of the set, $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_k, \mathbf{r}_{k+1}, \ldots, \mathbf{r}_n\}$, is obtained, as given below.

$$\mathbf{r}(1+\delta) = \sum_{\substack{i=1 \\ i\neq j}}^{k} (\lambda_i + \delta \lambda_i^*) \mathbf{r}_i + \sum_{i=k+1}^{n} \lambda_i \mathbf{r}_i \tag{6.4.18}$$

$$\mathbf{r} = \frac{\sum_{\substack{i=1 \\ i\neq j}}^{k} (\lambda + \delta\lambda_i^*)\mathbf{r}_i + \sum_{i=k+1}^{n} \lambda_i \mathbf{r}_i}{1+\delta} \tag{6.4.19}$$

This contradicts that $\{\mathbf{r}_1, \mathbf{r}_2, \ldots \mathbf{r}_n\}$ is structurally minimal and proves the following theorem. □

Theorem 6.4.4 *Given a chemical system at steady-state, the set of direct mechanisms and the set of structurally minimal pathways of the system are equivalent.*

Proof Follows directly from Corollary 6.4.2 and Theorem 6.4.3. □

6.5 Equivalence of extreme pathways and structurally minimal pathways

There is no need for detailed analysis to see the similarities between extreme pathways and structurally minimal pathways, or to be more exact, the similarities between Type-I extreme pathways and structurally minimal pathways. (As a matter of fact, this whole chapter is based on the recognition that during the tests of the software implementing the P-graph framework reaction-pathway identification algorithms, the number of extreme pathways and structurally minimal pathways were always the same. However, the exact relationship was unclear at the time.) Structurally minimal pathways lead from input/starting metabolites to output/product metabolites while Type-I extreme pathways involve the conversion of primary inputs into primary outputs. In this section, it will be proven, that for a given overall reaction, the Type-I extreme pathways and structurally minimal pathways are equivalent. The following was proven in [111]:

Theorem 6.5.1 *A convex flux cone determined by eqns (6.3.1)-(6.3.3) has a set of systematically independent generating vectors. Furthermore, these generating vectors (extremal rays) are unique up to a multiplication by a positive scalar. The generating vectors are called extreme pathways.*

First, it will be shown, that for a given overall reaction eqns (6.2.6)-(6.2.8) define the same cone as eqns (6.3.1)-(6.3.3). Note, that a reaction vector $\mathbf{o}_i = [o_{1,i}, o_{2,i}, \ldots, o_{m,i}]^T$ is a coloumn of the stoichiometric matrix, thus (6.2.6) can be reformulated as

$$\mathbf{S}'\boldsymbol{\lambda} = \mathbf{E}, \quad \lambda_i \geq 0 \quad i = 1, 2, \ldots, n \tag{6.5.1}$$

where $\mathbf{S}' \in \mathbb{R}^{m \times n}$, $\boldsymbol{\lambda} \in \mathbb{R}^n$ and $\mathbf{E} \in \mathbb{R}^m$. The difference between the matrices $\mathbf{S}$ and $\mathbf{S}'$ and the vectors $\mathbf{v}$ and $\boldsymbol{\lambda}$ is the absence of the exchange flux components. Thus, the column dimension of $\mathbf{S}$ and the dimension of $\mathbf{v}$ is always greater than the dimension of $\mathbf{S}'$ and $\boldsymbol{\lambda}$ respectively.

Only one exchange flux is permitted per metabolite. Thus, the model defined by eqns (6.3.1)-(6.3.3) can be "generalized" in the following way: append an $m \times m$ diagonal matrix denoted by $\mathbf{I}'$ to the first n columns of $\mathbf{S}$. If a metabolite has an exchange flux assigned to it then $I'_{m,m}$ is -1 and 0 otherwise. $\mathbf{v}$ is expanded accordingly. Now $\mathbf{S} \in \mathbb{R}^{m \times (n+m)}$ and $\mathbf{v} \in \mathbb{R}^{(n+m)}$. The "generalized" form of eqn (6.3.5) is eqn (6.5.2).

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ b_1 \\ b_2 \\ b_e \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.5.2}$$

The $v_j \geq 0$ $(j = 1 \ldots n)$ inequalities remain valid, but for a given overall reaction, the b_j $(j = 1, \ldots, m)$ values are known. Furthermore, note that the b_j components of $\mathbf{v}$ constitute $\mathbf{E}$, the vector of the overall reaction from eqn (6.5.1), while the v_j components, the fluxes through the reactions are the same as the λ_j components of the $\boldsymbol{\lambda}$ vector from eqn (6.5.1). Thus, eqn (6.5.2) has the following structure;

$$[\mathbf{S}'|\mathbf{I}'] \begin{bmatrix} \boldsymbol{\lambda} \\ \mathbf{E} \end{bmatrix} = \mathbf{0} \tag{6.5.3}$$

The result of this matrix-vector multiplication can be formulated as

$$\mathbf{S}'\boldsymbol{\lambda} + \mathbf{I}'\mathbf{E} = \mathbf{0} \tag{6.5.4}$$

Since $\mathbf{I}'$ is diagonal and has only -1 as nonzero elements, eqn (6.5.4) becomes

$$\mathbf{S}'\boldsymbol{\lambda} - \mathbf{E} = \mathbf{0} \tag{6.5.5}$$

which equals to eqn (6.5.1). Thus, the convex cone determined by eqns (6.2.6)-(6.2.8) is equivalent to the cone determined by eqns (6.3.1)-(6.3.3).

