PhD Dissertation

Optimization of energy and utility supply systems

DOI:10.18136/PE.2023.833

András Éles

Supervisor:
István Heckl, PhD

University of Pannonia
Faculty of Information Technology
Department of Computer Science and
Systems Technology

Doctoral School of Information Science and Technology

Veszprém
2022

Optimization of energy and utility supply systems

Az értekezés doktori (PhD) fokozat elnyerése érdekében készült a Pannon Egyetem Informatikai Tudományok Doktori Iskolája keretében

informatikai tudományok tudományágban

Írta: Éles András

Témavezető: Dr. Heckl István

Elfogadásra javaslom (igen / nem)

. . . .
(témavezető)

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

Veszprém, ...

. . . .
(a Szigorlati Bizottság elnöke)

Az értekezést bírálóként elfogadásra javaslom:

Bíráló neve: . . . . igen / nem

. . . .
(bíráló)

Bíráló neve: . . . . igen / nem

. . . .
(bíráló)

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

Veszprém, ...

. . . .
(a Bíráló Bizottság elnöke)

A doktori (PhD) oklevél minősítése . . . .

Veszprém, ...

. . . .
(az EDHT elnöke)

Contents

Acknowledgements

I would like to express my gratitude towards everyone who provided support to me, to those who helped a lot for me personally and in making my PhD dissertation possible.

I specially thank my supervisor Dr. István Heckl for his guidance throughout the years, for providing me the opportunities of research topics, ideas and advice, and for keeping an eye on my workflow. I also thank Prof. Heriberto Cabezas for his advice and providing case studies. In the last twelve years, I had the luck to meet and work with many colleagues and fellow students, from whom I received much help and learned a lot. I specially thank Máté Hegyháti for introducing me to scientific projects and in particular the area of optimization and mathematical programming.

I also thank my girlfriend Anna for encouraging me and helping to keep my duties in mind. I also thank my family in Debrecen, by whom my IT career was made possible in the first place.

Abstract

At our present time, the production of many products and the provision of many services are preceded by a thorough planning. This is due to the large number of possibilities regarding the selection of raw materials and technologies, distribution of resources, and exact decisions for executing tasks. The most common goal is profit maximization, however, several other aspects such as environmental impact and sustainability are gaining increasing attention. Energy supply in particular is a field with many problems where these aspects become relevant. Renewable energy sources may have limited availability and spatial distribution, requiring the formulation of complex optimization problems to obtain their full potential. Transportation itself can also be a critical component of a supply chain, as it can significantly contribute to costs and environmental impacts.

In my thesis, optimization models and solutions for multiple optimization problems targeting energy and general utility providing systems are presented. The P-Graph framework, which is a general purpose modeling and optimization tool for Process Network Synthesis (PNS) problems, was applied to an energy supply problem of a single processing plant. This problem focuses on the decision of whether investment into biogas utilization and solar panels is economical in the long term, and determines which types of the locally available biomass are the most advantageous for this purpose. The approach involves the multi-period modeling scheme and an operating unit model with flexible inputs. In the second part, a general modeling technique is presented, which makes complex operations with flexible inputs possible to be managed with the tools of the P-Graph framework. This technique allows arbitrary linear constraints on the inputs of an operation. The flexible input scheme is also demonstrated on a case study involving the energy supply of a rural area by locally produced biomass. Afterwards, a Mixed-Integer Linear Programming (MILP) model and an algorithmic framework are presented, which are capable of solving the mobile workforce management problem with a slot-based modeling technique. The model takes a wide range of circumstances into account, including packing and unpacking times, time windows, resource utilization and task relations.

Tartalmi kivonat

Manapság sok termék és szolgáltatás előállítását előzi meg alapos tervezés. Ennek oka, hogy sok lehetőség van a nyersanyagok, technológiák megválasztására, az erőforrások elosztására, és a tevékenységek pontos végrehajtására. A leggyakoribb cél a profit maximalizálása, azonban több egyéb szempont kap egyre nagyobb figyelmet, mint például a környezeti hatások vagy a fenntarthatóság. Az energiaellátás konkrétan egy olyan problémákat felvető terület, ahol ezek a szempontok meghatározók. A megújuló energiaforrások elérhetősége korlátozott és földrajzilag elszórt lehet, így a bennük rejlő potenciál kiaknázása összetett optimalizálási feladatok formális megfogalmazását teszi szükségessé. Maga a szállítás is kritikus eleme lehet egy ellátási láncnak, mivel lényegesen hozzájárulhat a költségekhez és a környezeti hatásokhoz.

Dolgozatomban optimalizálási modelleket és megoldásokat mutatok be, amelyek általános energia- és egyéb szolgáltató rendszereket céloznak meg. A P-Gráf keretrendszerrel, amely folyamathálózat-szintézis (PNS) feladatok megoldására egy általános célú modellező és optimalizáló eszköz, egy gyár energiaellátására vonatkozó feladatot oldottam meg. Ez a feladat arra fókuszál, hogy biogáz- vagy napenergia alapú energiatermelésbe megéri-e befektetni hosszabb távon, és meghatározza, hogy a biomassza mely típusai a legkedvezőbbek erre. A megközelítés többperiódusú modellezési módszert és rugalmas bemenetű műveleti egység modellt használ. A második részben egy általános modellezési technikát ismertetek, amely lehetővé teszi összetett, rugalmas bemenetű műveletek kezelését a P-Gráf módszertan eszközeivel. Ez a technika tetszőleges lineáris korlátozást támogat a művelet bemeneteire. A rugalmas bemenetű modellezési módszert egy esettanulmány is demonstrálja, amelyben egy kisebb vidéki terület energiaellátásának a helyben termelt biomassza felhasználásával való megoldása a cél. Ezt követően egy Kevert Egészes Lineáris Programozási (MILP) modellt és keretalgoritmust mutatok be, amik képesek egy mobil munkaerő menedzsment feladat megoldására számos körülmény, mint a pakolási idők, időablakok, erőforrások kezelése és taszkok közötti relációk figyelembe vételével.

Resumen

En la actualidad, la producción de muchos productos y la prestación de muchos servicios están precedidos por una minuciosa planificación. Esto se debe a la gran cantidad de posibilidades en cuanto a la selección de materias primas y tecnologías, la distribución de recursos y las decisiones exactas para ejecutar las varias tareas. El objetivo más común es la maximización de las ganancias, sin embargo, varios otros aspectos, como el impacto ambiental y la sostenibilidad, están ganando cada vez más atención. El suministro de energía en particular es un campo con muchos problemas donde estos aspectos toman gran relevancia. Las fuentes de energía renovable pueden tener disponibilidad y distribución espacial limitadas, lo que requiere la formulación de problemas complejos de optimización para obtener todo su potencial. El transporte en sí mismo también puede ser un componente crítico de una cadena de suministro, ya que puede contribuir significativamente a los costos y los impactos ambientales.

En mi tesis, se presentan modelos de optimización y soluciones para múltiples problemas de optimización dirigidos a sistemas de suministro de energía y de servicios generales. La estructura teórica y "software" de P-Graph es una herramienta de modelado y optimización de propósito general para los problemas de Síntesis de Redes de Procesos (Process Network Synthesis, PNS). Esta se aplicó a un problema de suministro de energía de una sola planta de procesamiento. Este problema se centra en la decisión de si la inversión en la utilización de biogás y paneles solares es económica a largo plazo, y determina qué tipos de biomasa disponible localmente son los más ventajosa para este propósito. El enfoque implica el esquema de modelado multiperíodo y un modelo de unidad operativa con entradas flexibles. En la segunda parte, se presenta una técnica general de modelado, que permite gestionar operaciones complejas con entradas flexibles con las herramientas de P-Graph. Esta técnica permite restricciones lineales arbitrarias en las entradas de una operación. El esquema de entrada flexible también se demuestra en el estudio de un caso que involucra el suministro de energía de una zona rural utilizando biomasa producida localmente. Posteriormente, se presenta un modelo de programación lineal entera mixta (Mixed Integer Linear Programming, MILP) y un método algorítmico, que son capaces de resolver el problema de gestión de la fuerza laboral móvil con una técnica de modelado basada en ranuras. El modelo tiene en cuenta una amplia gama de circunstancias, incluidos los tiempos de embalaje y desembalaje, ventanas de tiempo, utilización de recursos y relaciones de tareas.

List of abbreviations

ABB	Accelerated Branch and Bound
ACO	Ant Colony Optimization
CHP	Combined Heat and Power
CMN	Carbon Management Network
EUR	Euro (currency)
EV	Electric Vehicle
GA	Genetic Algorithm
HUF	Hungarian Forint (currency)
LHS	left-hand side
LNS	Large Neighborhood Search
MILP	Mixed-Integer Linear Programming
MSG	Maximal Structure Generation
NPP	Net Primary Production
PNS	Process Network Synthesis
PSO	Particle Swarm Optimization
RHS	right-hand side
SA	Simulated Annealing
SNS	Separation Network Synthesis
SSG	Solution Structure Generation
TCPNS	Time-Constrained Process Network Synthesis
TS	Tabu Search
TSP	Travelling Salesman Problem
VRP	Vehicle Routing Problem

Chapter 1

Introduction

1.1 Motivation

Optimization in general means looking for a solution of some given problem that best fits our needs. This goal is characterized by the freedom of choice we have, the criteria determining which solutions are acceptable, and the objective telling how good a particular solution is. What makes optimization an important field of mathematics and information technology is the complexity of these problems arising from practice, including real-world instances. Organizing the supply chain of a resource, provision of a service, decision on production in a single facility, down to managing the transportation of personnel and resources, and calculating exact timing of activities are just a few examples where optimization can be crucial. The number of system components, their interactions, viable choices on possible resources, used technologies and procedures, and exact decisions from design to operation may have a significant impact on feasibility, profitability, sustainability and other properties of the whole system. The traditional method for supporting decision making, which is relying on the intuition of experts of the given area can be insufficient to tackle problems with this complexity. This establishes the need of computational methods that systematically take into account a wide range of possible decisions, preferably all that can be advantageous.

Naturally, the economical benefits had been in focus of decision making. For this reason, profit or throughput maximization, cost minimization are common objectives in real-world optimization problems. However, other aspects like long-term sustainability, environmental and social impacts are gaining increasing attention. These aspects orient optimization problems in the form of available opportunities to choose from, restrictions on the search space, or alteration of the objective, making problem formulations more complex.

Sustainability governs efforts to find practically possible ways of providing conditions on Earth suitable for human life in the long term. This is quite a challenge due to several reasons. The human population is large and increasing [1], which causes a steady increase in energy footprints [2], while the consumption of resources is also showing an increasing trend. There was a fourfold increase in private consumption expenditures from 1960 to 2000 [3]. The human population is using a large portion, approximately 38% of the world terrestrial Net Primary Production (NPP), leaving a smaller amount to the planetary ecosystem [4], and energy production is still heavily relying on fossil fuels [5]. The supply chain of a product or service may use a large amount of energy, due to manufacturing, transportation, and management of waste. Therefore, one possibly effective way of decreasing the human footprint is the optimization of the parts of these already existing processes, considering the propagation of technologies and methods which are more sustainable, while economically still viable. Developing effective tools to support decision making, on a level capable of managing the complexity of such conflicting goals is at the core of my interest.

1.2 Optimization methods

In the recent years I met several instances of optimization problems that were motivated by a real-world application and/or had a theoretical importance. It is apparent that there are some well-established and general solution techniques. Generality in this sense means that the same technique can be adapted to a large number of otherwise unrelated case studies. On the other hand, the extension of an existing framework or the combination of different tools might also yield the best result. Algorithms may exist that both guarantee fast performance and globally optimal result, but only for special problem classes which are often inadequate to model real-life problem instances.

Heuristic approaches are based on some good rule of thumb for finding solutions of a problem. Optimality of the solution is not guaranteed in this case. The effectiveness of a heuristic method is determined by how well the rule captures the essence of finding better solutions. There are well-known general metaheuristics, for example Genetic Algorithms (GA), Simulated Annealing (SA), Tabu Search (TS), Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). Note that many other individual algorithms can be regarded as heuristic as well.

Mathematical programming techniques are another kind of approach widely used. In a mathematical programming model, the freedom of decisions is expressed as model variables. The restrictions determining acceptable solutions are formulated as model constraints, which are usually equations and inequalities involving the variables and parameters, while the objective of optimization is a function of these variables and parameters. A great advantage of mathematical programming is that attention can be focused on formulating the model itself, but the solution procedure can be performed by some external software called solver, depending on problem class. More general problem classes are applicable on a larger set of problems, usually in exchange for a larger computational complexity. A good compromise in between is the class of Mixed-Integer Linear Programming (MILP) models, which is widely used in optimization problems. In MILP models, variables may either have integer or real values, and constraints and the objective are all linear in terms of the variables. Globally optimal solutions can be theoretically guaranteed for MILP models, but in practice, computational complexity prevents us from achieving it for larger problem instances.

The P-Graph framework is another effective solution method I particularly relied on during my work. This is a technique capable of optimization at various fields, published in 1992 by Friedler et al. [6]. The framework consists of the mathematical model of P-Graphs, algorithms and software tools to represent, formulate, and solve Process Network Synthesis (PNS) problems. A P-Graph defines a process as a directed bipartite graph, where the two partitions of nodes represent states and operations, and arcs represent material flow. The aim of a PNS problem is to determine a process network if its possible components and interactions are given. Therefore, PNS covers the core decisions in optimization problems which can be modeled by process networks. Accompanying the PNS problem with data like flow rates, costs and capacities, and utilizing solution algorithms like the Accelerated Branch and Bound (ABB) algorithm [7] make the framework a complete optimization tool and alternative to other approaches.

The P-Graph Studio [8] is a demonstration software, which is freely available and can be used to formulate PNS problems graphically and calculate solutions for them with the ABB or apply other algorithms. This software was used to obtain the results in this work for case studies where the P-Graph framework was used, and notably the graphical representations of P-Graphs as well.

1.3 Overview of this document

In my PhD Dissertation, applications and extensions of the P-Graph framework, MILP model-based and algorithmic methods are presented on multiple case studies targeting energy supply, optimal transportation and workforce management. The literature review of the applied tools, solution methods, similar and related case studies is presented in Chapter 2. The main parts of my scientific contribution are shown as three Theses, each detailed in the respective chapters.

In my first Thesis (Chapter 3), the optimization of the energy supply of a manufacturing plant is presented. The P-Graph framework was used to formulate the optimization model. This work is motivated by the opportunity to invest into renewable energy sources to satisfy the energy requirements of the plant. Heating and electricity demands were estimated based on historical data. Different types of biomass produced locally in the region were included in the model as possible energy sources, from which biogas can be produced and used in a biogas furnace or a Combined Heat and Power (CHP) plant. Solar panels were also considered, as well as the business as usual method of purchasing natural gas from the provider or electricity from the grid. The final version of the model utilizes the multi-period modeling scheme for energy demands, which is a more precise representation than the ordinary single period scheme. Additionally, the pelletizer and biogas plant were modeled as flexible operating units, a method which was generalized afterwards.

In my second Thesis (Chapter 4), a general modeling method is described to implement operations with flexible inputs in the P-Graph framework. The notion of flexible inputs refer to scenarios where the ratios of input materials of an operation are not fixed but variable. The presented method does not only support the flexibility of inputs, but arbitrary linear constraints can be imposed on input amounts. The novelty of this approach is that instead of manual workarounds, the existing implementations for P-Graph framework, including the ABB algorithm can be used to obtain this model, using operating units with fixed input ratios only. The energy supply problem from Chapter 3 is revisited for the illustration of the approach. The advantage of using flexible input models, and the demonstration of the introduced P-Graph modeling technique are presented in another case study involving the energy supply of a small rural region.

In my third Thesis (Chapter 5), a general MILP model-based solution approach is presented for a mobile workforce management problem. This is a complex optimization problem which involves an optimal assignment of a given set of spatially distributed tasks among a given set of executing teams, with precise ordering and timing. A slot-based MILP model is proposed, which takes a previously unmatched variety of circumstances into account, including packing and unpacking times of mobile teams, relative and absolute time windows of execution, consumable and tool resources, and many kinds of pair-wise relationships between tasks. For larger scale problems, an algorithmic framework is developed which is based on the MILP model and provides heuristic solutions in acceptable computational time. The effectiveness of both the standalone MILP model and the algorithmic framework were tested on many different problem instances, from different aspects.

Chapter 2

Literature Review

This chapter is structured as follows. In Section 2.1, the P-Graph framework is presented in detail with the most important algorithms, extensions and applications, as P-Graphs were a key tool for my results. In Section 2.2, a review of the literature of mobile workforce management, related problems and approaches is presented.

2.1 The P-Graph framework

2.1.1 Fundamentals

The Process Graph, or P-Graph is a mathematical representation of process networks. It was first published in 1992 by Friedler et al. [6]. The advantage of a P-Graph over other representations like flow sheets is that a process network can be unambiguously described.

A P-Graph is a directed bipartite graph defined by a pair of node sets, (M, O), and arcs in between these. The two sets have the following meaning.

- M is the set of material or M-type nodes. A material node can naturally represent actual materials, but can have other more abstract meanings like energy, other physical quantities, virtual resources, different states of a given entity, and even the fulfillment of logical conditions.

- O is the set of operating unit or O-type nodes. An operating unit can represent any transformation between the quantities represented by the material nodes, most notably production, conversion or transfer between different states of the same or different resources.

The arcs of the P-Graphs represent the direction of material flow in the process network (see Figure 1).

- An arc from an M-type node to an O-type node indicates that the operating unit has the resource as its input. Therefore, the operation requires the resource to be available for consumption.

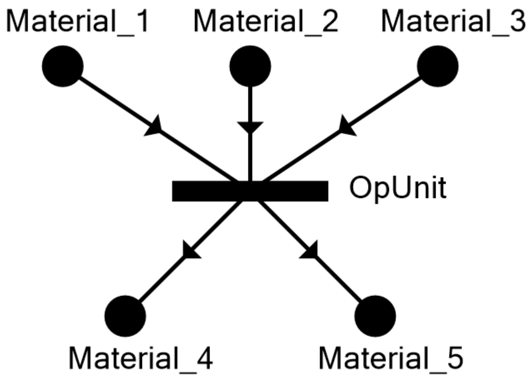


Figure 1: Operating unit node called *OpUnit*, which consumes three input materials, numbered 1, 2 and 3, and produces two output materials, numbered 4 and 5.

- An arc from an O-type node to an M-type node indicates that the operating unit has the resource as its output. Therefore, performing the operation produces the resource, making it available for further consumption.

We also say that an M-type node has inputs and outputs: these are the O-type nodes from which there is an arc to the M-type node, or to which there is an arc from the M-type node, respectively.

To specify the boundaries of a process network, M-type nodes are further partitioned into three types.

- Raw material nodes are supposed to be available in the process network from an external source, these are not produced. The set of raw materials is typically denoted by R.
- Product nodes represent the final products to be created in the process. These must be produced by operating units of the process network. The set of products is typically denoted by P. Note that product nodes may represent arbitrary goals and results other than actual physical products, like energy, logical conditions or byproducts.
- The rest of material nodes are intermediate material nodes. These materials are produced, and are required for the production of other intermediate and product materials. The set of intermediates can be denoted by I.

A Process Network Synthesis (PNS) problem is characterized by a (P, R, O) triplet, where P is the set of products, R is the set of raw materials and O is the set of available operating units. The goal of PNS is to identify solution structures, as explained later. With the material nodes defined by P, R, and the inputs and outputs of all operating units, a P-Graph representation is obtained.

The conventional graphical representation of a P-Graph, representing a PNS problem, can be seen in Figure 2. A huge advantage of the P-Graph framework is that modeling process networks can be technically more straightforward than using other approaches, like mathematical programming methods.

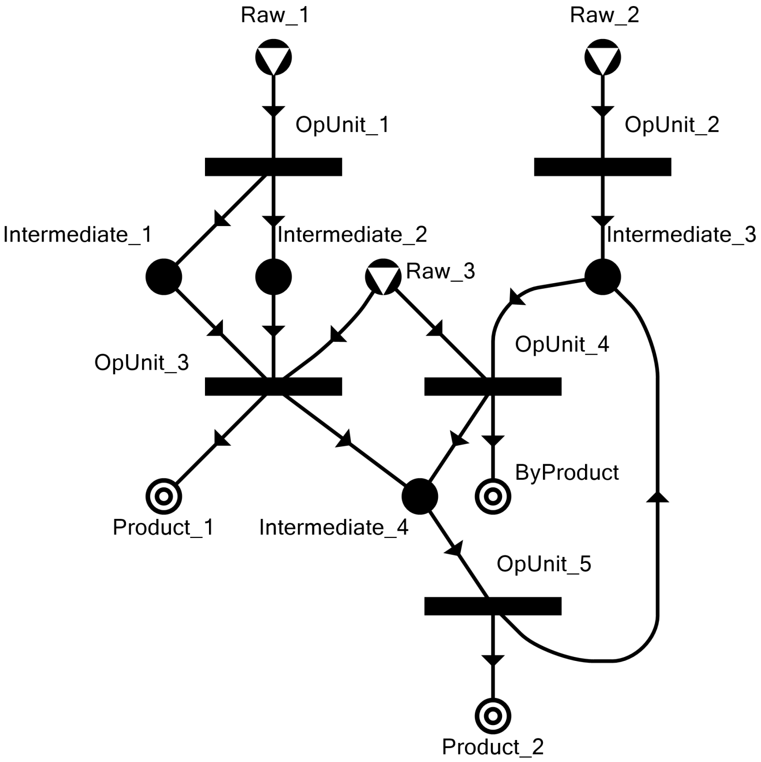


Figure 2: Example P-Graph with different material node types.

2.1.2 Basic algorithms

The solution structure of a PNS problem is a subgraph of its underlying P-Graph which represents a feasible implementation of the process network. A selection of operating units to be involved in the process is made, and all of their corresponding input and output materials are included. A solution structure is itself a P-Graph. The necessary and sufficient conditions for a P-Graph to be a solution structure of a PNS problem can be expressed in the following five axioms [6].

S1. All final products P are included.

S2. An included material node m has no inputs if and only if $m \in R$.