Now it has to be shown, that like the extreme pathways, the structurally minimal pathways are also the edges of the cone, i.e., they have to be systematically independent and they have to generate the cone. The property of systematic independence follows from the definition of structurally minimal pathways, i.e., a vector corresponding to a structurally minimal pathway cannot be written as a positive linear combination of other structurally minimal pathways. The second property is proven indirectly.

In the previous section, it is proven by Theorem 6.4.4 that structurally minimal pathways are equivalent to direct mechanisms. It is also known from the previous sections that the set of all direct mechanisms in a system contains a basis for the vector space of all mechanisms and unlike a linear algebraic basis, the set of direct mechanisms is a unique property of the system. It follows from the above that the set of structurally minimal pathways is unique and every reaction pathway can be expressed in terms of structurally minimal pathways (i.e., they generate the cone). Thus, the following has been proven:

Theorem 6.5.2 *The set of Type-I extreme pathways and structurally minimal pathways are equivalent for a given overall reaction.*

Corollary 6.5.3 *The set of Type-I extreme pathways and direct mechanisms are equivalent for a given overall reaction.*

6.6 Summary

I have proven that the terms direct mechanism, structurally minimal pathway and Type-I extreme pathway are equivalent. Direct mechanisms were used mainly to analyse catalytic reactions [63], extreme pathways for metabolic pathway analysis [104, 107, 113, 133] and structurally minimal pathways for both [85, 88, 89, 118].

The latter suggests that the theoretical models developed for enzymatic and catalytic reactions are similar. Now, I have formally proven, that direct mechanisms and Type-I extreme pathways are indeed interchangeable. Thus, the algorithms and modeling methods developed for direct mechanisms are valid for structurally minimal pathways and extreme pathways and this holds vice versa as well.

This has certain benefits. For example, the algorithm computing extreme pathways is a fairly sequential one, since it computes a sequence of tableux from the initial tableu, the stoichiometric matrix, and each tableu is computed from the previous one [12]. This gives only a little room for parallelization. However, the algorithm used to determine structurally minimal pathways is based on the branch and bound principle and is highly parallelizable. Modern computer architectures typically increase their performance by simultaneously using multiple processors and RPIPBT is well-suited for this approach.

There are certain similarities between the methodologies that could be exploited further with the integration of the graph-theoretical and linear algebraic concepts. Such a method is already under development by the supervisor of the author and hopefully it will be published soon.

6.7 Related publications

Refereed Journal Papers

1. M. Barany, B. Bertok, C. Imreh, L. T. Fan, and F. Friedler. On the equivalence of direct mechanisms and structurally minimal pathways, Journal of Mathematical Chemistry, 50:1347-1361, 2012. (IF=1.303)

2. M. Barany, B. Bertok, L. T. Fan, and F. Friedler. Relationship between extreme pathways and structurally minimal pathways, Bioprocess and Biosystems Engineering, 36:1199-1203, 2013. (IF=1.869)

Chapter 7

Summary

The second chapter has introduced four problems often arising in industrial applications. These problems are generally considered to be hard but are also important enough to have a rich literature which has been surveyed by chapter two.

Chapter three reviewed the P-graph framework which can be considered as the common denominator of the subsequent chapters. The P-graph framework algorithmically generates the mathematical programming model of a process-network synthesis problem directly from the input parameters. It is formally proven that the generated model contains the global optimum.

The first new result is introduced in chapter four. Here, I have shown that a vehicle scheduling problem can be modeled by the P-graph framework. I have proven that the optimal solution is embedded in the generated model. Moreover, I have proposed an algorithm which generates the P-graph model of a vehicle scheduling problem in polynomial time.

In chapter five I have shown that it is also possible to model a CVRP problem by the P-graph framework. I have proven, that the generated model contains the global optimum of the problem. As in the previous chapter, I have also proposed an algorithm which generates the P-graph model of a capacitated vehicle routing problem.

Reaction-pathway identification problems were solved earlier by the P-graph framework. In chapter six I have expanded these results further by showing and proving that the concepts direct path, extreme pathway and structurally minimal pathway are equivalent.

At the time this thesis is written the new results in chapter four and six have been already published in peer reviewed scientific journals. A paper based on chapter five is under preparation.

Future work

The new scientific results of the thesis lay the foundations for the following future developments:

- **Multi-objective optimization by the P-graph framework.** The proposed model highlights through the CO_2 emission that it is easy to incorporate additional factors into the model which lays the foundations for the multi-objective optimization by the P-graph framework.