S3. All included operating unit nodes o are defined in the PNS problem, which means $o \in O$.

S4. All included operating unit nodes o must have a directed path within the structure ending in some $m \in P$.

S5. All included material node m is an input or the output of some included operating unit o.

The Maximal Structure Generation (MSG) algorithm is an important step in finding solution structures for a PNS problem [9]. The algorithm excludes parts of the P-Graph which are unusable to obtain any solution structures. It can be proven that the resulting maximal structure contains all solution structures as a subgraphs. Note that the maximal structure is itself a solution structure.

The Solution Structure Generation (SSG) algorithm uses the concept of decision mapping to systematically generate all solution structures of a PNS problem [10]. The number of solution structures can be exponential in the number of nodes of the original P-Graph. For this reason, if the number of solution structures of a P-Graph is large, then the running time of the SSG algorithm is high as well. On the other hand, as solution structures capture the basic logical rules for a feasible process network, the number of solution structures are still significantly smaller than the total number of cases considered if the generation was done with some naive, brute-force method instead.

The process network usually has accompanying numerical data of the materials and operating units involved. Some of the most important for material nodes are costs or revenues, minimum and maximum flows. For operating unit nodes, there can be flow rates for each input and output material, minimum and maximum capacities, and fixed and proportional investment and operating costs. An example P-Graph with flow rates is depicted in Figure 3.

The resulting solution structures from the SSG algorithm can be individually further investigated whether they correspond to actually feasible solutions for the modeled real-world problem. This results in all structurally feasible solutions to be taken into account. Note that being a solution structure is a necessary but not sufficient condition of an actually feasible solution, because it is possible that, for example, constraints like minimum production amounts, maximum operation capacities and raw material availability cannot be simultaneously satisfied.

The real-world optimization problem a PNS problem represents is further characterized by flow rates, capacities, costs and other data. This results in an optimization model based on PNS. A possible procedure for solving such an optimization model is the application of the Accelerated Branch and Bound (ABB) algorithm, supported by the P-Graph Studio software. This algorithm, after simplifying the process network with the MSG, takes into account all possible solutions structures, although does not enumerate all of them as SSG does. Instead, a Branch and Bound framework is used to find the solution structure with the optimal solution. The objective can be cost minimization or profit maximization, depending on context.

Another advantage of the P-Graph framework, particularly ABB in the current implementation, is that the optimization procedure is directly capable of reporting the N best (near-optimal) solution structures for some integer N, instead of just the single optimal solution [11]. The same can be tedious to achieve with mathematical programming tools. Reports suggest that P-Graphs can be useful for teaching

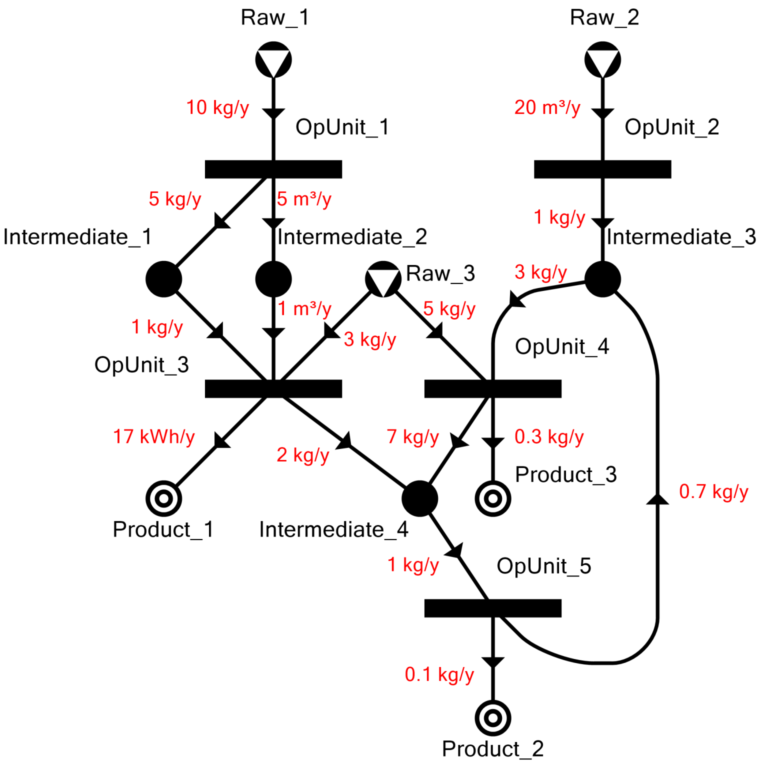


Figure 3: P-Graph with flow rates, the structure is identical to the one in Figure 2.

purposes as well in the topic of system modeling and optimization [12], possibly in complement with other mainstream techniques like mathematical programming [13].

2.1.3 Extensions

The P-Graph Studio demonstration software [8] is available and still being developed, as new features are included and fine-tuning of the algorithms is taking place to improve performance [14]. A review was made by Klemes and Varbanov about the variety of problems that can be modeled and optimized with the P-Graph framework [15], particularly mentioning possible extensions and future development. Note that, the theoretical and software background had also been improved throughout the years with several extensions.

The Time-Constrained PNS (TCPNS) problem formulation makes timing constraints possible for flows [16]. This effectively incorporates scheduling decisions in the model. This makes it possible to address situations when a fixed set of tasks is given and their order must be found [17]. Note that similar decisions had already been modeled in certain circumstances, for example vehicle routing where deliveries are fixed in time [18], before TCPNS was introduced. These methods demonstrate that operating units in the P-Graph framework do not necessarily represent production steps, but possibly other abstract relationships like precedence.

Flexible inputs are an issue which arises when an operating unit should accept a variable ratio of their input materials. The original implementations of a PNS problem and solving algorithms only support constant value as flow rate for each arc, which result in operating units with a fixed ratio of all of their input materials with respective to each other. Recent results have shown that in this class of problems, if the underlying MILP model is extended with additional constraints, PNS is capable of modeling operating units with flexible inputs and constraints on these inputs [19]. Note that the extended model is not solvable directly with the original method, the ABB algorithm. In Chapter 4, a method is presented to achieve the same result but staying within the frames of the original framework. This scheme of flexible inputs and constraints based on these inputs has a huge potential in applications, for example, it was already used in input-output modeling [20].

Multi-period modeling is another extension of the P-Graph framework. Basically, for capacities and materials flows, only the total amounts are considered. A P-Graph therefore captures structural feasibility and material balance, in a single point of time or a given time horizon. This is sufficient for a working model as long as the distribution of total flows among the total capacities represented this way are not relevant. This is the case when practically unlimited storage is available for the supplies or the demands, or production can be evenly distributed over time. In some cases, however, either the supplies are available or the demands are required unevenly throughout a time horizon. For this purpose, the time horizon can be separated into multiple different time periods with their own rates for supplies and/or demands. This can be done within the P-Graph framework [21]. If there are further constraints for production, for example minimal required flow for equipment units, their modeling becomes more complicated. The multi-period modeling scheme can be extended to such scenarios [22]. Although multi-period modeling is a logical extension of the framework, which means it can be implemented with existing techniques, there is software support for it in the P-Graph Studio software [23].

2.1.4 Applications

Although P-Graphs were originally developed to model chemical processes, it is a general modeling tool with a vast number of applications including PNS, reliability engineering and systems analysis. Available tools and applications were described in detail in a recent book [24]. Another review of applications was done by Lam [25], which presents a collection of case studies involving single plant management, whole supply chains, and transportation. Another review focuses on applications where sustainability was a key motivation, involving processing facility and supply chain optimization [26]. The review by Klemes and Varbanov mentions extensions and emphasizes future applications [15].

The methodology in general is an alternative to the implementation of MILP models, or other solution methods. For example, pinch analysis [27] is a widely used tool for system design. One possible application of pinch analysis is energy sector planning with carbon emission constraints [28]. It was demonstrated that this problem can be formulated and solved with the P-Graph framework [29].

There are also applications where the P-Graph framework is used in combination with other techniques. An example for this is when P-Graphs and conventional mathematical programming tools were used simultaneously to obtain a decomposition of the main problem [30]. Several types of biomass were considered in connection with the palm oil industry, but the authors remark that their results should be regularly revised to reflect rapidly changing circumstances. Another set of biomass sources was inspected in a problem involving wood process residues [31]. A phenomenon could be observed in this case study: minor changes in the problem data can result in significant changes in the order of optimality among the solution structures. This makes the ability of the P-Graph framework to report multiple near-optimal solutions a valuable feature. A possible solution strategy is identifying a list near-optimal solutions as a first level of problem decomposition and using it as a basis for further analysis and optimization [32].

Spatial distribution of biomass availability and demand points for them can be simultaneously taken into account, resulting in the optimization of a whole renewable energy supply chain [33]. The authors in this case study also utilized mathematical programming and the P-Graph framework together. The mathematical programming tools were used to make clusters of the spatial points, then a PNS problem was formulated to optimize material flows in between these clusters.

When energy requirements are considered in plant design, heat and electricity are commonly addressed in a single model. The available biomass sources can be considered in parallel to purchasing natural gas or electricity from the grid directly, or maybe as a complete replacement [34]. The case study reported a possible 17% decrease in total operating costs if biomass sources are integrated into the supply. Other objectives like ecological footprint can also be taken into account [35]. The P-Graph framework was also used to find bottlenecks in the supply chain of biomass source utilization [36]. The approach relied on the near-optimal solution structures which could be obtained with the appropriate tools. The method was used for the improvement of sustainability indices in three different scenarios [37]. Case studies may consider special biomass types, for example intercrops [38].

Separation Network Synthesis (SNS) arises in a production environment when chemicals must be separated into their pure components, which can be done in multiple possible paths. The number of possible resulting materials can be practically infinite, however, special types of SNS problems can be formulated as a PNS problem and afterwards solved by the ABB algorithm [39].

Production line balancing is the problem of assigning productions steps to executing staff or machines at an assembly line, in order to minimize bottlenecks and maximize throughput of the line. The case when task order is fixed was presented to be solved by either the P-Graph framework and ABB algorithm alone or a dedicated MILP model [40]. This application is rather special because only assignment and ordering decisions are mapped to PNS instead of flow of actual materials. Similar cases where assignment of resources is the key decision can be addressed, for example workforce management [41].

Polygeneration plant modeling and optimization can also be achieved with P-Graphs [42]. This particular case study involved multiple uncertain demands like heating, cooling, water and electricity. Risks and possible reactions in case of an equipment failure event is also a critical issue. This was also handled within the same framework, as an alternative to MILP models [43]. In general, uncertainty in supplies and demands, and risks of inoperability have a significant effect on supply chains. The P-Graph framework can be used for considering reliability constraints [44], or as a tool for risk analysis [45]. For this reason, it was also used to investigate the effect of inoperability in the supply networks of bioenergy parks [46], and to find optimal adjustments [47]. In the modeling point of view, a rather simple case is when risks can be represented as parameters of the available options. This is the case when the probability of losses can be estimated for each activity and penalty terms can be assigned in the objective accordingly [48]. Authors aimed at minimizing fatalities throughout the bioenergy supply chain in this example. It is also possible to use P-Graph as a backbone of a fuzzy model instead of a deterministic one [49].

Synthesis of Carbon Management Networks (CMN) is a class of usually complex problems important from the viewpoint of sustainability. The P-Graph framework was utilized for the synthesis of a biochar-based CMN [50]. The proposed model uses a set of sources of biochar and a set of sinks that can contain it. Operating units represent transportation between any pairs of the possible sources and the sinks. Another example for CMN addressed with the P-Graph framework used the Monte Carlo simulation to test near-optimal solutions reported by the PNS solver and test their robustness [51].

The multi-period modeling scheme in the P-Graph framework first appeared in a case study where corn production was investigated [52]. Both the supply side and the demand side were specific for each period. In another model formulation, annual electricity production for varying sources and demands was optimized. Afterwards, a case study was performed involving polygeneration, where steam, chilled and treated water demands were addressed [53]. These scenarios demonstrate that the multi-period modeling scheme can be applied independently on the raw materials and on the final products of a supply chain, based on which of the two is fluctuating.

2.2 Mobile workforce management

The problem class of mobile workforce management is not a strictly defined term. It can refer to a broad range of instances where tasks are performed at spatially distributed places, which are visited by personnel, teams or machines which perform these tasks. For this reason, mobile workforce management involves scheduling decisions. On the other hand, because the order in which tasks are visited involves routing decisions, mobile workforce management is also strongly related to the Travelling Salesman Problem (TSP), and more generally, the Vehicle Routing Problem (VRP) when there are multiple actors and custom goals.

This section, dedicated to mobile workforce management, is therefore divided into three parts: review on methods and some studies for scheduling, then for VRP problems, and finally instances when these are in combination, which characterize mobile workforce management problems.

2.2.1 Scheduling problems

A scheduling problem involves the timing and sequencing of tasks, and assigning personnel, equipment units and other resources to these tasks if relevant. A common objective is the minimization of makespan, which is the time window for all events taking place, but other goals like profit or throughput maximization, cost or earliness and tardiness minimization can be considered. With the exception of a couple of specific problem classes, scheduling can be an NP-hard problem. For example, flow shop scheduling with multiple machines in general already falls into this category [54]. More complex problem classes can be tackled by heuristic approaches [55].

Many fundamentally different approaches like heuristics, combinatorial algorithms, and mathematical programming models exist for scheduling. For example, SA had been used for job shop scheduling [56], and GA have also been proposed [57]. Solution algorithms can be fine-tuned in many different ways to obtain better results [58]. Metaheuristic methods can cover a large search space of possible solutions, but on the downside, can also be very specific to a particular problem instance.

An example for combinatorial algorithmic approaches for scheduling is the S-Graph framework [59], which uses a directed graph for representing precedence relations and sequencing decisions to enumerate the search space of a scheduling problem. S-Graphs can be used to schedule batch processes. The framework can address different storage policies [60], or different timing constraints [61]. The approach can outperform mathematical programming methods [62], and is capable of eliminating cross-transfers which are a critical modeling issue for batch processing plants [63]. It is also possible to incorporate linear programming to the S-Graph framework [64], and to handle uncertain data [65]. Particularly, the S-Graph framework was also used to solve specific kinds of mobile workforce management problems [66].

Mathematical programming approaches are very common, either as standalone solutions or as part of some algorithmic framework - not only for scheduling, but a much broader range of problems. MILP is a good compromise between the efficacy of general solver tools and modeling power. Therefore, many approaches for specific scheduling problems are MILP models. A state-of-the-art review on different techniques was done by Mendez et al. [67]. One property of mathematical programming approaches is that equivalent but technically easier model formulations can be developed, which was demonstrated particularly for scheduling problems [68]. Multiple techniques exist to model the same process, but the best choice from these strongly depends on the actual problem and its size [69]. Note that although the mentioned approaches focus on batch process scheduling, the techniques can be adapted to general purposes.

Selection of the decision variables is the most important characteristic of an MILP modeling technique. Several concepts are used for example discrete time intervals [70], variable time points [71], unit-specific time points [72], time slots [73], and precedence relationships between tasks [74].

Time slots establish an assignment of tasks to sequencing decisions. A slot-based model had been proposed for multipurpose scheduling problems with different storage methods [75]. The scheduling of domestic appliances with time-based energy prices and subject to user preferences was solved by a slot-based MILP model [76]. These case studies suggest that time slots can provide a useful technique for other purposes as well.

2.2.2 Vehicle Routing Problems

In a VRP problem, the main decisions are the assignment of available vehicles to sites, and the order in which these sites are visited. Other constraints and options can be involved, for example time windows for each task and/or vehicle, precedence relationships between task executions, and multiple depots. A recent review of real-life considerations of VRP problems was done by Vidal et al. [77]. Another review from Moghdani et al. focuses on VRP problems involving green technology, targeted at reducing greenhouse gas emissions [78].

Standalone MILP models had been developed for both TSP and the more general VRP problems. Not only multiple vehicles, but multiple starting depots can be considered [79]. Algorithmic improvements had been proposed for a faster traverse of the search space of a VRP problem [80]. However, due to the complexity of the problems, a more common approach is the combination of mathematical programming tools with decomposition methods, metaheuristics, or some other algorithmic framework. The hybridization of metaheuristics and mathematical programming is referred to as matheuristics, which has a wide range of applications [81].

Transportation efforts, usually expressed as cost, distance or travelling time, are key factors in VRP problems. These can be estimated a priori, and sometimes depend on current vehicle load. An example is when travelled distances are weighted by passenger count [82]. The authors for this example formulate an MILP model, but solve it with a specific algorithm. In another instance, total mean time and their variance for vehicles were minimized by an application of the ACO metaheuristic [83]. Other goals were also formulated.

Time windows may exist for vehicles to arrive at certain sites. A possible scenario is when products have a limited lifespan, comparable to transportation times. Then, production can reasonably be the part of the decision problem. For time windows of perishable products, a nonlinear programming model was proposed, which is then solved by an adaptation of the Nelder-Mead method [84]. For a similar problem, a GA solution was also presented [85]. An MILP model was formulated for chemotherapy products and delivery [86]. Although a single vehicle was used multiple times, the computational complexity of the model turned out to be too high, therefore the authors applied the Benders decomposition method to solve the problem [87]. A similar problem for nuclear medicine delivery was solved by an MILP model, in this case, within a Large Neighborhood Search (LNS) framework [88]. A stochastic programming approach was presented for VRP for the case of passenger transport with airport buses, taking timing constraints for passenger groups into account [89]. An application of the PSO metaheuristic for VRP with time windows had also been proposed [90].

Another possible extension of VRP problems is vehicle capacity, which poses a significant limitation on the possible routes. The TS metaheuristic was used for an instance with vehicle capacities and timing constraints [91]. For larger problem sizes, the authors suggested a Lagrangian relaxation method.

Electric Vehicles (EV) and in general, alternative transportation and delivery technologies like drones are gaining increasing attention. Current EVs can usually be only effectively applied on short routes. In some cases, This motivation had a significant impact on the research of VRP problems. A variant of the LNS method based on an MILP model was proposed for solving the VRP problem for a fleet of EVs with time windows [92]. In another scenario, a standalone MILP model was proposed which considers multiple vehicle depots, time windows and different battery technologies as well [93], although this model was only tested on small scale problem instances, few vehicles and sites. The VRP problem for available drones as an alternative for delivery with trucks was also considered [94]. An MILP model had been proposed, which was later used by a branch and price algorithmic framework [95]. Solution methods were also proposed for scenarios were problem data contains uncertainty [96].

2.2.3 General problem formulations

There are many areas where mobile workforce must be managed involving both scheduling and routing decisions. Examples include the delivery of products, maintenance tasks at spatially distributed targets, or any kind of service requiring travelling to clients. A survey on the common characteristics, and most popular solution methods of mobile workforce management problems was done by Castillo-Salazar et al. [97].

Dependency of tasks on each other is a common trait of problems involving scheduling and routing of mobile workforce. Precedence relations are the most common example for such dependencies. Precedence requires that two events happen in a given order. This can be the execution of tasks, or more precisely, the completion of a task and the beginning of another. A typical example is when a procedure consists of multiple activities to be done in order. Then, there is a precedence relation for each consecutive pair of tasks representing these activities. Precedence can appear as constraints in simpler problems like TSP as well [98]. The vast majority of mathematical programming approaches fall into the category of precedence-based models, which capture the decisions on precedence of tasks as actual decision variables. Standalone models are uncommon, mathematical programming techniques are rather used in combination with some metaheuristic, resulting in matheuristic approaches [99]. For a mobile workforce management problem involving precedence relations, an approach using MILP modeling in an ACO framework was proposed [100]. The authors remark the difficulty of applying neighborhood search techniques when precedence relations as constrains are present in the problem. Other metaheuristics had also been applied, even in a standalone way [101].

Time windows, when appearing as constraints, also significantly increase the complexity of the problem. An MILP model was proposed to solve a workforce scheduling problem with time windows and precedence relations [102]. Note that the authors suggested an iterative solution algorithm.

Starkey et al. [103] solved a mobile workforce management problem where geographic places were grouped into worker areas, and then served independently by travelling engineers. This case study was motivated by the telecommunications field. The proposed solution methods was a GA framework involving fuzzy logic. More recently, the exact conditions on rearrangement of existing worker areas were investigated using a similar approach [104].

The mobile workforce can be grouped into teams, often it means a single vehicle which is treated as a single unit in the model. Decision on how to form these teams if the skills of the personnel are different is itself a difficult problem requiring heuristic approaches [105]. A scenario with different skills, multiple vehicle depots and the concurrent objectives of weighted total time and total cost was solved using a multi-stage heuristic approach relying on an MILP model [106]. The tendency of approaches shows that mathematical programming approaches are popular, but due to problem complexity, these are usually included in some algorithmic framework.

Chapter 3

Energy supply optimization with P-Graphs

3.1 Overview

In this chapter, a new P-Graph model for energy supply optimization is presented through a case study. The subject is a manufacturing plant for which heat and electricity demand can potentially be supplied from biomass or solar energy. The most profitable options for investment were to be determined. The novelty of the proposed model is that it features two approaches. First, the operating units for the pelletizer and biogas plant were modeled with flexible inputs, a technique which provides a precise control over equipment capacity and energy throughput, independent of each other. Second, the multi-period modeling scheme was applied to parts of the supply side and the demand side of the model, which better reflects total requirements than the corresponding single period counterpart. This model, and the techniques presented in general can be used for similar scenarios for energy supply optimization of a single facility.

The synthesis of supply chains of renewable energy sources, including solar, wind, hydropower and biomass sources is a challenging task in general, and has drawn attention in the recent years. A review by Nemet et al. focuses on energy planning and corresponding optimization [107]. Another recent review was made about system dynamics approaches for renewable supply chain management [108]. The complexity of these systems require the adequate modeling and optimization tools, regardless of whether a single processing facility or a whole regional supply chain is in scope. Mathematical programming approaches are a conventional and popular technique for this purpose, based on which many novel methods had been proposed [109]. Sharma et al. published a detailed review of mathematical programming models solely for the purpose of biomass supply chain modeling and optimization [110].

Specific case studies where problems are too large or have special properties may require other model developing techniques. The P-Graph framework has advantages over conventional approaches. Extending a P-Graph model with a new option can be much easier than for a mathematical programming model. The N best solution structures, which are supported by the current implementation, are also important in the investigation of the results. For these reasons, the P-Graph framework was chosen for modeling and optimization.

The work was published in two articles. The first article contains a basic P-Graph model for the case study and its solution [S2]. The second article contains the flexible input modification for the pelletizer and biogas plant equipment units, the multi-period model extension, and a detailed analysis [S3].

In Section 3.2, the basic properties and components of the model are presented. Section 3.3 is dedicated to the flexible input scheme and the multi-period extension of the model. Results are presented and discussed in Section 3.4.

3.2 Basic model components

The following are given for a manufacturing plant.

- Annual heating and electricity demand.
- Possible energy sources, including direct purchase, solar power and different types of locally available biomass, each with investment, operating and resource unit costs.
- The investment horizon.

The objective is the minimization of the total annual costs of the system, assuming that the annual heating and electricity requirements are met. The method was the modeling of the system as a single PNS problem forming a P-Graph superstructure of all energy options. This PNS problem was directly solved using the ABB algorithm. Individual solution structures were manually inspected.

Due to the long time elapsed since the data collection phase of this case study, some data could have become obsolete by now, especially energy prices and investment costs for each particular source. Other data are specific to this particular case study, especially energy demands and raw material availability. However, the proposed P-Graph model can be easily reused if supplied with the required data. It can even be extended by other energy sources not covered in this work, without tampering with the existing parts of the model.

First, the single period model is formulated, and then the extension into the multi-period variant is detailed. This section only covers the single period case.

Monetary values are expressed in HUF currency. At the time of data collection, 1 EUR was between 300-330 HUF.

Table 1: Available consumption data from the previous years for directly purchased natural gas, the only energy source for heating the plant at that time.

Natural gas (m^3)	2009	2010	2011	2012	2013	2014
January	133,999	128,744	157,085	123,770	75,782	48,635
February	123,836	95,406	137,103	124,305	51,407	49,067
March	120,326	77,536	123,795	83,362	43,560	16,847
April	37,378	58,464	83,305	61,092	15,452	4,337
May	35,057	63,719	51,009	37,482	2,785	4,247
June	37,065	52,094	30,924	17,340	1,919	2,688
July	30,396	44,485	31,560	12,891	1,554	2,416
August	34,232	44,628	30,105	20,179	1,534	2,117
September	28,607	81,730	30,024	19,829	3,072	2,136
October	82,299	105,612	74,841	25,235	4,208	10,982
November	105,599	104,195	125,638	50,535	24,273	43,769
December	116,459	156,139	129,481	73,819	57,240	62,139
Total	885,253	1,012,752	1,004,870	649,839	282,786	249,380

Table 2: Available consumption data from the previous years for electricity directly purchased from the grid, the only source for the plant at that time.

Electricity (kWh)	2009	2010	2011	2012	2013	2014
January	905,533	796,117	993,044	788,703	453,838	255,517
February	1,128,039	715,508	926,508	769,565	382,042	270,539
March	1,328,232	809,142	1,074,706	736,811	359,696	217,190
April	1,076,030	787,400	963,416	624,634	310,077	176,142
May	1,142,927	918,350	890,317	862,085	228,225	200,673
June	1,176,784	1,021,286	843,147	327,853	251,323	191,459
July	1,215,169	1,170,359	871,462	502,244	254,907	270,710
August	1,281,732	1,089,277	928,240	327,853	241,197	414,119
September	1,183,526	983,531	872,692	454,764	201,446	425,678
October	1,002,034	989,398	868,299	923,389	211,333	439,628
November	926,870	969,743	880,829	346,867	248,156	439,837
December	872,000	901,317	856,199	399,713	289,379	502,415
Total	13,238,876	11,151,428	10,968,859	7,064,481	3,431,619	3,803,907

3.2.1 Heating and electricity requirements

There are two demands that must be satisfied: annual heating and annual electricity requirements. These types of demands are modeled by two final product nodes in the P-Graph, each with a minimal inflow equal to the required amount. The actual amounts are estimated based on previous years, with the data and method described below.

Monthly consumption data for natural gas (see Table 1) and electricity (see Table 2) were available from the previous years. These were the only sources used by the plant to provide heating and electricity for production. The following observations can be made.

- Heating requirements are significantly higher in winter. This is not surprising because indoors heating requirements are concentrated in this season. This fact is a key motivation for the multi-period variant of the problem.
- Electricity consumption does not seem to depend on time, but rather on production load or other factors which were unavailable in this case study.
- Energy usage decreased gradually throughout the years, most likely due to the plant improving production efficiency. Note that this trend is not strict, as for the last year, 2014, electricity consumption was actually more than in 2013.

For these reasons, annual heating and electricity needs were determined. In both cases, Equation (3.1) shows the applied formula, which serves as a rough estimation for future years. The weighted average of the last three years are multiplied by a factor of 1.15 in case of a slight increase.

$$S_{future} = \frac{0.8 \cdot S_{2012} + 0.9 \cdot S_{2013} + 1.0 \cdot S_{2014}}{2.7} \cdot 1.15 \tag{3.1}$$

For the single period model, a heating requirement for 436,045 m^3 natural gas was obtained. Assuming a heating value of 34 $\mathrm{MJ/m}^3$ for natural gas, the total energy amount is 4,118,206 kWh. The unit kWh was used throughout the model for expressing not only electrical but heat energy amounts as well, for the sake of uniformity of the model components. The electricity requirement calculated in the same manner was 5,342,793 kWh. Note that other values are used by the multi-period model variant, which is explained later.

The obtained amounts were assumed to be the fixed demands for each future year. Note that the PNS problem can easily be resolved with different data if needed. It is also possible to assume different demands each year, but this option is not covered in this work.

Table 3: Raw materials in the case study, with unit price and annual availability.

Energy source	Unit price	Available
Natural gas	114 HUF/m^3	unlimited
Electricity	38 HUF/kWh	unlimited
Solar energy	free	unlimited
Saw dust	24 HUF/kg	150,000 kg
Wood chips	22 HUF/kg	600,000 kg
Sunflower stem	5 HUF/kg	500,000 kg
Vine stem	7 HUF/kg	600,000 kg
Corn cob	6 HUF/kg	1,200,000 kg
Energy grass	8 HUF/kg	1,600,000 kg
Wood	20 HUF/kg	2,000,000 kg

3.2.2 Energy sources

The following options were considered as energy sources.

- Purchasing natural gas from the provider and electricity from the grid directly.
- Solar power plant, which can contribute to both heating and electricity.
- Several types of biomass from local agriculture, from which biogas can be produced and subsequently burnt in either a furnace or a Combined Heat and Power (CHP) plant.

Based on these energy sources, raw materials are identified in the model, each having a unit price and total availability (see Table 3). Each raw material in the case study is represented by a raw material node in the P-Graph model. The current implementation for PNS problems in the P-Graph Studio software directly supports unit prices and maximum flows for raw materials.

The so-called business as usual solution for the problem consists solely of the direct purchase of electricity and natural gas. The plant management required these options to be available in the final solution as well, for safety reasons. Therefore, it is always possible to satisfy remaining demands by additional purchase. In the modeling point of view, even distribution is assumed for the supplies and the demands. Note that there is no point in overproduction, because selling energy surplus was not an option in the case study, although this could easily be modeled in the P-Graph framework. Purchase prices of 114 HUF/m^3 for natural gas and 38 HUF/kWh for electricity from the grid were assumed.

Solar power plants, in the modeling point of view, produce unlimited and free energy. All costs associated with solar energy are rather attributed to the investment and operating cost of the solar power plant instead.

Seven biomass types were available in the vicinity of the plant from agriculture. Collection of biomass from further distances would have been problematic in both

Table 4: Equipment units considered in the case study, each representing an investment with additional operating costs.

Equipment unit	Investment costs		Operating costs	
	fixed	proportional	fixed	proportional
Solar power plant	50 M HUF	353.98 HUF/kWh	none	22.12 HUF/kWh
Pelletizer	5 M HUF	10 HUF/kg	none	4 HUF/kg
Biogas plant	20 M HUF	240 HUF/kg	none	10 HUF/kg
Biogas furnace	10 M HUF	20 HUF/kWh	6 M HUF	4 HUF/kWh
Biogas CHP plant	20 M HUF	36 HUF/kWh	6 M HUF	6 HUF/kWh

the economical and the sustainability point of view. For each biomass type, unit price is expressed per kg, and there is an annual maximum available amount. It is assumed that there is no competition for these resources.

3.2.3 Possible investments

The main investments are equipment units for which decisions shall be made in the model. These are shown in Table 4, and some explanation is provided below. Investments represent the equipment units that make alternative energy sources possible to exploit.

There are four kinds of costs associated with each equipment unit.

- Fixed investment cost is paid once if the equipment is utilized, regardless of equipment capacity, and is divided equally in the whole investment horizon.

- Proportional investment cost is paid once, based on equipment capacity, and is divided equally in the whole investment horizon.

- Fixed operating cost is paid yearly if the equipment is utilized, regardless of equipment capacity.

- Proportional operating cost is paid yearly, based on equipment capacity.

The solar power plant had proportional cost data available in terms of produced kWh/y. The numbers shown are an estimation based on a single option, which required a 400,000 HUF investment cost, and 25,000 HUF operating cost, per 1,130 kWh/y capacity.

The pelletizer and the biogas plant had proportional costs given in terms of input material amount, expressed in kg/y, in contrast to the biogas furnace and the biogas CHP plant, which had proportional costs given in terms of the heating value of the biogas consumed, expressed in kWh/y.

3.2.4 Operating unit models

All transformations occurring on the path between raw materials and the final products are represented by operating units in the model. These include direct purchase and the mentioned equipment units for solar power generation and the biomass processing chain. Some equipment units in reality are modeled by multiple operating units in the P-Graph, and vice versa, a single operating unit node in the P-Graph may be a black box representation of multiple connected equipment units in reality.

P-Graph models of the components are now shown and explained. Each operating unit node in the resulting PNS problem requires two key decisions to be made: whether to use the operating unit in the solution structure, and if yes, what is the activity (also named volume or capacity) for that unit. Material flows are linearly scaled based on activity. Material balance constraints must hold for all material nodes. These are the most important properties of the underlying optimization model described below.

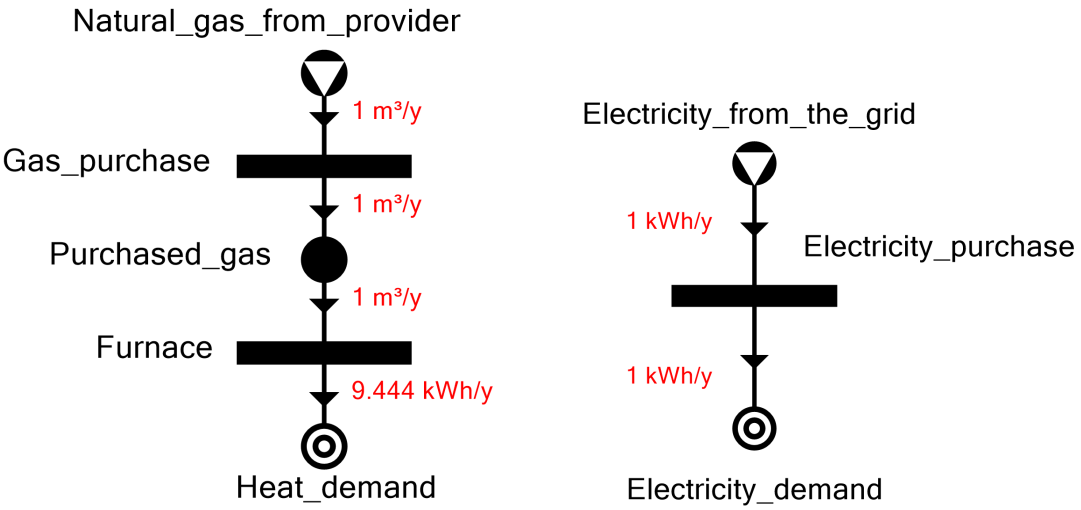


Figure 4: P-Graph model of the direct purchase of natural gas from the provider and electricity from the grid.

The direct purchase operations are represented by operating units each having one input and output (see Figure 4). The purchase of natural gas is implemented as a sequence of two operating units, one consuming the raw material natural gas and producing purchased gas, and the other representing the furnace in the plant responsible for heating. In theory, the intermediate purchased gas could be used for other purposes. The final product is the heating demand. The purchase of electricity from the grid is represented by a single operating unit which produces the electricity demand. Market prices are associated with the two raw materials.

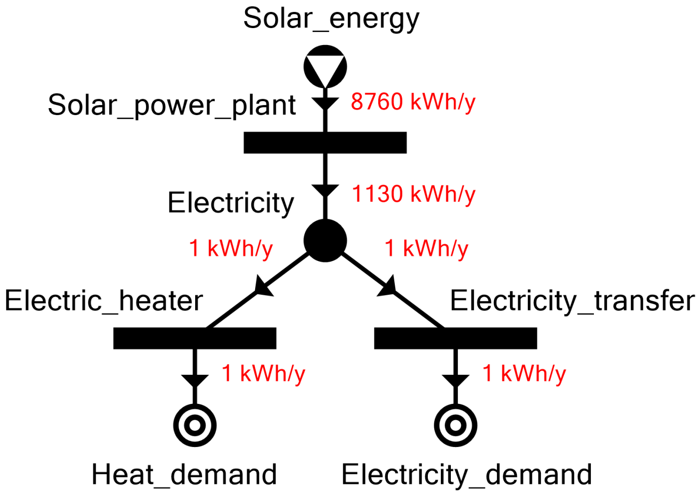


Figure 5: P-Graph model of the utilization of solar power in this case study.

Table 5: Biomass types available for energy production.

Biomass type	Heating value (kWh/kg)	Pelletizing needed
Saw dust	4.50	yes
Wood chips	4.25	yes
Sunflower stems	3.75	yes
Vine stems	4.10	yes
Corn cobs	4.00	no
Energy grass	4.80	no
Wood	4.16	no

The solar power plant was one of the alternative energy supply options considered in the model (see Figure 5), consisting of solar cells. A single operating unit is producing electricity from solar radiation. Flow rates for this unit, which are 8,760 kWh and 1,130 kWh represent the efficiency of this particular technology. Though, for the modeling point of view, the 8,760 kWh data is irrelevant, since solar radiation is unlimited and all costs are associated with the operating unit itself. Electricity produced this way can supply the electricity demand, or it can be used for heating if needed, despite the latter being inefficient.

The other source of energy considered in the case study is biomass, from which biogas is produced. This component is more complicated, because there are multiple different types of biomass, but some of these must be pelletized first before being fed into a biogas plant, while others can be directly fed. Data of the biomass types are shown in Table 5.

The initial attempt at modeling the processing chain of biomass is detailed in publication [S2], and can be summarized as follows (see Figure 6).

- The pelletizer is represented by an operating unit. It consumes biomass types requiring pelletizing and produces pellets.
- The biogas plant is represented by an operating unit. It consumes pellets and biomass types not requiring pelletizing and produces biogas.
- An input material node is introduced for collecting all inputs of the pelletizer. The same is done for the biogas plant.
- For each biomass type, a raw material node is introduced, which is consumed by a logical operating unit transforming the biomass into the input materials of the pelletizer and the biogas plant.
- All material nodes in this component, with the exception of the raw materials, are expressed in final heating power, in kWh.

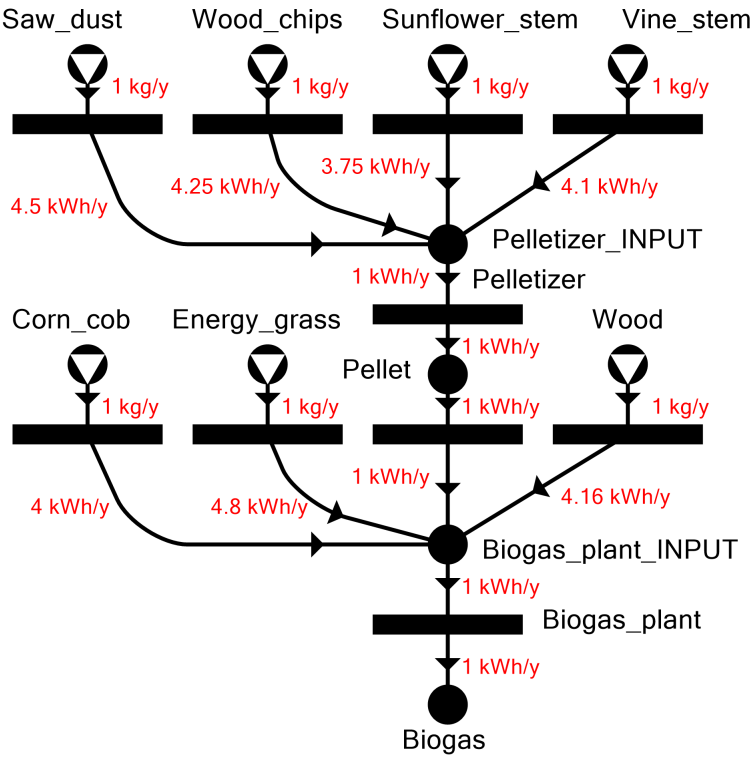


Figure 6: Initial model for the process of transforming the different biomass types into biogas.

The biogas can be consumed in either a biogas furnace or a biogas CHP plant, as modeled in Figure 7. The material node for biogas at this point expresses future heating power. The flow rates in this part of the model represent the final efficiency of the two technologies for satisfying heating and electricity demands. Note that energy losses are modeled in these flow rates, and can also be represented in the transformation rates for each biomass type.

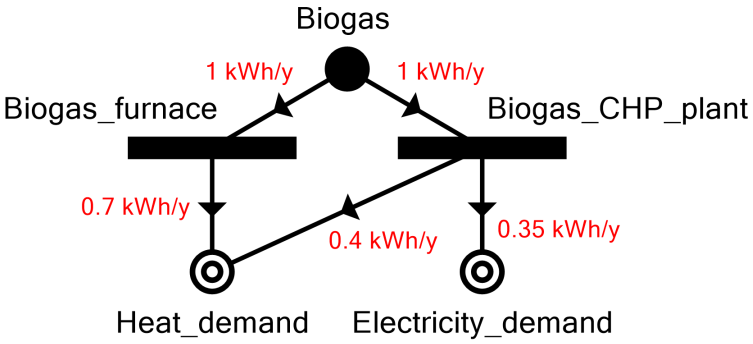


Figure 7: Model for utilization of biogas in a furnace or a CHP plant. Flow rates are adjusted to represent the final conversion factors from raw material nodes representing biomass amounts to the product nodes representing energy demands.

3.3 Specialties in the model

3.3.1 Inputs for biomass types

The initial model of the pelletizer and the biogas plant already makes flexibility available in terms of their inputs. This means that each input amount can be independently chosen, and be used in the same equipment unit. A single operating unit node is not adequate for such a model, as the flow rates of the inputs would force a fixed ratio on the input amounts. However, an operation with flexible inputs can be modeled by using multiple operating unit nodes, as demonstrated later.

The problem with this model, shown in Figure 6, is that the capacity constraints for both the pelletizer and the biogas plant should rather be expressed in terms of total mass of the input consumed, instead total energy content. This was not a serious modeling issue, for the following two reasons.

- Capacities, and implied operating costs are relatively low.
- The energy content of the considered biomass types are close in magnitude.

To make a more precise model, another approach was proposed instead, as shown in Figure 8. This approach allows more flexibility in terms of the input models. While arbitrary input ratios are still allowed, the constraints on the input capacities are made independent of the output amounts. The approach can be described as follows.

- Input material nodes are introduced for the flexible pelletizer and biogas plant models. For the flexible pelletizer model, 4 raw material input nodes are introduced for the biomass types requiring pelletizing. For the flexible biogas plant model, 7 input nodes are introduced: 3 raw material nodes for biomass types not requiring pelletizing, and 4 additional intermediate material nodes for the four types of pellets produced by the pelletizer.
- For each of the $4+7$ input nodes, an operating unit node is introduced, which produces the corresponding output node. These output nodes represent the pellet types for the pelletizer, and the biogas for the biogas plant.
- All material nodes except the final biogas node represent amounts in terms of total mass, not total energy content. The price and the total available amount of each biomass type is associated with the corresponding raw material node.
- Flow rates for energy content per mass are all represented in the production arcs starting from the operating unit nodes for each input, and ending in the biogas material node. All other arcs have a flow rate of 1.
- For both the pelletizer and the biogas plant models, a capacity material node is introduced which represents available input mass. This is consumed by all the $4+7$ input operating units with a flow rate of 1, or in other words, a flow rate equivalent to the mass of the corresponding biomass type consumed.
- The capacity material nodes are produced by two logical operating unit nodes representing the pelletizer and biogas plant. Production rates are equal to the activity of the pelletizer and the biogas plant. All of the investment and operating costs in the model are associated with these two logical operating unit nodes.

In this design, the constraint on total inputs for both the pelletizer and the biogas plant are expressed in terms of input mass, while output is expressed in terms of heating value. It can be seen that, in general, any non-negative weighted sum of the inputs could be used as a capacity constraint for total input amount. It would also be possible for the same equipment unit to include multiple such constraints, while the output also remains independent.

Further simplifications were made in the resulting model. Observe that the only use of pellets in the model is subsequently feeding them into the biogas plant. Therefore, the operating unit nodes producing the pellets were eliminated, and the pelletizer capacity is then consumed by the appropriate operating unit nodes representing the biogas plant inputs.

In this final design, depicted in Figure 9, there are seven raw material nodes, each representing a biomass type, and have unit prices and total availability associated. For each raw material node, there is a logical operating unit node creating biogas, in an output ratio representing heating value and possibly energy conversion losses.