- **Modeling more complex vehicle routing problems.** The P-graph model of the CVRP provides a good basis for further developments. With the current implementation of the loading and unloading operations it is straightforward to model pickup and delivery problems. As a recent result, the P-graph framework was extended with time constraints. The inclusion of these constraints into the model may yield the P-graph model of the VRPTW problem.

- **Developing synergistyc methods forreaction-pathway identification.** Based on the achieved results it is possible to develop synergistic methods by integrating the graph-theoretical and linear algebraic concepts. Already under development.

Chapter 8

New scientific results

1. I have shown that vehicle scheduling problems can be modeled by the P-graph framework.

   (a) I have proven that the solution of a vehicle scheduling problem is also a solution to the synthesis problem modeling the vehicle scheduling problem. I have also proven that there is no such solution to the synthesis problem modeling the vehicle scheduling problem which is not a solution to the vehicle scheduling problem.

   (b) I have proposed an algorithm which systematically generates the P-graph model of the vehicle scheduling problem in polynomial time.

   (c) I have proven that the generated model contains the global optimum of the problem.

2. I have shown that vehicle routing problems can be modeled by the P-graph framework.

   (a) I have proven that the solution of a capacitated vehicle routing problem is also a solution to the synthesis problem modeling the capacitated vehicle routing problem. I have also proven that there is no such solution to the synthesis problem modeling the capacitated vehicle routing problem which is not a solution to the vehicle capacitated vehicle routing problem.

(b) I have proposed an algorithm which systematically generates the P-graph model of a capacitated vehicle routing problem in polynomial time.

(c) I have proven that the generated model contains the global optimum of the problem.

3. I have shown that the concepts direct path and structurally minimal pathway used in reaction pathway identification are equivalent.

   (a) I have formally proven that an overall reaction can be expressed by exactly one linear combination of the elementary reaction vectors generated by the steps constituting a direct mechanism.

   (b) I have formally proven that there is no proper subset of elementary reactions forming a structurally minimal pathway that can constitute a direct mechanism.

4. I have shown that the concepts extreme pathway and structurally minimal pathway used in reaction pathway identification are equivalent.

   (a) I have formally proven that the set of Type-I extreme pathways and structurally minimal pathways are equivalent for a given overall reaction.

   (b) As a consequence I have proven that the set of Type-I extreme pathways and direct mechanisms are equivalent for a given overall reaction.

Chapter 9

Publications

Refereed Journal Papers

1. M. Barany, B. Bertok, Z. Kovacs, F. Friedler and L.T. Fan, Optimization Software for Solving Vehicle Assignment Problems to Minimize Cost and Environmental Impact of Transportation, Chemical Engineering Transactions, 21:499-504, 2010.

2. M. Barany, B. Bertok, Z. Kovacs, F. Friedler and L.T. Fan, Solving Vehicle Assignment Problems by Process-network Synthesis to Minimize Cost and Environmental Impact of Transportation, Clean Technologies and Environmental Policy, 13:637-642, 2011. (IF=1.753)

3. M. Barany, B. Bertok, C. Imreh, L. T. Fan, and F. Friedler. On the equivalence of direct mechanisms and structurally minimal pathways, Journal of Mathematical Chemistry, 50:1347-1361, 2012. (IF=1.303)

4. M. Barany, B. Bertok, L. T. Fan, and F. Friedler. Relationship between extreme pathways and structurally minimal pathways, Bioprocess and Biosystems Engineering, 36:1199-1203, 2013. (IF=1.869)

5. B. Bertok, M. Barany and F. Friedler. Generating and Analyzing Mathematical Programming Models of Conceptual Process Design by P-graph Software,

Industrial & Engineering Chemistry Research, 52:166-171, 2013. (IF=2.235)

International conference presentation

1. M. Barany, B. Bertok, Z. Kovacs and F. Friedler, Optimization software for solving vehicle assignment problems to minimize cost and environmental impact of transportation, presented at the PRES 2010 (11th Conference on Process Integration, Modelling and Optimisation for Energy Saving and Pollution Reduction), Praha, Czech Republic, August 28-September 1, 2010.

Bibliography

[1] `https://www.informs.org/Sites/Getting-Started-With-Analytics/Analytics-Success-Stories/`, [Online; accessed 20-July-2015].

[2] R. Adonyi, B. Bertok, and F. Friedler, *Optimization of distribution networks*, PRES 2008, August 2008.

[3] E. Ahmetovic and Z. Kravanja, *Simultaneous synthesis of process water and heat exchanger networks*, Energy **57** (2013), 236–250.

[4] I.P. Androulakis and V. Venkatasubramanian, *A genetic algorithmic framework for process design and optimization*, Energy **15** (1991), 217–228.

[5] R. Angira and B.V. Babu, *Optimization of process synthesis and design problems: A modified differential evolution approach*, Chemical Engineering Science **61** (2006), 4707–4721.

[6] B. M. Baker and M.A. Ayechew, *A genetic algorithm for the vehicle routing problem*, Computers & Operations Research **30** (2003), 787–800.

[7] R. Baldacci, A. Mingozzi, and R. Roberti, *New route relaxation and pricing strategies for the vehicle routing problem*, Operations research **59** (2011), 1269–1283.

[8] M. Barany, *Framework for solving vrp and analog problems with the p-graph framework*, April 2011.

[9] M. Barany, B. Bertok, Z. Kovacs, and F. Friedler, *Optimization software for solving vehicle assignment problems to minimize cost and environmental impact of transportation*, PRES 2010, August 28-September 1 2010.

[10] M. Barany, B. Bertok, Z. Kovacs, F. Friedler, and L.T. Fan, *Solving vehicle assignment problems by process-network synthesis to minimize cost and environmental impact of transportation*, Clean Technologies and Environmental Policy **13** (2011), 637.

[11] J. E. Bell and P. R. McMullen, *Ant colony optimization techniques for the vehicle routing problem*, Advanced Engineering Informatics **18** (2004), 41–48.

[12] S. L. Bell and B. O. Palsson, *Expa: a program for calculating extreme pathways in biochemical reaction networks*, Bioinformatics **21** (2005), 1739–1740.

[13] B. Bertok, M. Barany, and F. Friedler, *Generating and analyzing mathematical programming models of conceptual process design by p-graph software*, Industrial & Engineering Chemistry Research **52** (2012), 166–171.

[14] B. Bertok, K. Kalauz, Z. Sule, and F. Friedler, *Combinatorial algorithm for synthesizing redundant structures to increase reliability of supply chains: Application to biodiesel supply*, Industrial & Engineering Chemistry Research **52** (2012), 181–186.

[15] A.A. Bertossi, P. Carraresi, and G. Gallo, *On some matching problems arising in vehicle scheduling models*, Networks **17** (1987), 271–281.

[16] B. Bertók, *Folyamathálózatok struktúráinak algoritmikus szintézise*, Ph.D. thesis, Doctoral School of Information Science and Technology , Veszprém University, 2003.

[17] L. Bianco, A. Mingozzi, and S. Ricciardelli, *A set partitioning approach to the multiple depot vehicle scheduling problem*, Optimization Methods and Software **3** (1994), 163–194.

[18] M. Boudart and G. Djega-Mariadassou, *Kinetics of heterogeneous catalytic reactions*, Princeton University Press, Princeton, N.J., 1984.

[19] S. Bunte and N. Kliewer, *An overview on vehicle scheduling models*, Public Transport **1** (2009), 299–317.

[20] G. Carpaneto, M. Dell'Amico, M. Fischetti, and P. Toth, *A branch and bound algorithm for the multiple depot vehicle scheduling problem*, Networks **19** (1989), 531–548.

[21] W.C. Chiang and R.A. Russell, *Simulated annealing metaheuristics for the vehicle routing problem with time windows*, Annals of Operations Research **63** (1996), 3–27.

[22] N. Christofides, A. Mingozzi, and P. Toth, *Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations*, Mathematical programming **20** (1981), 255–282.

[23] B. L. Clarke, *Stoichiometric network analysis*, Cell biophysics **12** (1988), 237–253.

[24] G. Clarke and J.W. Wright, *Scheduling of vehicles from a central depot to a number of delivery points*, Operations research **12** (1964), 568–581.

[25] G.B. Dantzig and J.H. Ramser, *The truck dispatching problem*, Management science **60** (1959), 80–91.

[26] G. Desaulniers, J. Lavigne, and F. Soumis, *Multi-depot vehicle scheduling problems with time windows and waiting costs*, European Journal of Operational Research **111** (1998), 479–494.

[27] W.B. Dolan, P. T. Cummings, and M.D. LeVan, *Process optimization via simulated annealing: Application to network design*, AIChE Journal **35** (1989), 725–736.

[28] H.G. Dong, C. Y. Lin, and C.T. Chang, *Simultaneous optimization approach for integrated water-allocation and heat-exchange networks*, Chemical Engineering Science **63** (2008), 3664–3678.

[29] J.M. Douglas, *A hierarchical decision procedure for process synthesis*, AIChE Journal **31** (1985), 353–362.

[30] S. Elion, C. Watson-Gandy, and N. Christofides, *Distribution management, mathematical modeling and practical analysis*, Griffin, London, 1971.

[31] L. T. Fan, B. Bertok, and F. Friedler, *A graph-theoretic method to identify candidate mechanisms for deriving the rate law of a catalytic reaction*, Computers & chemistry **26** (2002), 265–292.

[32] L. T. Fan, B. Bertok, F. Friedler, and S. Shafie, *Mechanisms of ammonia-synthesis reaction revisited with the aid of a novel graph-theoretic method of determining candidate mechanisms in deriving the rate law of a catalytic reaction*, Hungarian Journal of Industrial Chemistry **29** (2001), 71–80.

[33] M. Feinberg, *Chemical reaction network structure and the stability of complex isothermal reactors, Part I*, Chemical Engineering Science **42** (1987), 2229–2268.