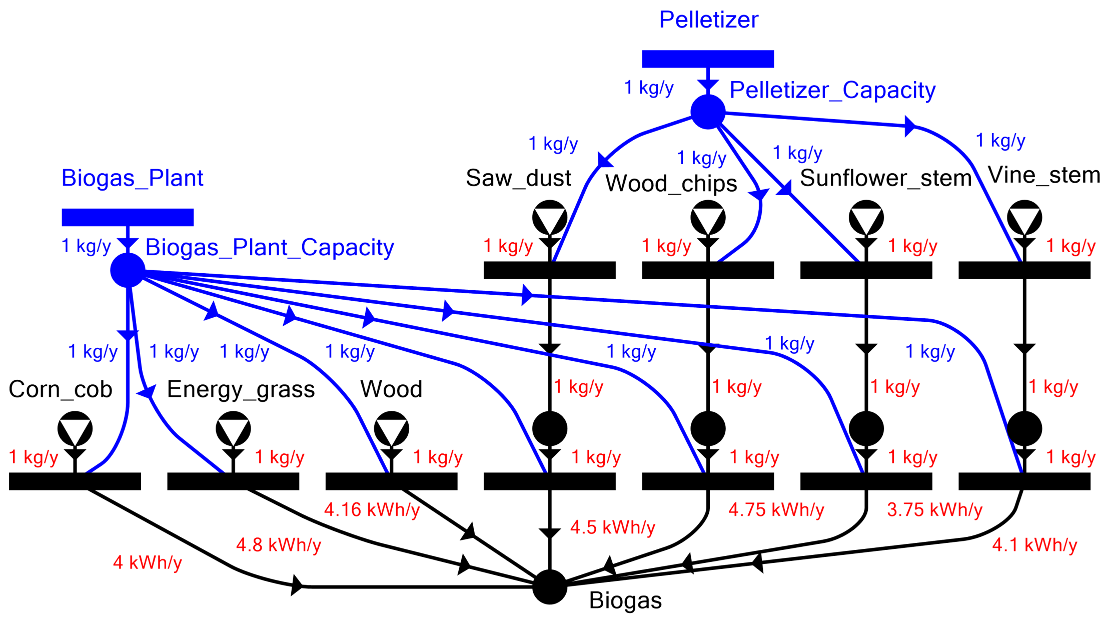


Figure 8: Modified model of the biomass processing chain, which captures equipment unit capacities more precisely. The blue parts represent the newly added elements.

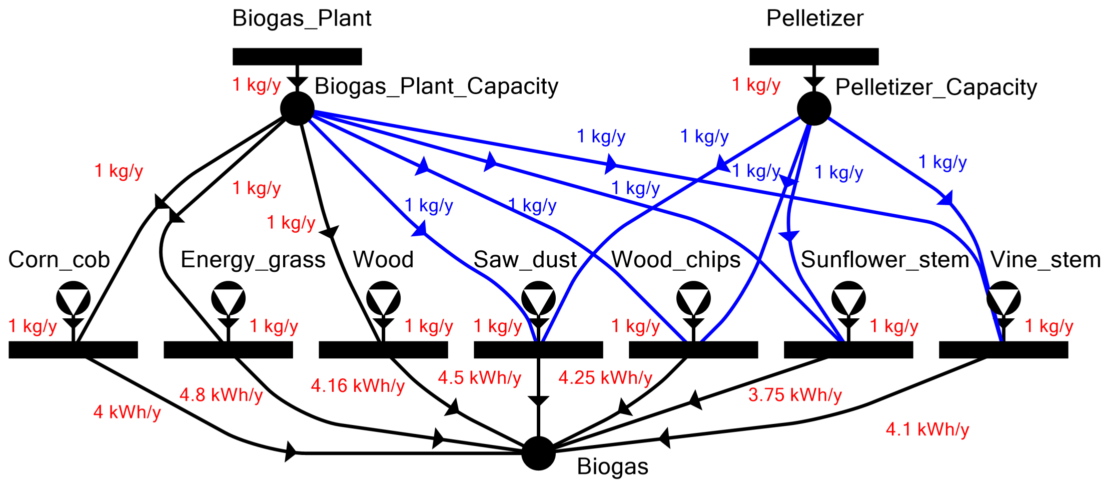


Figure 9: Simplified version of the biomass processing chain model, which is the final form used in the case study. The blue arcs represent the simplification.

3.3.2 Multi-period extension

So far in this chapter, the components of the single period model have been presented. The final, single period P-Graph model is shown in Figure 10, which has two issues significantly affecting the continuity of the demands or the supplies.

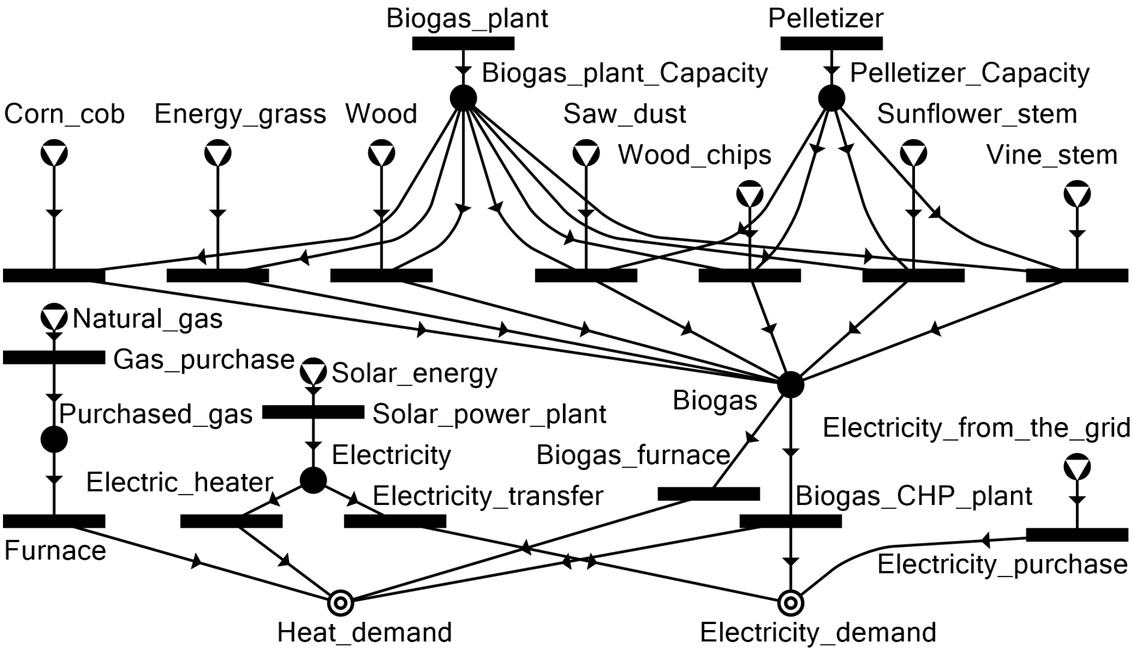


Figure 10: Final form of the single period model for the energy consumption optimization of the manufacturing plant.

- Heating demands become higher in the winter months.
- More solar energy can be collected in summer than in winter.

These can be addressed by dividing the single period, which is a year of operation, into smaller periods. Each of the periods has its own supply and demand amounts. In this case study, the following two periods are introduced.

- The **winter** period consists of the months December, January and February.
- The **mid-year** period consists of the other months.

The months are assumed to be of the same length, which means that the ratio of period lengths is exactly 1 : 3. Note that the winter period is split among two calendar years.

Note that it would not be adequate to formulate and solve the model for the periods separately, as investment decisions, particularly equipment sizing are shared among all periods. Therefore, all periods should be considered in a single optimization model.

These observations provide the main motivation to apply the multi-period modeling scheme in this case study. The meaning of multi-period extension is that instead of assuming a balanced supply and/or demand throughout a year, balance throughout each of the periods is assumed. The rates can vary among periods,

Table 6: Heating and electricity demands for the winter and the mid-year period, calculated from natural gas (see Table 1) and electricity consumption (see Table 2), using the formula shown in Equation (3.1).

Energy source	Period	Calculated demand
Heating (natural gas)	winter	187,585 $\mathrm{m}^3 \approx$ 1,771,637 kWh
	mid-year	248,460 $\mathrm{m}^3 \approx$ 2,346,569 kWh
	total	436,045 $\mathrm{m}^3 \approx$ 4,118,206 kWh
Electricity	winter	1,536,566 kWh
	mid-year	3,806,227 kWh
	total	5,342,793 kWh

which can be used to model supply and demand fluctuations. More periods could lead to more accurate but also larger models.

Note that biomass availability is also typically fluctuating over the year, so multiple periods could also be introduced for biomass distribution. This is not considered in this case study. Instead, it is assumed that biomass supplies can be distributed evenly for consumption (with the help of storage for example).

Although only heating is targeted, it is straightforward to also include the electricity demand into the multi-period scheme. Therefore, the total heating and electricity demands are both split into winter and mid-year demands, resulting in four distinct demands. The estimation values of these demands are calculated by the same Equation (3.1) used for the single period case, but only for the data corresponding to the period. Values obtained are shown in Table 6. Note that the sum of demands for the two periods are equal to the original, single period demand for both heating and electricity. Also note that the heating demand in winter is proportionally more than mid-year, although less overall, because winter is 3 times shorter than the mid-year period.

Modeling production capacity of equipment units is a key factor in the multi-period model. The pelletizer, biogas plant, biogas furnace and biogas CHP plant all realize $\frac{1}{4}$ of their capacity in winter and $\frac{3}{4}$ of their capacity mid-year. This corresponds to the ratio of periods lengths due to the fact that these equipment units are uniformly productive throughout the year.

Different ratios should apply to the solar power plant. Solar radiation is where the supply side is significantly different in the two periods. In the single period model it was already assumed that in certain circumstances, solar supply can be considered evenly distributed in the modeling point of view. This was a simplification, which is now relaxed so that supply is assumed to be even in each of the periods individually. The constant λ is introduced to express the ratio of the throughput of the solar power plant in the mid-year and the winter periods, per unit time. As a rough estimation, we use $\lambda = 2$.

All of these statements lead to the exact determination of how the solar power plant realizes its capacity in the two periods, see Equation (3.2) and Equation (3.3).

Winter has 3 months and the mid-year period has 9 months, these are weighted by the productivity of the solar power plant, which is 1 in winter and λ in the mid-year period. The total productivity is split in the obtained ratio. Therefore, the solar power plant realizes $\frac{1}{7}$ of its production capacity in winter and $\frac{6}{7}$ of its production capacity mid-year.

$$E_{winter} = E_{annual} \cdot \frac{3 \cdot 1}{3 \cdot 1 + 9 \cdot \lambda} = \frac{1}{7} \cdot E_{annual} \tag{3.2}$$

$$E_{mid-year} = E_{annual} \cdot \frac{9 \cdot \lambda}{3 \cdot 1 + 9 \cdot \lambda} = \frac{6}{7} \cdot E_{annual} \tag{3.3}$$

The meaning of this part of the multi-period extension can be translated as follows: instead of assuming a 353.98 HUF investment and 22.12 HUF operating costs per 1 kWh capacity throughout the year (as in Table 4), we assume the same costs per $\frac{1}{7}$ kWh capacity in winter and $\frac{6}{7}$ kWh capacity mid-year, which can be arbitrarily scaled but cannot be decoupled.

The final multi-period model, shown in Figure 11, is constructed as follows.

- The single-period model is duplicated, one instance is for each period.
- The raw material nodes and the capacity-producing logical operating unit nodes for the pelletizer and biogas plant are kept common and not duplicated. The logical operating unit nodes produce two logical capacity materials, one for each period.
- For each of the biogas furnace, biogas CHP plant and solar power plant, another capacity-producing logical operating unit node is introduced as a common node for the two periods, which produce two different logical capacity materials, one for each period.
- The common operating unit nodes produce their logical capacity materials for the winter and mid-year periods in flow rates of $\frac{1}{4}$ and $\frac{3}{4}$, with the exception of the solar power plant, for which these rates are $\frac{1}{7}$ and $\frac{6}{7}$.
- Raw material costs and availability are associated with the common raw material nodes for biomass types.
- Investment and operating costs are associated with the common operating unit nodes.
- The final heating demands in each period are represented by four product material nodes with the flow requirements shown in Table 6.

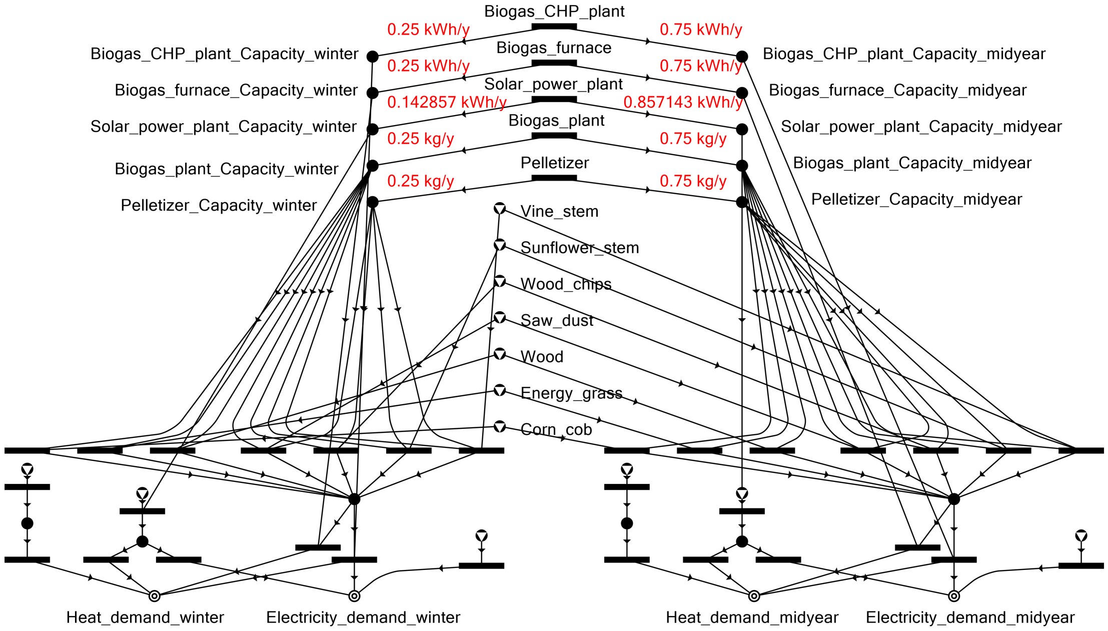

Biogas_CHP_plant
0.25 kWh/y
0.75 kWh/y
Biogas_CHP_plant_Capacity_winter
Biogas_furnace
Biogas_CHP_plant_Capacity_midyear
0.25 kWh/y
0.75 kWh/y
Biogas_furnace_Capacity_winter
Solar_power_plant
Biogas_furnace_Capacity_midyear
0.142857 kWh/y
0.857143 kWh/y
Solar_power_plant_Capacity_winter
Biogas_plant
Solar_power_plant_Capacity_midyear
0.25 kg/y
0.75 kg/y
Biogas_plant_Capacity_winter
Pelletizer
Biogas_plant_Capacity_midyear
0.25 kg/y
0.75 kg/y
Pelletizer_Capacity_winter
Pelletizer_Capacity_midyear
Vine_stem
Sunflower_stem
Wood_chips
Saw_dust
Wood
Energy_grass
Corn_cob
Heat_demand_winter
Electricity_demand_winter
Heat_demand_midyear
Electricity_demand_midyear

Figure 11: Final form of the multi-period model for the case study, which represents heating demands and solar energy supplies more accurately.

3.4 Results and discussion

Both the single period and the multi-period models were implemented using the P-Graph Studio software (version 5.2.3.1) as PNS problems. Although multi-period models are now directly supported [23], the multi-period instance was manually constructed. Both models were solved using the ABB algorithm, assuming first a 20 years long investment horizon, and then lower, 10 years long and 5 years long horizons were set. Total annual costs were to be minimized.

The ABB algorithm ran on a Lenovo Y50-70 laptop with an Intel i7-4710HQ processor and 8 GB RAM. Solution was almost instant, at most 1 s for the single period, and at most 2 s for the multi-period model. This was as expected due to the relatively small model sizes. The MSG algorithm did not exclude any part of the PNS problem, since only relevant options were included in the first place. A couple of the top reported solution structures were manually investigated. The results and the most important findings are discussed below.

The PGSX files implementing the models, and all other materials regarding the case study are available as supplementary materials in the main covering publication [S3], and on the web [111].

3.4.1 Single period model

The 10 best solution structures for the single period model are shown in Table 7, with utilized technologies and biomass types indicated. Note that the pelletizer and

Table 7: Top 10 solution structures for the **single period** model, in order of total costs, with biomass types and decisions on key investments shown. Columns: Cc=corn cobs, Eg=energy grass, Wd=wood, Sd=sawdust, Wc=wood chips, Ss=sunflower stems, Vs=vine stems, Eh=electric heating from solar power, Et=electricity from solar power, Bf=biogas furnace, Bc=biogas CHP plant, Pg=purchase of natural gas, Pe=purchase of electricity.

Total costs M HUF/y		Cc	Eg	Wd	Sd	Wc	Ss	Vs	Eh	Et	Bf	Bc	Pg	Pe
1.	220.709	X	X									X		X
2.	224.057		X				X	X				X		X
3.	224.325		X	X				X				X		X
4.	224.357		X					X				X	X	X
5.	224.496		X		X			X				X		X
6.	224.526		X			X	X					X		X
7.	225.895		X	X			X					X		X
8.	226.049		X				X					X	X	X
9.	226.380	X	X							X		X		
10.	226.723		X		X	X	X					X		X

the biogas plant are omitted from the table, since the presence of these two units are a direct consequence of the presence of their respective possible input materials.

The optimal solution is 220.709 M HUF/y, relying on corn cobs and energy grass, used in a biogas CHP plant, and purchasing the rest of the electricity demand directly. There is a small jump in terms of objective to the second best structure, but the difference between the 1st and 10th best structures is still below 2.72 %. The distribution of technologies in these solutions is the following.

- The biogas CHP plant seems to be the key decision leading to these solutions. The biogas furnace does not appear, despite being cheaper. This suggests that the ability of the CHP plant to produce electricity is valuable.

- All 7 biogas types are present, and each solution uses 2-4 of these. The total availability, cost and pelletizing requirement cause the subtle differences in the solution structures.

- Energy grass seems to be the best of the biomass types, because it is included in all of the shown solution structures.

- In the 4th and 8th structures, natural gas is also purchased.

- The 9th structure is fundamentally different from all others, because solar power is used here to cover the remaining electricity demand. In all other solutions, purchase of electricity from the grid is used instead. This suggests

Table 8: Top 10 solution structures for the **multi-period** model, in order of total costs, with biomass types and decisions on key investments shown. Columns: Cc=corn cobs, Eg=energy grass, Wd=wood, Sd=sawdust, Wc=wood chips, Ss=sunflower stems, Vs=vine stems, Eh=electric heating from solar power, Et=electricity from solar power, Bf=biogas furnace, Bc=biogas CHP plant, Pg=purchase of natural gas, Pe=purchase of electricity. Cell values: W=winter, M=mid-year, X=both periods.

Total costs M HUF/y		Cc	Eg	Wd	Sd	Wc	Ss	Vs	Eh	Et	Bf	Bc	Pg	Pe
1.	228.942	X	X									X	W	X
2.	228.986	M	X									X	W	X
3.	229.205	W	X									X	W	X
4.	229.358		X					X				X	W	X
5.	229.362	W	X					M				X	W	X
6.	229.363	W	X				M	W				X	W	X
7.	229.366	W	X				M					X	W	X
8.	229.378		X				W	M				X	W	X
9.	229.385		X	X								X	W	X
10.	229.391		X				X					X	W	X

that solar power is roughly as effective as purchasing electricity, provided that the data in this case study are accurate.

It should be noted that due to a management requirement, natural gas and electricity purchase from the grid are maintained anyway as options, to ensure energy safety. This happens even if these units are not included in the optimal solution. In the modeling point of view, this is not a problem. In fact, the practical robustness of a solution structure is increased this way.

3.4.2 Multi-period model

Again, the 10 best solution structures for the multi-period model are presented in Table 8. It is also shown whether an option is used in winter, mid-year or both.

The optimal solution for the multi-period model is depicted in Figure 12. The activity of different energy sources according to the optimal solution is shown in Table 9. The objective is 228.942 M HUF/y. This best solution is similar to the one obtained for the single period model: corn cobs and energy grass are consumed, and a biogas CHP plant is utilized. The purchase of natural gas is needed, but only in winter. The purchase of additional electricity is needed in both periods. The following observations can be made about the top solutions.

- Despite the similarity in structure, the multi-period optimum is 3.73 % worse. This is expected, because the multi-period model is more restrictive. Demands

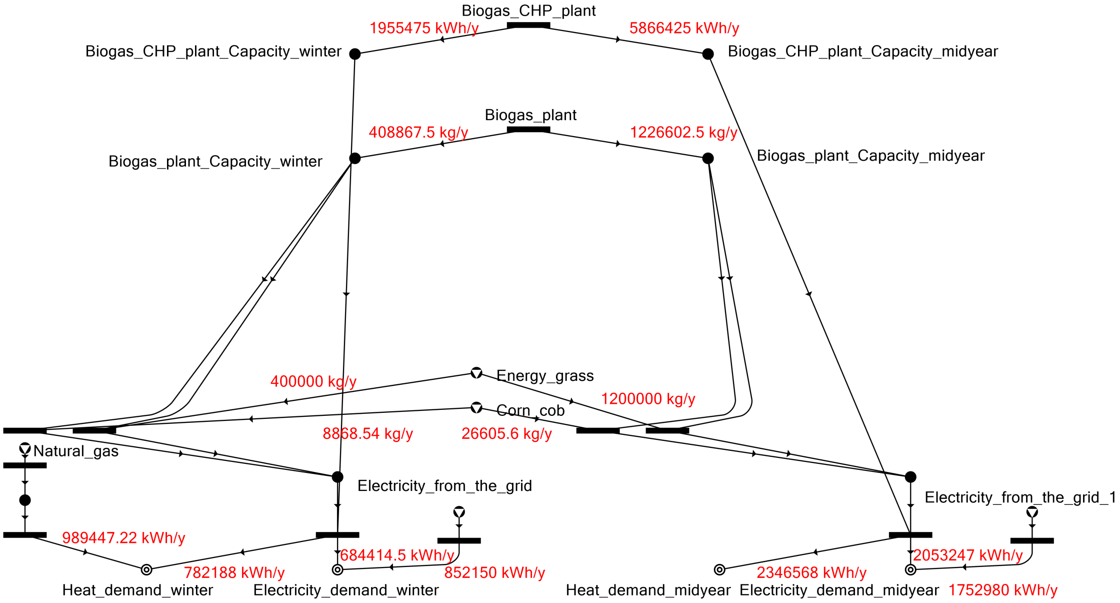


Figure 12: Optimal solution structure for the multi-period model, assuming a 20 years long investment horizon.

are higher when must be met individually for each period, than only in total, so more capacity is required.

- There is much less variability in the solution structures than in the single period case. For example, the first three structures differ only in the decision whether corn cobs should be used in both periods (1st), in mid-year only (2nd) or in winter only (3rd). The difference between the 1st and 10th solutions is minimal, 0.2 %. The reason behind these phenomena is that the multi-period model has combinatorially more options for small structural differences. If more structures were investigated, the ones analogous to the first 10 solutions

Table 9: Overview of the energy supply distribution in the optimal solution of the multi-period model.

Period	Energy source	Usage (annual)	Cost (annual)	Contribution	
				Heat	Electricity
Winter	Corn cobs	8,869 kg	53,214 HUF	0.80 %	0.81 %
	Energy grass	400,000 kg	3,200,000 HUF	43.35 %	43.73 %
	Natural gas	104,765 m^3	11,943,200 HUF	55.85 %	N/A
	Electricity	852,150 kWh	32,381,700 HUF	N/A	55.46 %
Mid-year	Corn cobs	26,606 kg	159,636 HUF	1.81 %	0.98 %
	Energy grass	1,200,000 kg	9,600,000 HUF	98.19 %	52.97 %
	Natural gas	0 m^3	0 HUF	0.00 %	N/A
	Electricity	1,752,980 kWh	66,613,200 HUF	N/A	46.06 %

of the single period model would be expected to turn up.

- Energy grass is still present in all cases, but only 4 other types of biomass appear as well. It is also a general rule that additional electricity is purchased in both periods, but natural gas is only purchased in winter.

Overall, the ability of the multi-period modeling scheme is demonstrated in these solution structures for formulating accurate models and finding the best solutions. The multi-period case is more restricted because constraints must hold in each period individually, but there is also more flexibility, as resources might be used independently in each period. It is also shown that the energy supply can be fundamentally different in mid-year compared to winter, even though exactly the same investments are available in both.

3.4.3 Investment horizon

The costs of the used technologies have two components: investment costs, which are paid once, and operating costs, which are paid yearly. The investment horizon is used to annualize the investment costs, by setting up a fixed time span in the model for which the most profitable solution is to be found. Note that this is not equivalent to the payback time of an equipment unit, since the investment horizon is a single parameter considered for the whole model.

Since short-term investments are preferred, the value of the investment horizon is a critical modeling parameter. Shorter horizons of 10 years and 5 years were also investigated for both the single and the multi-period model. The objective values for the two best solutions in each case are shown in Table 10. The main observations are the following.

- For short horizons, the business as usual solution turns out to be the best. This means purchasing natural gas and electricity from the grid and not doing any investments. This particular structure results in exactly the same solution in both the single period and multi-period case, with the same objective value regardless of the investment horizon, which is 252.735 M HUF/y.

- The second best solution for the 10 years long horizon is 6.15 % worse than the business as usual solution in the single period model, and 4.71 % worse in the multi-period model. These second best solutions are very similar to the best ones observed for the 20 years case. The multi-period solutions are actually equivalent: energy grass and corn cobs are used in a CHP plant, purchasing additional natural gas in winter and electricity in both periods. The much worse objective value is due to the decreased investment horizon.

- The situation is even worse for the 5 years long horizon. With respect to the business as usual solution, the second best structure is 35.7 % worse for the single period model, and is 28.27 % worse for the multi-period model. The

Table 10: The two best solutions for different investment horizons assumed for the single period and the multi-period model.

Investment horizon	Model	Total costs M HUF/y		Comment
20 years	single period	1.	220.709	similar to multi-period
		2.	224.057	
	multi-period	1.	228.942	same as for 10 years, Solution 2
		2.	228.986	
10 years	single period	1.	252.735	business as usual
		2.	268.288	similar to multi-period, Solution 2
	multi-period	1.	252.735	business as usual
		2.	264.647	same as for 20 years, Solution 1
5 years	single period	1.	252.735	business as usual
		2.	342.985	similar to multi-period
	multi-period	1.	252.735	business as usual
		2.	324.184	similar to single period

solutions in these cases are fundamentally different than reported for longer horizons, because the biogas furnace is used here, relying on energy grass. The interpretation of this result is that if we are forced to make some investment (which is the case when not the business as usual solution is considered), the most sensible option is the cheapest one, which is the biogas furnace.

These results show that the investment to sustainable energy sources makes sense for a long investment horizon. In practice, much shorter investment horizons are considered. However, this finding is for this particular case study only. The situation can be different for other plants and systems, and in other environments. New options may become available, existing technologies may significantly improve, and new incentives can change the profitability of the same investment. Nevertheless, a P-Graph model was developed for the purpose of accurately modeling such scenarios and solving them to optimality, which is technically easy to be adapted to the aforementioned situations.

3.5 Thesis summary

Thesis 1. A P-Graph model was developed and its effectiveness demonstrated, for the purpose of energy supply optimization involving several biomass types and solar power generation as alternatives to purchasing natural gas and electricity, on the scale of a manufacturing plant. The model can be easily adapted to similar energy supply optimization problems.

Related publications: [S2], [S3].

T1.1. New operating unit models were proposed for the pelletizer and biogas plant equipment units, which allow flexible, independent inputs of several biomass types at the same time. In the PNS implementation of the models, capacity is calculated based on mass, which is different than heating value. This results in a more accurate model, which was demonstrated in the case study of the manufacturing plant.

T1.2. A multi-period extension of the model was performed, which represents seasonal solar power supply and energy demands more precisely, by the distinction of two periods: winter and mid-year. Results from solving this model indicate that biomass and solar power utilization can be economical, but a long investment horizon was needed in this particular case study.

Chapter 4

Operations with flexible inputs

4.1 Overview

In this chapter, a general method is presented to model operations with flexible inputs and arbitrary linear constraints on these inputs in the P-Graph framework. This model can be used directly as part of a PNS problem for optimization with already existing software like the P-Graph Studio and procedures like the ABB algorithm.

In a P-Graph, the capabilities of an operating unit node are described by two sets of material nodes: the inputs and the outputs of the operating unit. The presence of the operating unit in a solution structure requires the presence of all of its inputs, and establishes the presence of its outputs. A straightforward way of using a P-Graph for optimization is by assuming two decision variables for an operating unit in the model: a binary variable for deciding whether the operating unit is included or not, and a continuous variable determining the activity of the operating unit. The activity determines the consumption and production amounts for each material if the unit is active. This is the model used by the current implementation of the P-Graph Studio and corresponding solver with the ABB algorithm for PNS problems. Note that the P-Graph framework itself can be used with more complex, even nonlinear operating unit models.

The activity of the operating unit node in the aforementioned model determines all input and output amounts, each with an individual scale factor. Consequently, the ratio of the input material amounts for a single operating unit is fixed. This is acceptable for modeling some processes, for example chemical reactions, material transfers and conversion. But in some cases, the connection between inputs can be more flexible. For example, a furnace producing heat can be fed with different combustibles, and their ratio can be arbitrary. This phenomenon was also present in the case study detailed in Chapter 3 for the pelletizer and the biogas plant [S3]. In fact, this was an ad-hoc application of the flexible modeling scheme, the generalization of which is shown in this chapter. The solution for the problem is that not a single, but multiple operating unit nodes are used for modeling an operation with flexible inputs. In the new model the input amounts can be completely independent, but are usually in relation by some constraint, for example a total capacity limit. Besides, other constraints may exist which further limit feasible input amounts.

The goal in this chapter was to propose a general and systematic method for modeling operations with flexible inputs in the P-Graph framework. There are extensions of the P-Graph Studio software which support a particular modeling scheme, for example the TCPNS for scheduling [17], or multi-period models [23]. In particular, there had been PNS models proposed containing operating units with flexible inputs [19]. In this latter example, the authors manually extended the underlying MILP model of a PNS problem to allow flexibility. The novelty of the approach presented in this chapter is that the flexible modeling scheme is achieved as part of an ordinary PNS problem. Consequently, the existing software tools and algorithms, for example ABB can then be used on the resulting model formulation without any further software support or manipulation of the optimization procedure.

The flexible modeling scheme is achieved in two steps. First, completely independent input amounts are allowed, and then a method is provided for including arbitrary linear constraints for these amounts. In Section 4.2, P-Graph model components are presented in logical order, leading to the general scheme. Some additional approaches are also presented, and the behaviour of these model components regarding solution structures is also analyzed. The case study from Chapter 3 is revisited for demonstrating how a flexible operating unit can be extended with a linear constraint on the input biomass types. This demonstration is shown in Section 4.3. The main publication for this chapter contains the flexible modeling techniques and the demonstration [S4].

Another case study was made, for optimizing the transportation and processing of biomass to produce heat and electricity in a small rural region. The basis of this case study is an initial approach for the problem where fermenter units had fixed input ratios. The solution is reproduced as both an MILP model and a PNS model, where fermenters are modeled using the flexible input scheme. This case study does not only show the advantage of a flexible operation model, but also demonstrate the new technique introduced in this chapter. The case study is summarized in Section 4.4, and a more detailed presentation is given in Appendix A. Most results are already published [S6], but the PNS model formulation is to be published in the future.

4.2 Flexible operating units

The flexible operating unit model is designed to be used within the linear optimization model of a PNS problem, supported by the current implementation of the P-Graph Studio software. The structure of this optimization model is described by a P-Graph, which implies the solution structures. Additional data and constraints are included in the optimization model, which determine exact material amounts in the process, define the objective, and may prevent solution structures from being actually feasible.

These additional data and constraints are the following. Note that there are more supported or possible features, but only the listed ones are relevant in the flexible operating unit model.

- Each material may have a cost or value, associated with the material node.
- Material nodes may have material balance constraints. These may define a minimum and maximum value for the difference between input flows and output flows. Only raw materials may have a negative balance.
- Each arc in the graph has a flow rate, defining the input or output flow rate for the corresponding operating unit and material node. A material flow is obtained as the flow rate multiplied by the activity of the operating unit node.
- Operating unit nodes have fixed and proportional investment and operating costs. The fixed costs arise if the operating unit node is used. The proportional costs arise based on the activity of the operating unit node.
- Operating unit nodes also have a capacity associated, which is a constant upper bound for activity.
- Determine a selection of operating unit nodes and their activity resulting in optimal annual cost or profit.

The flexible operating unit is actually a model of an operation obtained as a collection of operating unit nodes and material nodes in the P-Graph. This model may represent a single equipment unit in reality which accepts an arbitrary combination of its possible inputs. This flexible operating unit model, or flexible operation in short, is now formally described.

Suppose that an operation has n input materials, named $A_1, A_2, \ldots, A_n$, and k output materials, named $B_1, B_2, \ldots, B_k$. The total amount of materials consumed are denoted by variables. For each input material A_i, the operation consumes the exact amount of x_i, and for each output material B_j, the operation produces the exact amount of y_j. Then, the operation can be modeled as a flexible operating unit if the values x_i and y_j representing a feasible realization of the operation can be characterized as follows.

- $x_i \geq 0, \forall 1 \leq i \leq n$.
- $y_j \geq 0, \forall 1 \leq j \leq k$.
- Constant coefficients $\nu_{i,j} \geq 0$ exist for all $1 \leq i \leq n$ and $1 \leq j \leq k$ such that:

$$y_j = \sum_{i=1}^{n} \nu_{i,j} x_i \quad \forall \ 1 \leq j \leq k \tag{4.1}$$

- There can be constraints for the input amounts, each of the following form:

$$\sum_{i=1}^{n} \lambda_i x_i \leq C \tag{4.2}$$

For the sake of simplicity, it is assumed that there is a single output material B for the operation ($k = 1$). Then, ν_i denotes the coefficient regarding input material A_i and B, and y denotes the total produced amount of B. Since each output is independently determined by the input amounts and output amounts do not take part in constraints, no generality is lost with this assumption.

Constraints on the input amounts are also linear. Note that λ_i and C are not assumed to be nonnegative in Inequality (4.2). An equivalent formulation, using only nonnegative coefficients C_{MIN} and C_{MAX}, and only positive coefficients λ_i is shown in Inequality (4.3). Here L and R are disjoint index sets for the left-hand side (LHS) and the right-hand side (RHS) of the inequality. Therefore, any input can be present at either but not both sides. Also, at most one of C_{MIN} and C_{MAX} needs to be positive, the other one can be assumed to be zero. This formulation will be used later in the modeling procedure.

$$\begin{gathered} C_{MIN} + \sum_{i \in L} \lambda_i x_i \leq \sum_{i \in R} \lambda_i x_i + C_{MAX} \\ C_{MIN}, C_{MAX} \geq 0, \ \min\{C_{MIN}, C_{MAX}\} = 0 \\ \lambda_i > 0 \ \forall \ i \in L \cup R \end{gathered} \tag{4.3}$$

In the following subsections, models for flexible operations are presented for particular modeling goals. Since each model consists of multiple operating unit nodes, it is important to determine what subset of these can be present in a solution structure. A good feature of the model is when infeasible subsets are excluded from solution structures early by the MSG algorithm, as it makes later steps of the solution procedure faster. On the other hand, a model may result in multiple subsets representing the same or very similar solutions as different solution structures. This makes the model structurally redundant, which is a disadvantage.

A small list of symbols for the upcoming models is provided in Table 11.

4.2.1 Single operating unit node

As a starting point, consider an operation modeled by a single operating unit node. Input materials are consumed in a fixed ratio. Therefore, input amounts x_i and the output amount y are all directly proportional to the single variable x representing the activity or volume of the operation performed, as shown in Equation system (4.4).

$$\begin{aligned} x_i &= a_i x \quad \forall \ 1 \leq i \leq n \\ y &= bx \end{aligned} \tag{4.4}$$

Table 11: Symbols used in the operation model descriptions.

a_i	Flow rate in a single operating unit node for input material A_i.
b	Flow rate in a single operating unit node, for the single output material B.
λ_i	Constant coefficient for input material A_i used in linear constraints.
n	Number of input materials.
ν_i	Linear contribution factor of input material A_i to the amount of the single output material B.
x	Activity or volume of a single operating unit node.
x_i	Activity or volume of the operating unit node introduced for input material A_i, which determines the amount of material A_i consumed by the modeled flexible operation.
y	Amount of output material B produced.

Coefficients a_i and b are not only constants describing the operation, but the exact flow rates of the single operating unit used for modeling (see Figure 13).

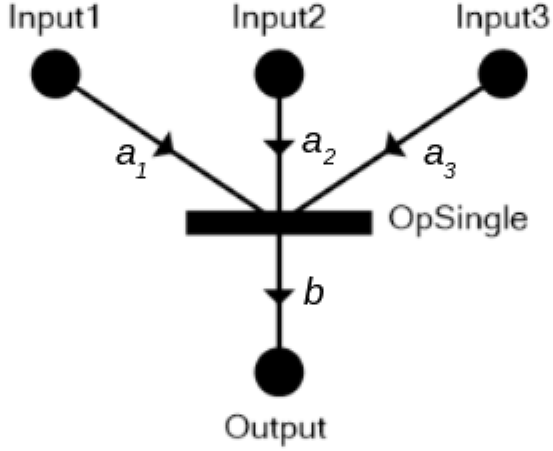


Figure 13: Modeling inputs with fixed ratios by a single operating unit node.

A solution structure may either include this single operating unit node or not, depending on whether the operation is included in the solution. Both options are relevant, therefore no redundancy is introduced in the model.

4.2.2 Independent inputs

In some scenarios, there can be multiple input materials that are not subject to a fixed ratio, but can be independently used. For example, a furnace may consume different combustible materials A_i, each having a unique contribution ν_i to the total heating power, which is the single output of the furnace.

The operation with multiple independent inputs can be described by Equation (4.5). Here, not a single variable x is used, but a variable x_i for each independent input material. Inputs have different contributions ν_i to the output y.

$$y = \sum_{i=1}^{n} \nu_i x_i \tag{4.5}$$

An operation with multiple independent inputs can be modeled by introducing an operating unit node for each input material A_i, which consumes the input material and produces the common output B (see Figure 14). Flow rates can be 1 for convenience from the inputs to the units, and ν_i from the units to the output.

It should be noted that capacities, investment and operating costs can be associated with each of the used operating unit nodes, resulting in more options for each input.

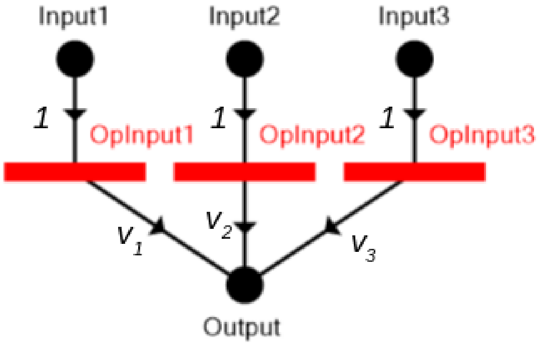


Figure 14: Modeling an operation with three independent inputs, using three operating unit nodes.

There are 2^n possible ways an operation with n independent inputs can be used in a solution structure, if the aforementioned model is used. In 1 of the cases, the operation is completely excluded, and in the $2^n - 1$ other cases, different subsets of the input materials are used, while the others are not. The latter can result in a large number of solution structures if n is large, while each can be a valid choice.

4.2.3 Output capacity with independent inputs

A possible extension of the case of independent inputs is the consideration of an upper bound for the output, which is essentially a total capacity limit. For example, the capacity of the furnace limits not only a single input material, but the total from all inputs.

The output capacity constraint with independent inputs requires Equation (4.5) for the independent inputs, and Inequality (4.6) for the new constraint, as shown below.

$$y \leq C_{MAX} \tag{4.6}$$

The capacity C_{MAX} can be a fixed constant, or can be linearly scaled with proportional investment costs. These two cases are directly supported by the current P-Graph Studio implementation. The problem is that this is possible for a single operating unit node, not for the total of many.

The key observation for providing a valid model for this case is that the constraint on the total outputs does not require the knowledge of the inputs. Therefore, a single operating unit node can be introduced after the original output, consuming it, then producing the new, final output material (see Figure 15). Any parameters related to the whole operation rather than a single input can be given for this single operating unit node, including capacity limit, but investment and operating costs can also be provided.

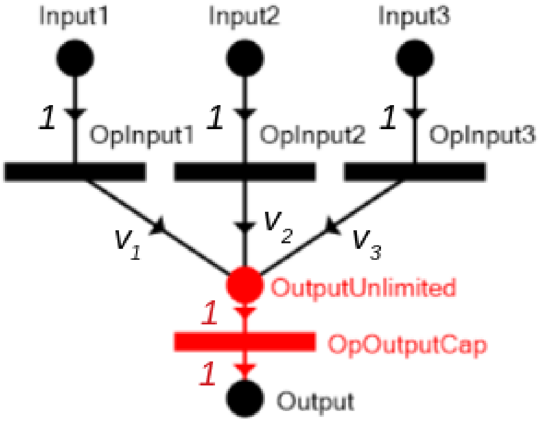


Figure 15: Addition of an output capacity constraint to an operation with independent inputs can be achieved by the introduction of a single operating unit node after the original output.

The solution structures in this case remain analogous to the case of independent inputs. The newly added operating unit is used in $2^n - 1$ cases where any input is used. Therefore, the number of solution structures remains the same after the implementation of the $y \leq C_{MAX}$ constraint.

4.2.4 Custom input capacity

The output capacity constraint can be viewed as an upper bound of a linear expression of the input material amounts, where the linear expression matches the output amount. The case where the linear expression can be arbitrary is more difficult.

For example, a pelletizing operation may have an output like the total heating power of the produced pellets, which is used by subsequent parts of a PNS model. However, the capacity of the pelletizer operating unit is rather expressed as the total mass or volume of materials involved, which are not necessarily proportional to heating power.

If a custom linear input capacity is imposed on an operation with independent inputs, then Equation (4.5) is needed for the independent inputs, and a new Inequality (4.7) is added to impose the constraint.

$$\sum_{i=1}^{n} \lambda_i x_i \leq C_{MAX} \tag{4.7}$$

Coefficients λ_i describe the custom linear capacity constraint, which are not necessarily equal to the ν_i coefficients determining the output amount. Note that $\lambda_i \geq 0$ is assumed, therefore each input is accounted against the upper bound C_{MAX} if $\lambda_i > 0$, or is indifferent if $\lambda_i = 0$.

The model for a custom linear input capacity is shown in Figure 16. A new logical operating unit is introduced with the sole purpose of producing a logical capacity material in C_{MAX} amount. This logical capacity material can be consumed by the operating units introduced for each A_i input material, with flow rates $\lambda_i > 0$. If $\lambda_i = 0$, no arc is introduced. As a result, if a composition x_i of input materials are consumed to produce an output amount $y = \sum \nu_i x_i$, the logical capacity material would also be required in $\sum \lambda_i x_i$ total amount.

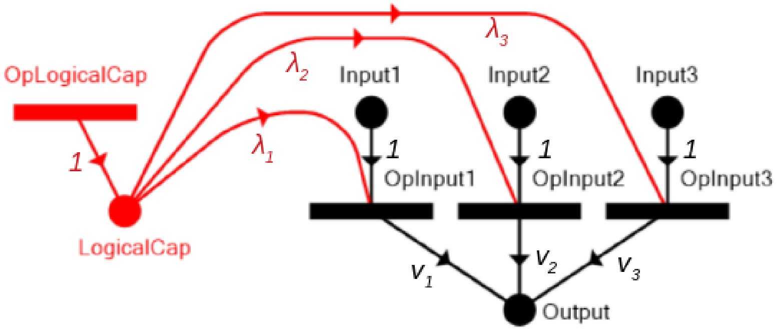


Figure 16: Modeling a custom linear input capacity for independent input materials, by the introduction of a logical capacity material and the logical unit producing it.