[34] ______, *Chemical reaction network structure and the stability of complex isothermal reactors, Part II*, Chemical Engineering Science **43** (1988), 1–25.

[35] M. Fischetti, A. Lodi, S. Martello, and P. Toth, *A polyhedral approach to simplified crew scheduling and vehicle scheduling problems*, Management Science **47** (2001), 833–850.

[36] I. Fishtik, C.A. Callaghan, and R. Datta, *Reaction route graphs. I. theory and algorithm*, Journal of Physical Chemistry B **108** (2004), 5671–5682.

[37] ______, *Reaction route graphs. II. examples of enzyme- and surface-catalyzed single overall reactions*, Journal of Physical Chemistry B **108** (2004), 5683–5697.

[38] ______, *Reaction route graphs. III. non-minimal kinetic mechanisms*, Journal of Physical Chemistry B **109** (2005), 2710–2722.

[39] E.S. Fraga and T.R.S. Matias, *Synthesis and optimization of a nonideal distillation system using a parallel genetic algorithm*, Computers & Chemical Engineering **20** (1996), S79–S84.

[40] E.S. Fraga and K.I.M. McKinnon, *The use of dynamic programming with parallel computers for process synthesis*, Computers & Chemical Engineering **18** (1994), 1–13.

[41] R. Freling, A. P. Wagelmans, and J. M. P. Paixao, *Models and algorithms for single-depot vehicle scheduling*, Transportation Science **35** (2001), 165–180.

[42] F. Friedler, K. Tarjan, Y. W. Huang, and L. T. Fan, *Graph-theoretic approach to process synthesis: polynomial algorithm for maximal structure generation*, Computers & Chemical Engineering **17** (1993), 929–942.

[43] F. Friedler, K. Tarjan, Y.W. Huang, and L.T. Fan, *Combinatorial algorithms for process synthesis*, Computers & Chemical Engineering **16** (1992), S313–S320.

[44] ______, *Graph-theoretic approach to process synthesis: axioms and theorems*, Chemical Engineering Science **47** (1992), 1973–1988.

[45] F. Friedler, J. B. Varga, and L. T. Fan, *Algorithmic approach to the integration of total flowsheet synthesis and waste minimization*, AIChE Symposium Series, Volume on Pollution Prevention via Process and Product Modifications (Eds: M. M. El-Halwagi and D. P. Petrides), 1995, pp. 86–97.

[46] F. Friedler, J. B. Varga, E. Fehér, and L. T. Fan, *Combinatorial accelerated branch-and-bound method for solving the MIP model of process network synthesis*, Nonconvex Optimization and its Applications (Eds: C. A. Floudas and P. M. Pardalos), pp. 609–626, Kluwer Academic Publishers, Norwell, MA, U.S.A., 1996.

[47] F. Friedler, J.B. Varga, and L.T. Fan, *Decision-mapping: a tool for consistent and complete decisions in process synthesis*, Chemical Engineering Science **50** (1995), 1755–1768.

[48] R. Fukasawa, H. Longo, J. Lysgaard, M. P. de Aragão, M. Reis, E. Uchoa, and R. F. Werneck, *Robust branch-and-cut-and-price for the capacitated vehicle routing problem*, Mathematical programming **106** (2006), 491–511.

[49] J. C. García-Ojeda, B. Bertok, F. Friedler, and L. T. Fan, *Building-evacuation-route planning via time-expanded process-network synthesis*, Fire Safety Journal **61** (2013), 338–347.

[50] M. Gendreau, A. Hertz, and G. Laporte, *A tabu search heuristic for the vehicle routing problem*, Management science **40** (1994), 1276–1290.

[51] B. L. Golden, E. A. Wasil, J. P. Kelly, and I. M. Chao, *Metaheuristics in vehicle routing*, Fleet Management and Logistics (T. G. Crainic and G. Laporte, eds.), Kluwer, Boston, 1998, pp. 33–56.

[52] B.L. Golden, S. Raghavan, and E. A. Wasil, *The vehicle routing problem: Latest advances and new challenges*, vol. 43, Springer, 2008.

[53] I.E. Grossman, *Mixed-integer programming approach for the synthesis of integrated process flowsheets*, Computers & Chemical Engineering **9** (1985), 463–482.

[54] I.E. Grossman, J.A. Caballero, and H. Yeomans, *Mathematical programming approaches to the synthesis of chemical process systems*, Korean Journal of Chemical Engineering **16** (1999), 407–426.

[55] ———, *Advances in mathematical programming for the synthesis of process systems*, Latin American Applied Research **30** (2000), 263–284.

[56] I.E. Grossman and Z. Kravanja, *Mixed-integer nonlinear programming techniques for process systems engineering*, Industrial & Engineering Chemistry Research **19** (1995), 189–204.

[57] I.E. Grossman and J. Santibanez, *Applications of mixed-integer linear programming in process synthesis*, Computers & Chemical Engineering **4** (1980), 205–214.

[58] A. Hadjar, O. Marcotte, and F. Soumis, *A branch-and-cut algorithm for the multiple depot vehicle scheduling problem*, Operations Research **54** (2006), 130–149.

[59] A. Haghani and M. Banihashemi, *Heuristic approaches for solving large-scale bus transit vehicle scheduling problem with route time constraints*, Transportation Research Part A: Policy and Practice **36** (2002), 309–333.

[60] K. M. Hangos, J. B. Varga, F. Friedler, and L. T. Fan, *Integrated synthesis of a process and its fault-tolerant control system*, Computers & Chemical Engineering **19** (1995), 465–470.

[61] J. Happel, *Study of kinetic structure using marked atoms*, Catalysis Reviews **6** (1972), 221–260.

[62] ______, *Letter to the editor. Kinetic studies on the mechanism of catalytic ethene oxidation*, Journal of Catalysis **109** (1988), 236–237.

[63] J. Happel and P. H. Sellers, *Analysis of the possible mechanisms for a catalytic reaction system*, **32** (1983), 273–323.

[64] J. Happel, P. H. Sellers, and M. Otarod, *The characterization of complex systems of chemical reactions*, Industrial & Engineering Chemistry Research **29** (1990), 1057–1064.

[65] S. Hassold and A. A. Ceder, *Public transport vehicle scheduling featuring multiple vehicle types*, Transportation Research Part B: Methodological **67** (2014), 129–143.

[66] V. Hatzimanikatis, C. A. Floudas, and J.E. Bailey, *Analysis and design of metabolic reaction networks via mixed-integer linear optimization*, AIChE Journal **42** (1996), 1277–1292.

[67] I. Heckl, R. Adonyi, B. Bertok, F. Friedler, L.T. Fan, and Z. Kovacs, *Formulating and solving vehicle routing problems by the p-graph framework*, VOCAL 2008, December 2008.

[68] J. Horiuti, *Theory of reaction rates based on stoichiometric number concept*, **213** (1973), 5–30.

[69] B. Imreh, J. Fulop, and F. Friedler, *A note on the equivalence of the set covering and process network synthesis problems*, Acta Cybernetica **14** (2000), 497–501.

[70] K. Kalauz, *Algoritmusok kidolgozása logikai, idő és megbízhatósági feltételéket teljesítő folyamathálózatok szintéziséhez*, Ph.D. thesis, Doctoral School of Information Science and Technology , University of Pannonia, 2015.

[71] K. Kalauz, Z. Sule, B. Bertok, F. Friedler, and L. T. Fan, *Extending process-network synthesis algorithms with time bounds for supply network design*, Chemical Engineering Transactions **29** (2012), 259–264.

[72] J. Kallrath, *Mixed integer optimization in the chemical process industry: Experience, potential and future perspectives*, Computers & Chemical Engineering **78** (2000), 809–822.

[73] R.M. Karp, *Reducibility among combinatorial problems*, Complexity of Computer Computations (R.E. Miller and T.W. Thatcher, eds.), Plenum Press, New York, 1972.

[74] R. Karuppiah and I.E. Grossman, *Global optimization for the synthesis of integrated water systems in chemical processes*, Computers & Chemical Engineeringh **30** (2006), 650–673.

[75] J. Klemes, F. Friedler, I. Bulatov, and P. Verbanov, *Sustainability in the process industry, integration and optimization*, McGraw-Hill, New York, 2010.

[76] N. Kliewer, Ba. Amberg, and Bo. Amberg, *Multiple depot vehicle and crew scheduling with time windows for scheduled trips*, Public Transport **3** (2012), 213–244.

[77] N. Kliewer, T. Mellouli, and L. Suhl, *A time-space network based exact optimization model for multi-depot bus scheduling*, European Journal of Operational Research **175** (2006), 1616–1627.

[78] G.R. Kocis and I.E. Grossman, *Global optimization of nonconvex mixed-integer nonlinear programming (minlp) problems in process synthesis*, Industrial & Engineering Chemistry Research **27** (1988), 1407–1421.

[79] Z. Kovacs, F. Friedler, and L.T. Fan, *Recycling in a separation process structure*, AIChE journal **39** (1993), 1087–1089.

[80] Z. Kravanja, *Challenges in sustainable integrated process synthesis and the capabilities of an minlp process synthesizer mipsyn*, Computers & Chemical Engineering **34** (2010), 1831–1848.

[81] G. Laporte, *The vehicle routing problem: An overview of exact and approximate algorithms*, European Journal of Operational Research **59** (1992), 345–358.

[82] ______, *Fifty years of vehicle routing*, Transportation Science **43** (2009), 408–416.

[83] G. Laporte and Y. Nobert, *Exact algorithms for the vehicle routing problem*, Annals of Discrete Mathematics **31** (1987), 147–184.

[84] J. Leboreiro and J. Acevedo, *Processes synthesis and design of distillation sequences using modular simulators: a genetic algorithm framework*, Computers & Chemical Engineering **28** (2004), 1223–1236.