The logical capacity material and the unit producing it is included in a solution structure if and only if any input material A_i with $\lambda_i > 0$ is used for production. Therefore, the number of solution structures is not increased by this implementation of the constraint.

There is a notable alternative approach for modeling a custom linear input capacity. Consider the logical capacity node as a product of the individual operating units for each A_i instead, which can be drained by the logical operating unit node (see Figure 17). The current P-Graph Studio implementation allows a maximum net flow of input materials, which can be set to 0 for the logical capacity material, while the operating unit can have a capacity C_{MAX} for draining. The problem with this seemingly equivalent approach is that solution structures do not reflect the relevant situations. If algorithm MSG is applied, by Axiom (S4), the logical part is always excluded for not contributing to the production of a product node. Consequently, the role of inputs and outputs are not perfectly interchangeable in operating unit models. Another approach could be introducing a final product node for the logical part, but on the other hand, that would make the original flexible operation mandatory, because, by Axiom (S1), all product nodes must be included.

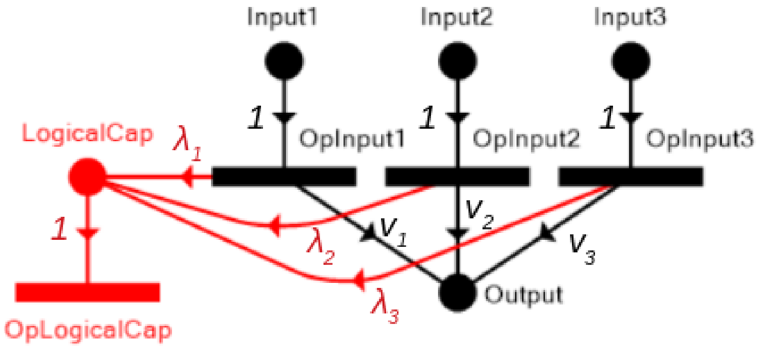


Figure 17: Wrong alternative approach for modeling a custom linear capacity.

4.2.5 Minimum input usage

There can also be a lower bound for the total input amounts instead of an upper bound. This can be the case, for example, when some regulation requires specific amounts of input materials to be consumed.

The same way as for upper bounds, Equation (4.5) is needed in this case, and a new Inequality (4.8) is added as a constraint.

$$C_{MIN} \leq \sum_{i=1}^{n} \lambda_i x_i \tag{4.8}$$

It is again assumed that $\lambda_i \geq 0$. To address this constraint a new logical product node can be introduced to the P-Graph (see Figure 18). The operating units associated with each input material A_i can produce this product node with λ_i flow rates. If $\lambda_i = 0$, no arc is needed. The product node should have a minimum flow of C_{MIN}, which completes the model.

If a product node must have a $C_{MIN} > 0$ inflow, a solution structure must include at least one input with $\lambda_i > 0$. This is a favourable behaviour, because this constraint does not only reduce the search space, but also the set of solution structures a solution procedure must take into account. According to this model, the flexible operation may not be omitted from any solution structure. If the constraint must only be imposed if the operation takes place, that is a different situation not covered here.

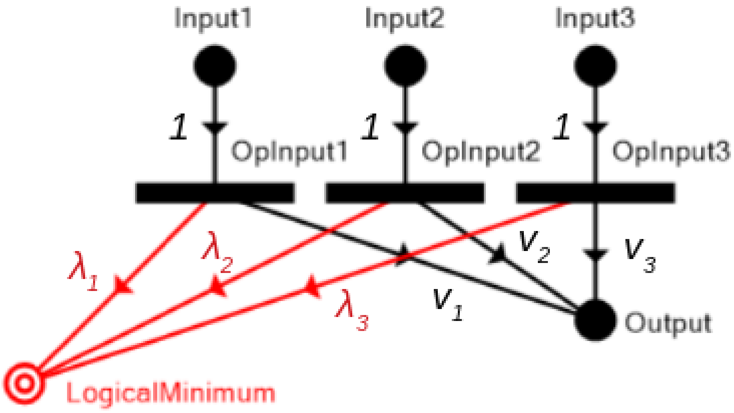


Figure 18: Modeling a minimum constraint on the total of inputs with the addition of a single logical product node.

4.2.6 Ratio constraints

Until this point, all constraints consisted of a nonnegative weighted sum of inputs bounded by a constant term. Consider the case without a constant term. Then a constraint only makes sense if input amounts times their $\lambda_i > 0$ coefficients appear on both sides of an inequality or equation. From now on, these constraints are called **ratio constrains**, which can be expressed in the general form shown in Inequality (4.9).

$$\sum_{i \in L} \lambda_i x_i \leq \sum_{i \in R} \lambda_i x_i, \quad L \cap R = \emptyset, \ \ \lambda_i > 0 \ \forall \ i \in L \cup R \tag{4.9}$$

Note that any nontrivial linear inequality for variables x_i without a constant term can be arranged into the form shown in Inequality (4.9). Equations as ratio constraints are not considered in this work, as an equation can possibly be mitigated by alternative P-Graph models, or expressed with two inequalities. A key property of ratio constraints is that x_i can be scaled, which means that multiplying x_i by the same positive constant maintains feasibility.

As an example, a furnace may take coal and wood as combustibles, but the total proportion of wood can be at most 20%, otherwise special maintenance would be needed. This can be expressed as a ratio constraint.

There can be multiple ratio constraints defined for the same operation.

4.2.7 Convex sums for ratio constraints

One possible approach to model a set of ratio constraints simultaneously is using **convex sums**. The set of possible input material compositions x_i for n inputs in $\mathbb{R}^n$ is a convex polytope, because all ratio constraints are linear. In certain situations, the interior of this polytope can be described by a convex sum of a finite set of its vertices. In a vector space, a convex sum of vectors $\mathbf{v}_i$ is of the form $\sum_i w_i \mathbf{v}_i$ where $w_i \geq 0$ are weights for which $\sum_i w_i = 1$. Since in ratio constraints x_i can be scaled, the condition $\sum_i w_i = 1$ is relaxed.

Two examples for convex sums are now shown below with corresponding P-Graph models.

Suppose that there are two input materials A_1 and A_2 and it is a requirement that none of the two can contribute by more than 80% to the total. This can be a practical consideration, as relying dominantly on a single resource can be unsafe. The input amounts are represented by x_1 and x_2. The required condition can be expressed as two ratio constraints (see Inequality system (4.10)).

$$\begin{aligned} x_1 &\leq 0.8\,(x_1 + x_2) \\ x_2 &\leq 0.8\,(x_1 + x_2) \end{aligned} \tag{4.10}$$

Note that these two constraints can be arranged into the form $x_1 \leq 4x_2$ and $x_2 \leq 4x_1$, respectively. Observe that $(x_1, x_2) = (1, 4)$ and $(x_1, x_2) = (4, 1)$ are the extreme cases for fulfilling these two inequalities. Therefore, without a precise proof, any (x_1, x_2) satisfies Inequality system (4.10) if and only if there exist weights $w_1, w_2 \geq 0$ such that Equation (4.11) holds.

$$(x_1, x_2) = w_1 \cdot (1, 4) + w_2 \cdot (4, 1) \tag{4.11}$$

This makes it possible to model the situation with just two operating unit nodes, for each extreme case (see Figure 19). The units consume A_1 and A_2 in the flow rates according to the corresponding extreme case, $(1, 4)$ or $(4, 1)$, and produce the final product with the flow rates $4\nu_1 + \nu_2$ and $\nu_1 + 4\nu_2$. The total output is $y = \nu_1 x_1 + \nu_2 x_2$ as desired.

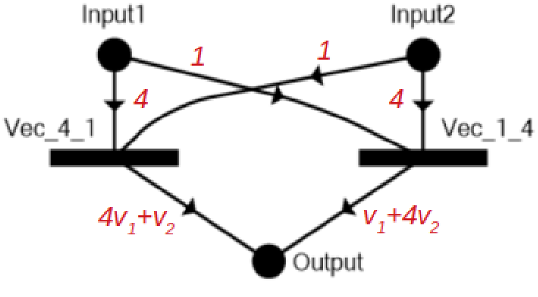


Figure 19: Modeling the first example: ratio constraints $x_1 \leq 4x_2$ and $x_2 \leq 4x_1$ with convex sums.

Solution structures for this model are now analyzed. If the operation is not used at all, none of the two units are chosen. If the operation is used, exactly one unit can be selected if the case is extreme, and both are needed if the composition is in between the extreme cases. In this example, there is not much redundancy in the model as there are only two input materials.

Now consider another example with three input materials A_1, A_2 and A_3, and a requirement that none of them can contribute by more than 70% to the total. The same design with convex sums can be applied. There are three ratio constraints, $x_i \leq 0.7\,(x_1 + x_2 + x_3)$, which could be arranged to different forms. There are six extreme cases for (x_1, x_2, x_3), which are $(7, 3, 0)$, $(7, 0, 3)$, $(3, 7, 0)$, $(0, 7, 3)$, $(3, 0, 7)$ and $(0, 3, 7)$. The P-Graph model now requires six operating units (see Figure 20).

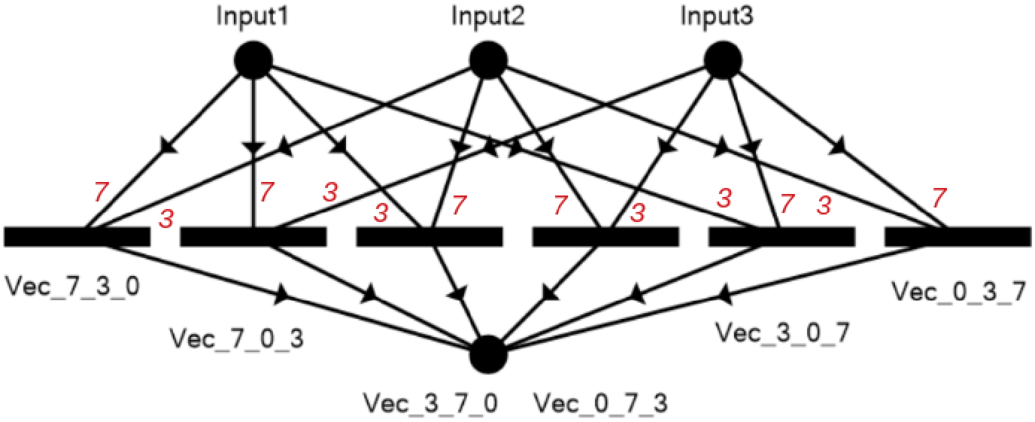


Figure 20: Modeling the second example: ratio constraints $x_1 \leq 0.7\,(x_1 + x_2 + x_3)$, $x_2 \leq 0.7\,(x_1 + x_2 + x_3)$ and $x_3 \leq 0.7\,(x_1 + x_2 + x_3)$ with convex sums.

In this case, the solution structure is again unique if the composition is an extreme case. If not, however, any 3 of the 6 extreme cases can be used to sum up to non-extreme ratios, which immediately results in $\binom{6}{3} = 20$ different, valid solution structures. There are more, even for a fixed (x_1, x_2, x_3), because expressing it as a convex sum of the extreme cases can be ambiguous.

Therefore, this model has a lot of redundancy, which is not good if the solution structures are to be enumerated. The technically same solution can even be represented in multiple solution structures. As a conclusion, using convex sums can be effective for very small cases, but become inefficient for even a moderate number of extreme cases.

4.2.8 Efficient ratio constraint implementation

Instead of relying on convex sums, a method is proposed to address one ratio constraint at a time. Recall Inequality (4.9) describing a ratio constraint.

$$\sum_{i \in L} \lambda_i x_i \leq \sum_{i \in R} \lambda_i x_i, \quad L \cap R = \emptyset, \;\; \lambda_i > 0 \;\forall\; i \in L \cup R$$

Consider the LSH and RHS of the inequality as two distinct logical materials connected as follows (see Figure 21).

- The LHS is consumed by the operating unit nodes introduced for A_i, in flow rates λ_i, for each $i \in L$.
- The RHS is produced by the operating unit nodes introduced for A_i, in flow rates λ_i, for each $i \in R$.
- The two logical material nodes are connected by a logical operating unit, consuming the RHS and producing the LHS in a $1:1$ ratio.

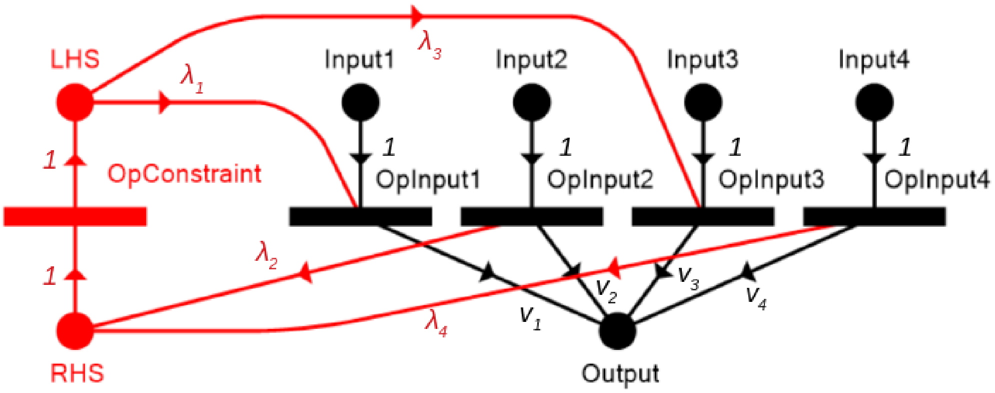


Figure 21: Modeling a ratio constraint with two logical material nodes and a logical operating unit node: example for $\lambda_1 x_1 + \lambda_3 x_3 \leq \lambda_2 x_2 + \lambda_4 x_4$.

Observe that the loop of material flow in the introduced logical part starts from the inputs involved in the RHS and ends in the materials involved in the LHS. The inflow of the RHS material node equals the actual RHS in the ratio constraint, and the outflow of the LHS material node equals the actual LHS in the ratio constraint. Because the LHS material can only be obtained by producing it from the RHS material, the constraint is satisfied. Note that there is no further, unnecessary restrictions caused by this design.

This design with a logical operating unit allows additional features to be implemented, for example minimum, maximum flows, fixed and proportional costs can be associated with the logical operating unit. If this is not necessary, the model can be further simplified by merging the logical parts into a single material node (see Figure 22). This is considered as the final design for ratio constraints.

If the two designs are investigated in terms of solution structures, it can be observed that if there is any input material A_i for which $i \in L$, then there must be at least one input material A_j for which $j \in R$. The reason for this is that when the logical material representing the LHS must be produced somehow, it is only possible by using a material from the RHS. Therefore, some infeasible solutions according to the ratio constraint are structurally excluded, which is again a good behaviour.

It should be noted that the introduction of logical nodes can negatively affect the readability of the graphical representation of the P-Graph model. One advantage of

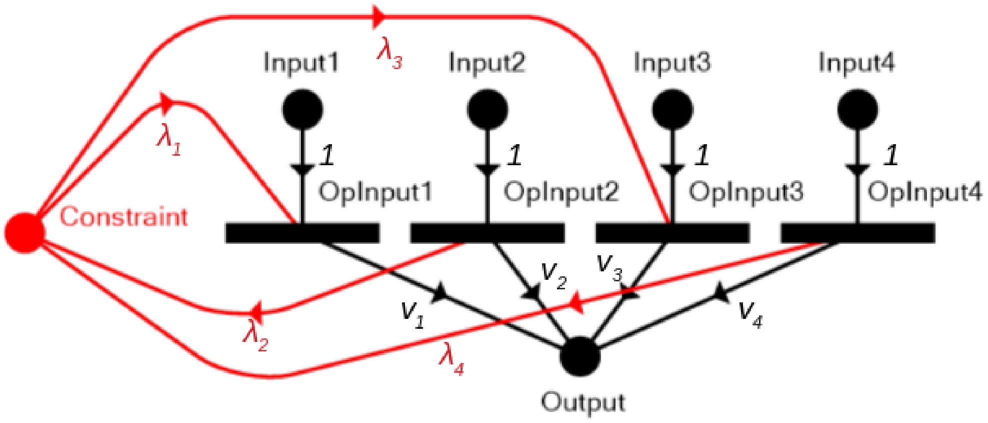


Figure 22: Simplified, final model of a ratio constraint with a single logical material node: example for $\lambda_1 x_1 + \lambda_3 x_3 \leq \lambda_2 x_2 + \lambda_4 x_4$.

the P-Graph framework over mathematical programming approaches is the easier readability, especially for decision makers not familiar with optimization tools.

4.2.9 General constraints

Using the aforementioned modeling techniques, the general constraint formulation described by Inequality (4.3) can be addressed at this point. A general P-Graph model can be constructed for a flexible operation, characterized by independent input amounts, restricted by constraints of the given form. It is assumed that the constraint is not trivially redundant or infeasible in the context of the operation. Recall the formulation in question as follows.

$$C_{MIN} + \sum_{i\in L} \lambda_i x_i \leq \sum_{i\in R} \lambda_i x_i + C_{MAX}$$
$$C_{MIN}, C_{MAX} \geq 0, \quad \min\{C_{MIN}, C_{MAX}\} = 0$$
$$\lambda_i > 0 \;\; \forall \;\; i \in L \cup R$$

The starting point of modeling is the n operating unit nodes introduced for each of the independent input materials A_i. First the constant terms are addressed as follows.

- If $C_{MIN} \neq 0$, a logical material node is introduced for the role of the RHS, which is a final product node in the P-Graph, with minimum required flow C_{MIN}.

- If $C_{MAX} \neq 0$, a logical material node is introduced for the role of the LHS, which is an intermediate material node in the P-Graph. Another, logical operating unit is introduced, which produces the material in C_{MAX} maximum amount.

- If $C_{MIN} = C_{MAX} = 0$, a single logical material node is introduced for the roles of both the LHS and the RHS.

The material amounts as terms of the constraint are addressed as follows.

- For any input material A_i with $i \in L$, the operating unit node introduced for A_i is connected to the logical material node for the LHS to consume it, with flow rate λ_i.
- For any input material A_i with $i \in R$, the operating unit node introduced for A_i is connected to the logical material node for the RHS to produce it, with flow rate λ_i.

Note that either the longer or the simplified modeling version can be used for ratio constraints. The simplified version means that the logical material nodes for the LHS and RHS can be merged, effectively resulting in a single logical material node for the constraint. Nevertheless, the logical operating units in this design, either for C_{MAX} or the longer version of the ratio constraint can be useful for associating further parameters of the process, for example limits on material flows, or investment and operating costs.

This method description was for flexible operations with a single output material B. If there are multiple output materials, these can simultaneously be included in the same model, where the operating unit nodes, common for each A_i are connected to each B_j with flow rates $\nu_{i,j}$.

4.3 Model examples

In this section, different instances of the aforementioned modeling techniques for flexible operations with P-Graphs are presented on an example model. In Chapter 3, the problem and its solutions are presented in detail, already utilizing some of the techniques. This case study is revisited, and extended with a ratio constraint to demonstrate how linear constraints can be imposed on the inputs of a flexible operation.

The problem was first published with a simple model [S2], and later was extended with an ad hoc application of a different flexible operation technique for the pelletizer and biogas plant operating units, and the multi-period modeling scheme [S3].

The energy consumption optimization of a manufacturing plant is to be optimized. Annual electricity and heating demands are given and must be satisfied with minimal operating and annualized investment costs. The business as usual solution is purchasing natural gas and electricity from the providers, but other energy sources are considered. One alternative is a solar power plant for electricity and possibly heat generation, with scalable size. The other, more important option is using seven, locally available types of biomass to produce biogas, and then feeding a biogas furnace or biogas CHP plant. Some types of biomass must be pelletized first, which is another production step in the chain. For further details of the problem, see Chapter 3 or the most recent publication [S3].

4.3.1 Flexible modeling approaches

The initial modeling technique with P-Graphs for this problem [S2] was the utilization of the technique which is termed in this chapter as **output capacity with independent inputs**. The key observation was that all types of biomass will eventually end up as heating power of the biogas produced from them. For this reason, all material nodes other than the raw material nodes were representing heating power of the materials involved. The biomass types that requires pelletizing were fed into the pelletizer, which produced one common output for pellets, which represented the total heating power of biomass involved. The other types of biomass, together with pellets were fed into the biogas plant to produce biogas, again expressed in terms of final heating power. See Figure 23 for an overview of this initial approach.

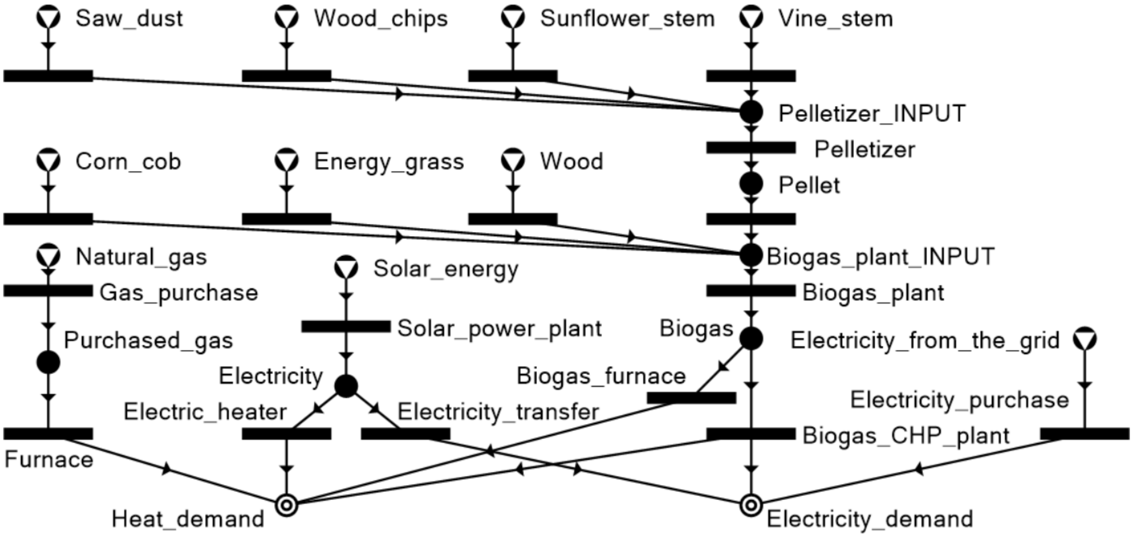


Figure 23: Initial approach for modeling the energy consumption optimization case study, utilizing the technique of output capacity with independent inputs.

The capacity of both the biogas plant and the pelletizer operating units were scalable in the model to fit annual production capacities. The more capacity an operating unit has, the more proportional investment and operating costs are accounted. The capacities of these units were expressed in terms of the total heating power of the products involved.

The issue with this initial technique is that the capacity of the pelletizer and biogas plant operating units should rather be expressed in terms of total mass of the biomass types involved. Note that although heating powers could have been quite different for the seven investigated biomass types, they were not in this case study. For this reason, the initial approach is also accurate to some degree.

The second, more precise P-Graph representation of the process is depicted in Figure 24. Note that only the single period model is presented here for the sake of simplicity. This is the technique termed **custom input capacity** for both the pelletizer and the biogas plant operating units. In this implementation, there are three different kinds of input materials for producing biogas:

- Ordinary raw material input nodes, representing the seven biomass types.
- Logical pelletizer capacity material node, which is produced by the logical pelletizer operating unit node in the amount determined by pelletizer size, and used up for each type of biomass requiring pelletizing, in amounts corresponding to their mass.
- Logical biogas plant capacity material node, which is produced by the logical biogas operating unit node in the amount determined by biogas plant size, and used up for any of the seven biomass types, in amounts corresponding to their mass.

The biogas is the output of all seven operating unit nodes introduced for each biomass type. Therefore, this model is not only an instance of the custom input capacity constraint for a flexible operation. This model incorporates two subsequent flexible operations with a custom input capacity constraint into a single one. The resulting flexible operation could be termed as biogas production in general, and has two distinct custom input capacity constraints: one for pelletizer capacity and one for biogas plant capacity. The investment and operating costs for these two technologies are associated with the logical operating unit nodes producing the capacity materials.

4.3.2 Extension with a ratio constraint

The problem is now extended by an additional constraint for demonstration purposes. For safety reasons, it is a reasonable requirement not to rely dominantly on a single resource, in case of unexpected changes in supply. One possible way of mitigating such risks is to maintain a backup plan for energy production – as done in the case study, by keeping the business as usual part of the structure, which is natural gas and electricity purchase.

Another mitigation technique is by adding constraints on input material composition. The optimal solution for the single period variant of the second model relied dominantly on energy grass, which made up to 72% of the total biomass used. Therefore, three distinct scenarios were considered and compared:

- Original model, as shown in Chapter 3, where the optimal solution structure used energy grass, and the mass of energy grass was 72% of the total for all seven biomass types.

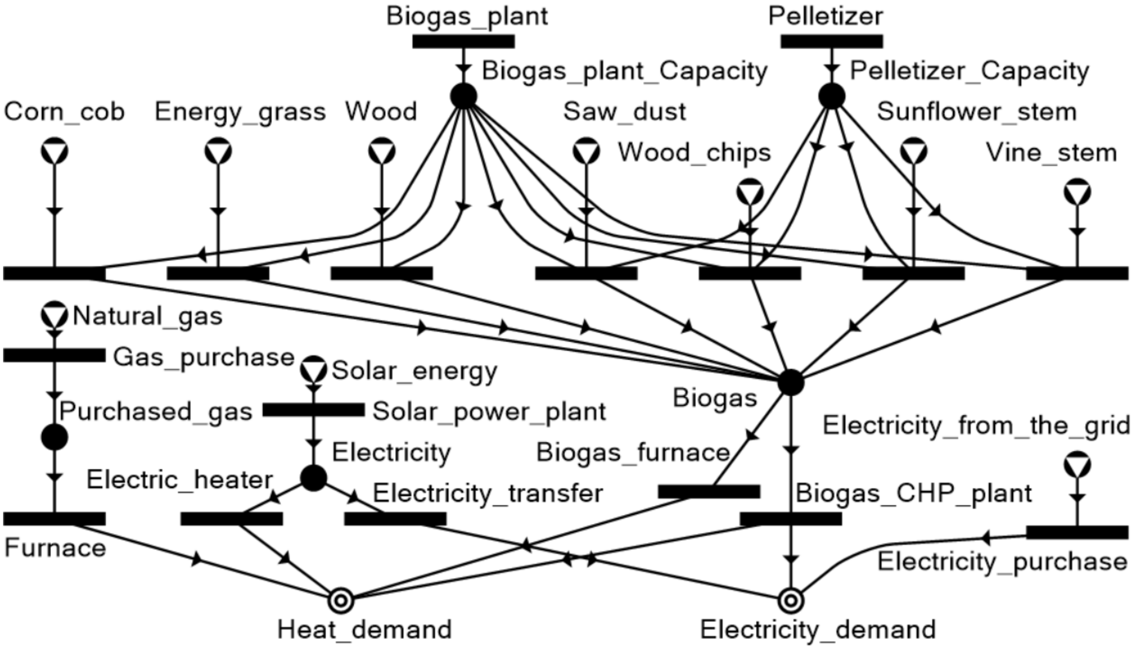


Figure 24: Improved model for the energy consumption optimization case study, utilizing the technique of custom input capacity.

- Extension by a constraint allowing a total of 70% energy grass.
- Extension by a constraint allowing a total of 50% energy grass.

The extensions are performed according to the **ratio constraints** modeling technique, without constant terms. First, the ratio constraint corresponding to the 70% limit is formulated as shown in Inequality (4.12).

$$x_{eg} \leq 0.7\left(x_{wc} + x_{sd} + x_{ss} + x_{vs} + x_{cc} + x_{eg} + x_w\right) \tag{4.12}$$

The variables represent input material amounts. Respectively, x_{wc} is for wood chips, x_{sd} is for saw dust, x_{ss} is for sunflower stems, x_{vs} is for vine stems, x_{cc} is for corn cobs, x_{eg} is for energy grass, and x_w is for wood.

The constraint can be arranged into the final form where each material appears at most one of the sides, and with a positive coefficient, as shown in Inequality (4.13).

$$3x_{eg} \leq 7x_{wc} + 7x_{sd} + 7x_{ss} + 7x_{vs} + 7x_{cc} + 7x_w \tag{4.13}$$

If the constraint is formulated for the 50% limit instead, the original form of the ratio constraint is the one shown in Inequality (4.14), and the final form is shown in Inequality (4.15).

$$x_{eg} \leq 0.5\left(x_{wc} + x_{sd} + x_{ss} + x_{vs} + x_{cc} + x_{eg} + x_w\right) \tag{4.14}$$

$$x_{eg} \leq x_{wc} + x_{sd} + x_{ss} + x_{vs} + x_{cc} + x_{w} \tag{4.15}$$

In both of the extension scenarios, the ratio constraint is implemented with the addition of a single logical material node, see Figure 25. The seven operating unit nodes responsible for the utilization of each biomass type are connected to the logical material node. The operating unit node utilizing energy grass consumes the logical material node. The other six operating unit nodes produce the logical material node. The two scenarios differ only in the flow rates, which can be directly read from the final forms of the ratio constraints as follows. Note that these flow rates could be simultaneously scaled by any positive constant factor, but the values shown are the most convenient choices.

- For the 70% constraint, the flow rate for energy grass is 3, and the flow rate for all other biomass types is 7.
- For the 50% constraint, the flow rates are all 1.

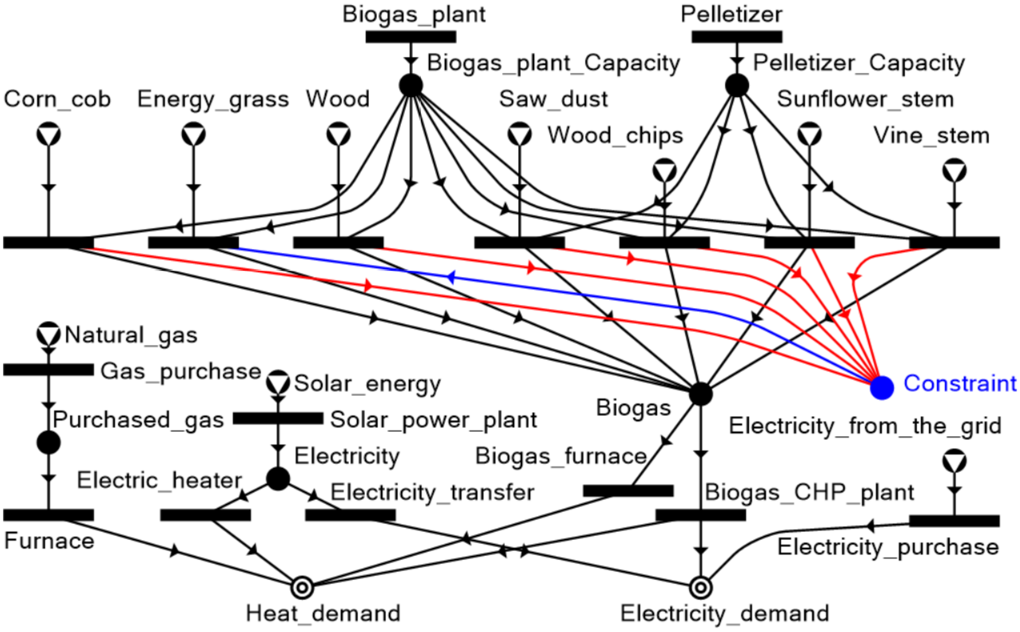


Figure 25: Modeling a ratio constraint for energy grass in the energy consumption optimization problem. Colors indicate direction of material flow.

All three scenarios were implemented as a PNS problem each, in P-Graph Studio v5.2, and solved by the ABB algorithm on a Lenovo Y50-70 computer with Intel i7-4710HQ CPU and 8 GB RAM. Also note that model sizes are not large, so the solution of the models was performed in less than a second in all scenarios. The optimal objectives for the 10 best solution structures are shown in Table 12.

Recall that the optimal solution for the original single period model version resulted in an objective of 220.709 M HUF/y. This solution structure involves corn

Table 12: Top 10 solution structures in the three scenarios, ranked for each scenario by total cost (shown in M HUF/y). Specialties of the solution structures: E=electricity purchase, S=solar power plant, N=natural gas purchase, P=pelletizer.

Solution structure	No limit		70% limit		50% limit	
1.	220.709	E	220.780	E	222.258	E
2.	224.057	E P	224.324	E P	227.928	S
3.	224.325	E P	224.890	E P	228.975	E P
4.	224.357	ENP	225.307	E P	229.391	E P
5.	224.496	E P	225.313	E P	229.404	E P
6.	224.526	E P	225.980	E P	230.529	ENP
7.	225.895	E P	226.451	S	231.749	P
8.	226.049	ENP	227.034	E P	232.308	P
9.	226.380	S	228.272	ENP	232.616	E P
10.	226.723	E P	228.284	ENP	232.667	P

cobs and energy grass, and a biogas CHP plant to provide heating and part of the electricity demand, and then purchasing the rest of the required electricity. This scheme is more or less true for the other solution structures as well. In some structures, there is natural gas purchase. The 9th solution structure is special, because a solar power plant substitutes electricity purchase.

Four solution structures for the 50% scenario are shown in Figure 26. It can be concluded that the constraints work as desired, both for the logical pelletizer and biogas plant capacity, and for the ratio constraint on energy grass composition. The objective does not change much for the 10 first solutions throughout the three scenarios, mainly because the energy grass can be substituted by the other biomass types if needed. A slight and a bit larger increase can be observed as the constraint is introduced for 70% and then decreased to 50%. This is natural, because the search space shrinks as the limit is set lower.

The top solution structure is the same in all three scenarios. It should be noted that this is the only structure in all three cases where neither pelletizing is needed nor solar power is utilized.

In spite of these observations, the other solution structures do differ in the three scenarios. Most of the variety of the structures result in selecting the set of biomass types to work with. The ranking of these solution structures can be rearranged if the availability of energy grass changes, since energy grass is dominant in all cases.

There are two major findings which worth attention.

- The special solution structure which does not require electricity purchase from the grid, but utilizes a solar power plant is 9th in the original scenario, but goes 7th in the 70% scenario, and 2nd in the 50% scenario. It seems that solar power easily turns out to be favorable if a slight decrease occurs in biomass availability. Note that this observation is specific to this case study.

- The 7th, 8th and 10th solution structures in the 50% scenario show a new possibility because no energy purchase is needed in these cases, and not even the solar power plant is utilized. These structures were not in the top 10 without the 50% limit for energy grass, for being inefficient.

In general, a new constraint may vary the relative order of existing solution structures, some may become infeasible.

The ratio constraint also has the advantage of excluding infeasible solution structures, in particular, where only energy grass is used but none of the other biomass types. Note that these structures were simply not in the top 10 for the original scenario without the ratio constraint for being relatively inefficient.

In this particular case study for energy consumption optimization, it turned out that the changes in terms of the objective are not significant due to the ratio constraint. However, the order of solution structures can be rearranged so that new solutions become favorable, even if the change in the objective is minor.

The models presented demonstrate multiple techniques for modeling operations with flexible inputs. Limitations on pelletizer and biogas plant capacity and the ratio constraint for energy grass impose the desired restrictions on the search space, and allow the seven types of biomass to vary, but under prescribed circumstances. These techniques can be useful in the future if a process requiring different inputs in arbitrary composition are to be modeled with the P-Graph framework. If such a model is designed, it can be directly solved by already existing tools, for example, using the P-Graph Studio software and the ABB algorithm.

4.4 Application case study

To demonstrate the advantage of flexible input models instead of fixed inputs, and the P-Graph technique for doing so, another case study was performed. Due to space limitations, only the important details are presented here. A more detailed description can be found in Appendix A.

4.4.1 Problem description

The goal was the sustainable and economical utilization of locally available biomass for a small rural region near the town of Bad Zell. The problem was originally proposed and solved by Niemetz et al. [38] using a PNS formulation. Key model components, also shown in Figure 27, are the following:

- Different types of biomass are available from local agriculture: manure, intercrops, grass and corn silage.

- Biomass can be transported to 3 possible processing locations.

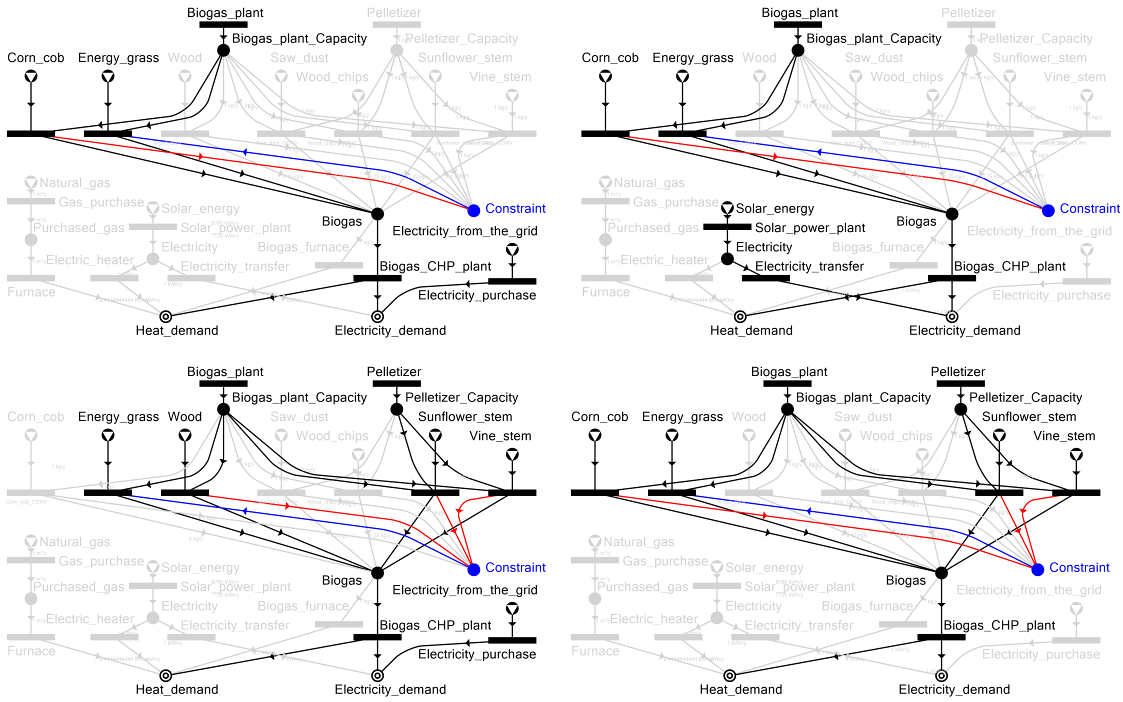


Figure 26: Solution structures for the 50% constraint, ranked by optimal objective value: 1 (top left), 2 (top right), 3 (bottom left), 7 (bottom right). Unused parts of the maximal structure previously shown in Figure 25 are grayed out.

- At each processing location, multiple fermenters can be built, in sizes 80 kW, 160 kW, 250 kW and 500 kWh, which produce biogas.
- At each processing location and the central town location, CHP plants can be built in the same sizes as fermenters, producing heating and electricity, which is sold.
- Supporting infrastructure must be built: biogas pipes, heat pipes, transformers and silo plates (not depicted).

In total, 8 fermenter designs were used in the original model. Each design used a fixed composition of the four available biomass types. The goal was to substitute these fixed input models with fermenters having flexible inputs and observe how the model and the optimal solution changes.

4.4.2 MILP model formulations

Although the case study was originally formulated as a PNS problem, the model formulations were not available, only the problem data and the optimal solution.

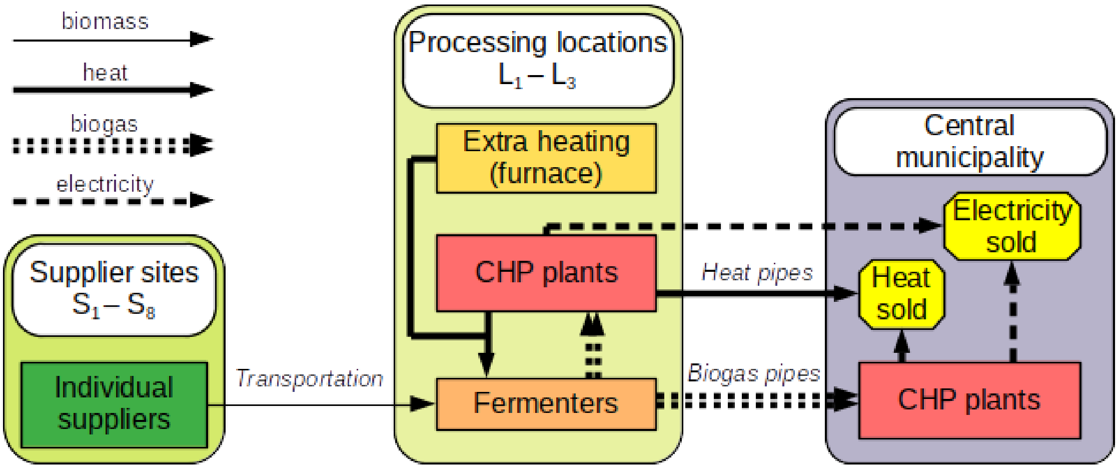


Figure 27: Main components and material flows in the Bad Zell case study.

Therefore, the first phase of this work was the reproduction of the originally published results. This was performed using MILP models first, to serve as a basis for comparison with the PNS implementation later.

The reproduction with a MILP model provided the following optimal solution for the problem, which was used as a basis for further models. One 160 kW and one 250 kW CHP plant at the central location, and one 80 kW CHP plant at processing location L_1 are built. These are supplied by two 250 kW fermenters at location L_1, with input ratios of $50:20:10:20$ and $75:15:10:0$ for the four biomass types, respectively. These were two of the eight possible input ratios for fresh matter input. According to this solution, 100% of manure, 75% of intercrops, 84% of grass and 74% of corn silage was used from the total available amounts, resulting in a 234,544 EUR/y profit.

The second phase of the work was expressing two key parameters: heating requirement and investment cost, as a linear function of biomass inputs, which was necessary for using flexible inputs. Note that other fermenter parameters were either irrelevant or independent of inputs. A multiple linear regression was made based on the available fermenter designs. The differences of the obtained linear estimation data and original parameters for the fixed input fermenter designs were between -7.8% and $+6.5\%$. The optimal solution with the estimated data used the same key decisions and slightly different material and monetary amounts, the objective was 233,033 EUR/y. Therefore, the linear estimations were considered accurate.

The third phase of the work was using the same estimated data, but with a flexible model instead of the fixed input model. Therefore, another MILP model was developed, now with flexible input design for fermenters. The two MILP models, representing the fixed and the flexible model could be directly compared on the same problem data.

The MILP model with the flexible fermenters resulted in a significantly different solution. Two 250 kW CHP plants at the central location and one 80 kW CHP plant at processing location L_1 is built. One 80 kW fermenter using only manure, and one 500 kW fermenter using biomass types in roughly 39 : 31 : 17 : 13 fresh matter input composition are built at L_1. This solution results in a higher heat and electricity throughput, better utilization of biomass (90% for corn silage, 100% for all other types), and a 306,711 EUR/y profit, which is 31% higher than the original. The resulting MILP model was also smaller, requiring only 0.5 s instead of 3.9 s to be solved with the GLPSOL MILP solver.

Concluding these results, it seems definitely better to use a flexible input model for optimization, than using a set of fixed input compositions to choose from. Note that the following assumptions must be made to use flexible inputs this way. First, the data used in the model are linear estimations that should be accurate enough. If a solution is reported for the flexible model, it should be verified whether the input composition is feasible in reality, that is, whether a fermenter unit can actually be built to work with the input amounts prescribed by the solution.

4.4.3 P-Graph implementation

The case study was also formulated as a PNS problem. The flexible input modeling technique presented in this chapter was required for the fermenter units, which is now shown in detail, see Figure 28.