[85] D. Y. Lee, L. T. Fan, S. Park, S. Y. Lee, S. Shafie, B. Bertok, and F. Friedler, *Complementary identification of multiple flux distributions and multiple metabolic pathways*, Metabolic Engineering **7** (2005), 182–200.

[86] J. Lenstra and A. R. Kahn, *Complexity of vehicle routing and scheduling problems*, Networks **11** (1981), 221–227.

[87] B. Lin and D.C. Miller, *Tabu search algorithm for chemical process optimization*, Computers & Chemical Engineeringh **28** (2004), 2287–2306.

[88] Y. C. Lin, L. T. Fan, S. Shafie, B. Bertok, and F. Friedler, *Graph-theoretic approach to the catalytic-pathway identification of methanol decomposition*, Computers & Chemical Engineering **34** (2010), 821–824.

[89] Y. C. Lin, L. T. Fan, S. Shafie, K. L. Hohn, B. Bertok, and F. Friedler, *Catalytic pathways identification for partial oxidation of methanol on copper-zinc catalysts: CH_3OH+ 1/2$O_2 \rightleftharpoons CO_2$ + 2H_2*, Industrial & Engineering Chemistry Research **47** (2008), 2523–2527.

[90] A. Lodi M. Fischetti and P. Toth, *A branch-and-cut algorithm for the multi depot vehicle scheduling problem*, Technical report, 1999.

[91] I. Maros, *Computational techniques of the simplex method*, Kluwer Academic Publishers, Boston, 2003.

[92] M. L. Mavrovouniotis and G. Stephanopoulos, *Synthesis of reaction mechanisms consisting of reversible and irreversible steps. 1. A synthesis approach in the context of simple examples*, Industrial & Engineering Chemistry Research **31** (1992), 1625–1637.

[93] ______, *Synthesis of reaction mechanisms consisting of reversible and irreversible steps. 2. Formalization and analysis of the synthesis algorithm*, Industrial & Engineering Chemistry Research **31** (1992), 1637–1653.

[94] M. Mesquita and J. M. P. Paixao, *Exact algorithms for the multi-depot vehicle scheduling problem based on multicommodity network flow type formulations*, Computer-Aided Transit Scheduling, volume 471 of Lecture Notes in Economics and Mathematical Systems (N. H. Wilson, ed.), Springer, Berlin, 1999, pp. 221–243.

[95] P. C. Milner, *The possible mechanisms of complex reactions involving consecutive steps*, **111** (1964), no. 304, 228–232.

[96] V.M. Nadgir and Y.A. Liu, *Studies in chemical process design and synthesis: Part v: A simple heuristic method for systematic synthesis of initial sequences for multicomponent separations*, AIChE Journal **29** (1983), 926–934.

[97] A. Nagy, *Globális optimalizálási algoritmusok pns feladatok megoldására*, Ph.D. thesis, Doctoral School of Information Science and Technology , Veszprém University, 2004.

[98] A. B. Nagy, R. Adonyi, L. Halasz, F. Friedler, and L. T. Fan, *Integrated synthesis of process and heat exchanger networks: algorithmic approach*, Applied Thermal Engineering **21** (2001), 1407–1427.

[99] N. Nishida, G. Stephanopoulos, and A.W. Westerberg, *A review of process synthesis*, AIChE Journal **27** (1981), 321–351.

[100] I. H. Osman, *Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem*, Annals of operations research **41** (1993), 421–451.

[101] J. M. P. Paixao and I. Branco, *A quasi-assignment algorithm for bus scheduling*, Networks **17** (1987), 249–269.

[102] J. A. Papin, N. D. Price, S. J. Wiback, D. A. Fell, and B. O. Palsson, *Metabolic pathways in the post-genome era*, Trends in biochemical sciences **5** (2003), 250–258.

[103] J. A. Papin, J. Stelling, N. D. Price, S. Klamt, S. Schuster, and B. O. Palsson, *Comparison of network-based pathway analysis methods*, Trends in biotechnology **22** (2004), 400–405.

[104] Jason A Papin, Nathan D Price, Jeremy S Edwards, and B. O. Palsson, *The genome-scale metabolic extreme pathway structure in haemophilus influenzae shows significant network redundancy*, Journal of theoretical biology **215** (2002), 67–82.

[105] V. Pillac, M. Gendreau, C. Guéret, and A.L. Medaglia, *A review of dynamic vehicle routing problems*, European Journal of Operational Research **225** (2013).

[106] D. Pisinger and S. Ropke, *A general heuristic for vehicle routing problems*, Computers & operations research **34** (2007), 2403–2435.

[107] N. D. Price, J. A. Papin, and B. O. Palsson, *Determination of redundancy and systems properties of the metabolic network of helicobacter pylori using genome-scale extreme pathway analysis*, Genome research **12** (2002), 760–769.

[108] C. Prins, *A simple and effective evolutionary algorithm for the vehicle routing problem*, Computers & Operations Research **31** (2004), 1985–2002.

[109] R. Raman and I.E. Grossman, *Relation between milp modelling and logical inference for chemical process synthesis*, Computers & Chemical Engineering **15** (1991), 73–84.

[110] C. Ribeiro and F. Soumis, *A column generation approach to the multiple-depot vehicle scheduling problem*, Operations Research **42** (1994), 41–52.

[111] C. H. Schilling, D. Letscher, and B. O. Palsson, *Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective*, Journal of theoretical biology **203** (2000), 229–248.

[112] C. H. Schilling and B. O. Palsson, *The underlying pathway structure of biochemical reaction networks*, Proceedings of the National Academy of Sciences **95** (1998), 4193–4198.

[113] ______, *Assessment of the metabolic capabilities of haemophilus influenzae rd through a genome-scale pathway analysis*, Journal of theoretical biology **203** (2000), 249–283.

[114] S. Schuster, T. Dandekar, and D. A. Fell, *Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering*, Trends in biotechnology **17** (1999), 53–60.

[115] P. H. Sellers, *An introduction to a mathematical theory of chemical reaction networks I*, Archive for Rational Mechanics and Analysis **44** (1971/72), 23–40.

[116] ______, *An introduction to a mathematical theory of chemical reaction networks II*, Archive for Rational Mechanics and Analysis **44** (1971/72), 376–386.

[117] ———, *Combinatorial classification of chemical mechanisms*, SIAM Journal on Applied Mathematics **44** (1984), 784–792.

[118] H. Seo, D. Y. Lee, S. Park, L. T. Fan, S. Shafie, B. Bertok, and F. Friedler, *Graph-theoretical identification of pathways for biochemical reactions*, Biotechnology letters **23** (2001), 1551–1557.

[119] A.M. Shoaib, S.M. Aly, M.E. Awad, D.C.Y. Foo, and M.M. El-Halwagi, *A hierarchical approach for the synthesis of batch water network*, Computers & Chemical Engineering **32** (2008), 530–539.

[120] J. J. Siirola, *Industrial application of chemical process synthesis*, Process synthesis (J. L. Anderson, ed.), Advances in chemical engineering, vol. 23, Academic Press, New York, 1996, pp. 63–170.

[121] J. Smidla and I. Heckl, *S-graph based scheduling method for parallel architecture*, Chemical Engineering Transactions **21** (2010), 937–942.

[122] I. Szalkai, *A new general algorithmic method in reaction syntheses using linear algebra*, Journal of Mathematical Chemistry **28** (2000), 1–34.

[123] ———, *Reakció mechanizmusok algoritmikus és matematikai vizsgálata*, Ph.D. thesis, Doctoral School of Information Science and Technology , University of Pannonia, 2014.

[124] É. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J.Y. Potvin, *A tabu search heuristic for the vehicle routing problem with soft time windows*, Transportation science **31** (1997), 170–186.

[125] P. Toth and D. Vigo (Eds.), *The vehicle routing problem*, SIAM Monographs on Discrete Mathematics and Applications, Society for Industrial & Applied Mathematics, Philadelphia, 2001.

[126] M. Urselmann, S.Barkmann, S. Sand, and S.Engell, *A memetic algorithm for global optimization in chemical process synthesis problems*, IEEE Transactions on Evolutionary Computation **15** (2011), 659–683.

[127] R. A. van Santen and M. Neurock, *Chemisorption theory*, Handbook of Catalysis (H. Knoezinger, P.J. Jacobs, and G. Ertl, eds.), Wiley-VCH, Weinheim, Germany, 1997, pp. 942–958.

[128] ______, *Theory of surface reactivity*, Handbook of Catalysis (H. Knoezinger, P.J. Jacobs, and G. Ertl, eds.), Wiley-VCH, Weinheim, Germany, 1997, pp. 991–1005.

[129] L. Vance, H. Cabezas, I. Heckl, B. Bertok, and F. Friedler, *Synthesis of sustainable energy supply chain by the p-graph framework*, Industrial & Engineering Chemistry Research **52** (2012), 266–274.

[130] Sz. Varró-Gyapay, *Trajectory set approximation for optimization and verification of it systems*, Ph.D. thesis, Budapest University of Technology and Economics Department of Measurement and Information Systems, 2014.

[131] K. Wang, Y.Qian, and P. Yao, *Synthesis and optimization of heat integrated distillation systems using an improved genetic algorithm*, Computers & Chemical Engineering **23** (1998), 125–136.

[132] A.W. Westerberg, *Review of process synthesis*, Computer applications to chemical engineering (R. G. Squires and G. V. Reklaitis, eds.), ACS Symposium Series, vol. 124, American Chemical Society, San Diego C.A., 1980, pp. 53–87.

[133] S. J. Wiback and B. O. Palsson, *Extreme pathway analysis of human red blood cell metabolism*, Biophysical Journal **83** (2002), 808–818.

[134] B. Yu, Z.Z. Yang, and B. Yao, *An improved ant colony optimization for vehicle routing problem*, European journal of operational research **196** (2009), 171–176.