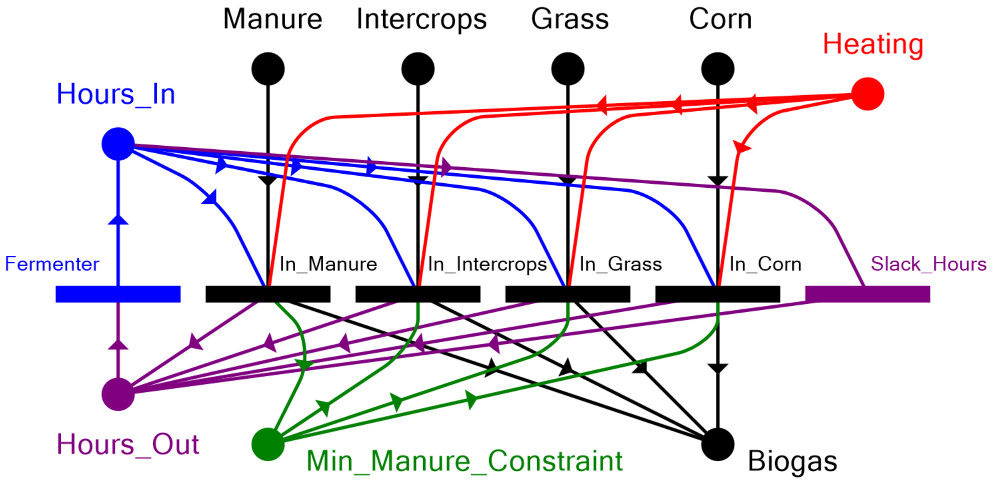


Figure 28: P-Graph model of the fermenter with flexible inputs.

The black part models the independent inputs of the fermenter, which are the four biomass types. Each input has its own operating unit node, which also play a key role in other components of the design. Flow rates for inputs are all 1, and flow rates for the output *Biogas* can be set according to the energy content of each biomass type.

The red part models fermenter heating. Since heating requirement is expressed as a linear function of biomass inputs (without a constant term), flow rates can be set according to each biomass type.

The blue part models the capacity of the fermenter. A single $Fermenter$ operating unit produces the $Hours\_In$ material, in a maximum amount of 7800 hours, which represents total working hours during the year. Flow rates can be set according to the rule that a fully operating fermenter can exactly feed a CHP plant with the same size, regardless of input type. Note that, counter-intuitively, working hours are not really split among input types, but rather a mix of inputs is fed to the fermenter in the composition described by the consumption rate of $Hours\_In$. This is only a modeling trick for implementing an upper bound for a weighted sum. Fixed costs of the fermenter are associated with the $Fermenter$ node.

The purple part was necessary for the investment costs. Since investment costs are expressed as a linear function of biomass inputs (without a constant term), it can be associated to the operating unit of each biomass type as a proportional cost. The problem is that it is allowed (although discouraged) to use a fermenter below full capacity, although the investment cost should not be scaled down in this case. To preserve the linearity of both the MILP and the P-Graph approach, the following estimation was made. Unused capacity contributes to the investment cost as if it was spent for the most expensive biomass type regarding investment costs. This biomass type is manure, for any fermenter size. The $Slack\_Hours$ operating unit is introduced for consuming the remaining fermenter capacity, and has a proportional cost equal to that of the operating unit for manure. All of $Hours\_In$ are reproduced into $Hours\_Out$, which is a new input for the $Fermenter$ operating unit. Also, the $Fermenter$ node has a minimum production also set to 7800 hours. These ensure that investment costs are always calculated for 7800 working hours.

The green part implements the constraint that at least 30% of input fresh matter must be manure. The introduced material node is connected to the operating unit of each biomass type. The operating unit of manure produces the material with a flow rate of 7, all other operating units consume it with a flow rate of 3.

The final P-Graph consists of 147 material nodes, 319 operating unit nodes, and 1144 arcs. These required 24 distinct fermenters (at three locations, in four possible sizes, two allowed per size). Due to the model size, the problem was not implemented using the GUI of P-Graph Studio, but solved by its underlying P-Graph solver (v2.0.3) after generating the model programmatically.

The solver with ABB algorithm required 413.45 s to finish. The significantly larger solution time can be attributed to the fact that in the MILP model, CHP plants of the same kind are represented as integer variables instead of multiple binary variables to decrease redundancy, while in the P-Graph implementation, each CHP plant has its own operating unit node. The ABB algorithm itself could treat redundancy better. Note that the MILP model which is also technically equivalent to the PNS problem could be generated by the P-Graph solver, and solved by the CBC MILP solver in 19.99 s, but the GLPSOL solver could not finish it in 1,000 s.

Nevertheless, the exact same solution was obtained for the PNS problem as for the MILP models. This holds for both the original MILP model implementation solved by GLPSOL to optimality, and the generated version technically equivalent to the PNS problem, solved to optimality by the CBC MILP solver.

This indicates that the P-Graph framework and its existing tools provide an alternative to mathematical programming models, and can be extended by modeling techniques.

4.5 Thesis summary

Thesis 2. A new modeling technique for the P-Graph framework was presented, which allows operations with flexible inputs and arbitrary linear constraints on input composition to be modeled purely as a PNS problem and solved directly by existing algorithms like ABB for P-Graph models.

Related publications: [S4], [S6].

T2.1. Nine scenarios were considered, representing different modeling goals for input composition and constraints. P-Graph models were provided, and possible solution structures were investigated. The final scenario allows independent inputs with arbitrary linear constraints on the input amounts. The importance of flexible inputs and the applicability of the proposed method were demonstrated on two case studies involving pelletizer and biogas plant models in a manufacturing plant, and fermenter models in a rural biomass supply chain.

Chapter 5

MILP model for mobile workforce management

5.1 Overview

In this chapter, a solution method for mobile workforce management problems is presented. Mobile workforce management focuses on the transportation of working individuals or teams to various places. There are tasks to be performed at these places, which can be delivery, on-demand or maintenance services, and may have constraints in terms of exact timing, resources needed, or dependencies on each other. In essence, both scheduling and vehicle routing is involved. Mobile workforce management aims to optimize short-term operation, and the emphasis is on precise sequencing and timing. This is in contrast to supply chain synthesis, where usually a long-term, static solution is to be found, and material balance is the key which poses the critical constraints. A common trait is that transportation may play a significant role, and is generally wanted to be minimal, obviously because it consumes resources and time.

It can be seen from the literature (see Section 2.2) that mobile workforce management problems are diverse in terms of the exact problem formulation. The resulting optimization problem is usually difficult, for which various approaches had been proposed. The goal in this work was to address the widest range of problem features arising in mobile workforce management in a unified approach. Therefore, an MILP model was formulated, and is presented in detail. A novelty of this model is that it is a slot-based approach in terms of decision variables, in contrast to literature models for which the precedence-based approach is generally used.

For larger problem instances, solving the proposed MILP model to optimality as a standalone method would require too much time. For this reason, an algorithmic solution framework was also designed to address larger problem instances. This framework provides solutions based on a heuristic in acceptable time. Several series of test problems were solved to demonstrate the applicability of the proposed approach and to investigate the effect of certain problem and model parameters.

First, the basics of the proposed algorithmic framework was published [S1]. A subsequent publication includes the final version of the algorithm, the full MILP model formulation, and various test results [S5].

Section 5.2 details the problem specification and shows an example problem and solution. The MILP model is presented in Section 5.3, the algorithmic framework is presented in Section 5.4, and computational results are shown in Section 5.5.

5.2 Problem specification

The mobile workforce management problem to be solved is now described. Each component of the targeted real-world problem has its own assumptions, listed later. The problem aims to execute a set of tasks given at various sites. An illustrative real-world example for this scenario is a public service company which executes maintenance and repair jobs on-demand, taking place at different points of the infrastructure (e.g. the power grid).

The company has teams to which the tasks are assigned. Each team starts at its depot at the beginning of the workday, then executes tasks one after the other, and finally returns before the end of the workday. The exact timetables of all working teams are to be found, subject to several constraints. An illustration of a potential solution is shown in Figure 29.

5.2.1 Modeling assumptions

Objective and scope of optimization

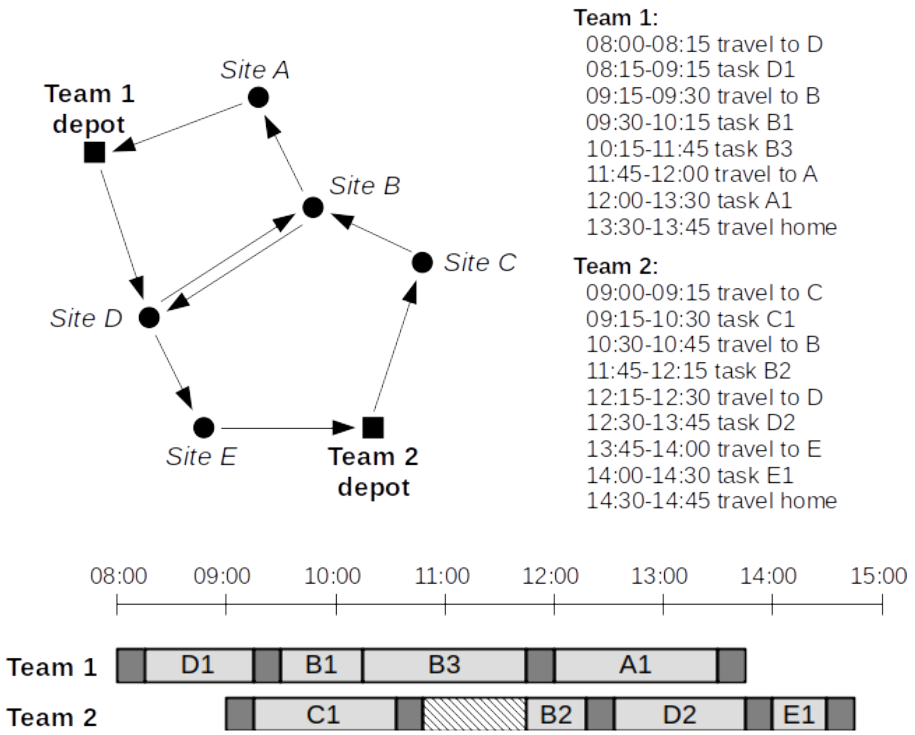
The objective is cost minimization for a workday subject to all tasks being executed. This excludes task selection from decision making. The rationale behind this is that distinction of critical and non-urgent tasks for a workday could be made at a higher level, but this is out of our scope. Nevertheless, cost parameters could be used to manipulate task selection and execution.

Task scheduling

Each task must be assigned to exactly one team. Assignment is one of the key decisions. The execution of tasks is subject to the following assumptions.

- Teams are fixed (forming and altering teams is out of scope).
- Execution is non-preemptive (task execution cannot be split in any way).
- Execution times and costs depend on the task and the team chosen.

Tasks are executed at different **task sites**. A team must travel from the depot to its first task site, between different task sites, and from its last task site back to the depot. The following assumptions are made about travelling.



Team 1:
08:00-08:15 travel to D
08:15-09:15 task D1
09:15-09:30 travel to B
09:30-10:15 task B1
10:15-11:45 task B3
11:45-12:00 travel to A
12:00-13:30 task A1
13:30-13:45 travel home

Team 2:
09:00-09:15 travel to C
09:15-10:30 task C1
10:30-10:45 travel to B
11:45-12:15 task B2
12:15-12:30 travel to D
12:30-13:45 task D2
13:45-14:00 travel to E
14:00-14:30 task E1
14:30-14:45 travel home

Figure 29: Illustration of a solution of the specified mobile workforce management optimization problem: given a set of tasks, find the timetable of teams for a workday. Note that this illustration is unrelated to the motivational example shown later.

- Each task is located at a single task site.
- Multiple (generally unrelated) tasks may be located at the same site.
- The distance of two sites is a fixed parameter, to which travelling times and costs are proportional. Each team has its own moving speed and cost ratio as parameters.
- Teams are allowed to be idle besides or instead of actually travelling.
- The total time travelled, total distance travelled, and total time spent by a single team in duty each can have an upper bound.

Before and after executing a task, other activities are mandatorily or optionally performed by a team, as shown in Figure 30, these are detailed later.

Packing and unpacking times

The problem formulation allows so-called **packing and unpacking** activities, representing preparations and post-work when a team moves between sites, subject to the following rules.

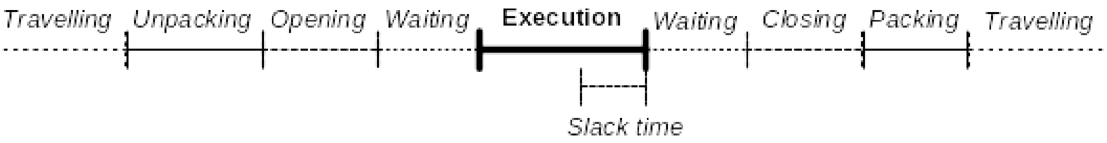


Figure 30: Activities a team may perform before and after a task is executed.

- If a team arrives at a site, it performs an unpacking activity.
- If its previous task was on the same site, unpacking is not performed.
- If a team leaves a task site, it performs a packing activity.
- If its next task is at the same site, packing is not performed.
- Packing and unpacking have fixed costs and times for each team.

Absolute and expected time windows

Tasks are mandatory, but the exact time of their execution may be subject to further restrictions and affect costs. For this reason, so-called **absolute and expected time windows** are introduced as follows (also see Figure 31).

- Each task has its own absolute and expected time window, which are time intervals during the workday.
- The absolute time window of a task *must* contain the beginning and the end of execution.
- The expected time window of a task *should* contain the beginning and the end of execution. Earlier beginning or later finishing incurs additional costs, proportional to earliness or lateness. Cost factors are specific to the task.

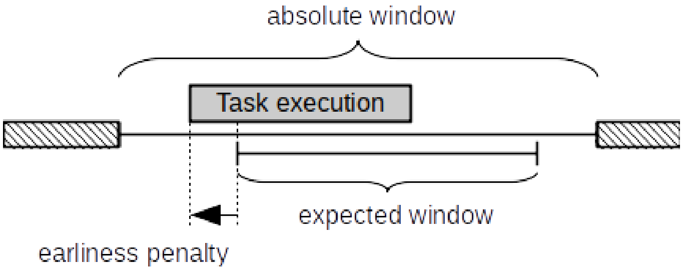


Figure 31: Absolute and expected time windows.

Consumable and tool resources

The problem formulation allows resources, which are required by task execution. The rules governing resource usage are the following.

- Each task may have a requirement for each resource, the amount depending on the task and the executing team.
- Each team has a carrying capacity for each resource, which cannot be refilled during the workday.
- Each resource has a maximum available amount to be distributed among teams, and a proportional usage cost.

Two kinds of resources are allowed: **consumables and tools**. Consumables are used up in tasks, but tools persist. For that reason, the total amount of a resource needed for a set of tasks is the sum of the individual requirements for consumables, and the maximum of the individual requirements for tools (see Figure 32).

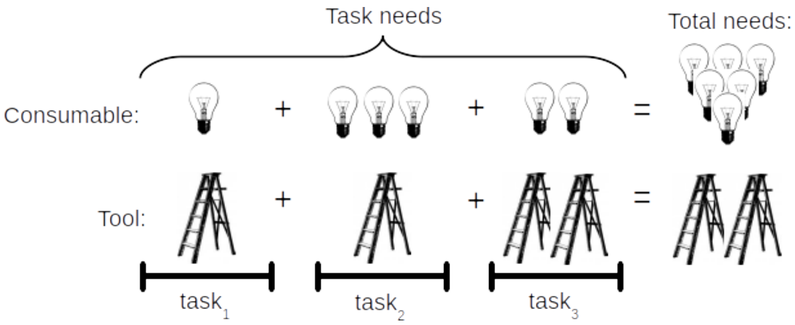


Figure 32: Consumable and tool resource needs of a single team.

Relationships between tasks

The following relationships are part of the problem specification, each introducing an additional constraint for the execution of two tasks k_1 and k_2 (see Figure 33).

- **Free precedence**: k_2 must start after k_1 is finished.
- **Same-team precedence**: k_2 must be done by the same team after k_1.
- **Protected precedence**: free precedence plus additional security measures. k_1 and k_2 may be done by two different teams, but leaving the site unattended is hazardous (for example in case of roadworks, electric boxes), so two options are available. The team executing k_1 can wait for the team executing k_2 to arrive, or it can perform a closing activity on the site, and then the team executing k_2 will first perform an opening activity. Both opening and closing activities require a given time and cost.

- **Mutual exclusion**: execution of k_1 and k_2 must not overlap.
- **Parallel execution**: execution of k_1 and k_2 should be started and finished simultaneously by two different teams. Slack time is introduced for the faster team to wait for the slower one to finish.

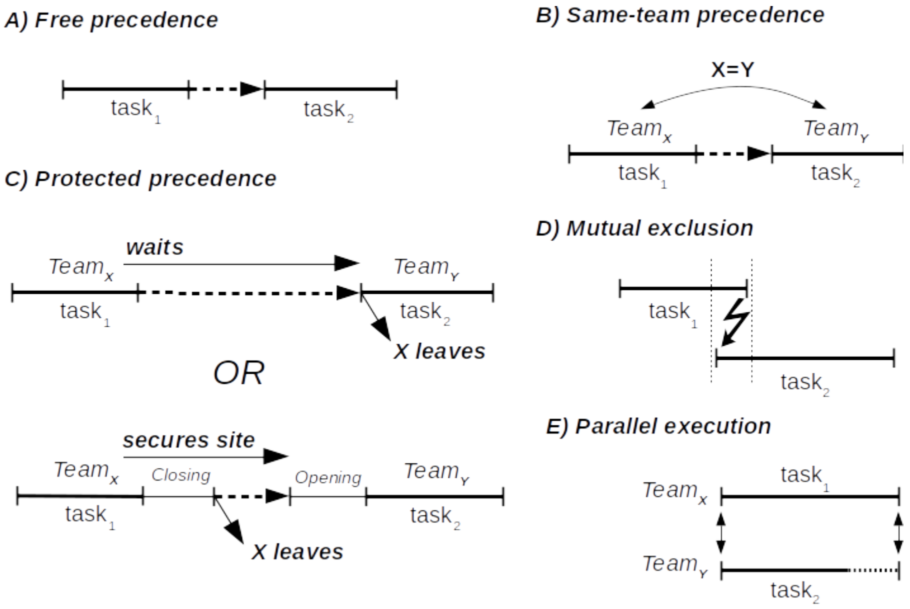


Figure 33: Possible pairwise relationships between tasks.

5.2.2 Motivational example

A motivational problem is presented here to demonstrate all components of the problem specification. A fictional company is responsible for the infrastructure of public lighting.

There are 8 mandatory maintenance tasks reported by the beginning of the workday: K_{1a}, K_{1b}, K_{1c1}, K_{1c2}, K_{1d}, K_2, K_{3a}, K_{3b}. These are located at three task sites S_1, S_2 and S_3, the index denoting the site of each task. Two working teams, $Team_1$ and $Team_2$ start from a depot site D at 8:00, execute the tasks, and return by 16:00. The sites are shown in Figure 34. Manhattan distances are used, which are calculated as follows.

$$D_{P_1,P_2} = |x_1 - x_2| + |y_1 - y_2| \quad \forall\ P_1(x_1, y_1),\ P_2(x_2, y_2) \tag{5.1}$$

Both $Team_1$ and $Team_2$ work for 60 EUR/h. $Team_1$ is lightweight, they move with 75 km/h, for 0.4 EUR/km, and do each task in 45 min for 100 EUR. $Team_2$ operates with machinery and consequently works faster but their other traits are worse: they move with 50 km/h, for 0.9 EUR/km, and do each task in 30 min for

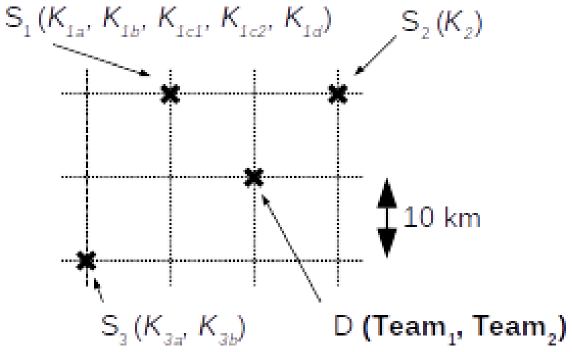


Figure 34: Position of task sites, with tasks in parentheses and depot with teams.

150 EUR. Note that the problem specification would allow limits on total working and travelling times, and also distinct execution times and costs for each pair of task and team, but these are omitted here.

Packing and unpacking activities represent the teams preparing the site for work and cleaning up afterwards. Both activities take 10 min and cost 10 EUR.

Absolute and expected time windows are allowed by the problem specification for each task separately. In this example, only K_{1b} has an absolute time window from 10:00 to 13:00, and an expected time window from 11:30 to 12:30, and violating the expected time window in either the direction of earliness or lateness costs 600 EUR/h. The reason for such constraints could be a client or co-operator only available in the given intervals.

Resource utilization is also included. One consumable resource type and one tool resource type is present, with usage costs of 15 EUR/unit and 100 EUR/unit. Each task requires one unit of each, and each team can carry 5 of the consumable and 1 of the tool resource.

There are also relationships between tasks, depicted in Figure 35, detailed below.

- Tasks at site S_1 describe a complex procedure. K_{1c1} and K_{1c2} must be executed in parallel. Otherwise, the order is K_{1a}, K_{1b}, K_{1c1} and K_{1c2}, and finally K_{1d}, and the site cannot be left unsecured. Therefore, the following protected precedence pairs are included in the problem: (K_{1a}, K_{1b}), (K_{1b}, K_{1c1}), (K_{1b}, K_{1c2}), (K_{1c1}, K_{1d}), (K_{1c2}, K_{1d}). Closing and opening a site to satisfy a protected precedence relationship cost 30 EUR and take 15 min per occasion.

- Tasks at site S_3 must be done in order K_{3a}, K_{3b}, and by the same team because information is required from K_{3a} to complete K_{3b}. This is formulated as a single task pair, as a same-team precedence relationship (K_{3a}, K_{3b}).

- Execution of K_{1a} interferes with critical parts of the grid, so while it is executed, neither K_2 and K_{3a} can be under execution. This is expressed as two mutual exclusion relationships, as (K_{1a}, K_2) and (K_{1a}, K_{3a}).

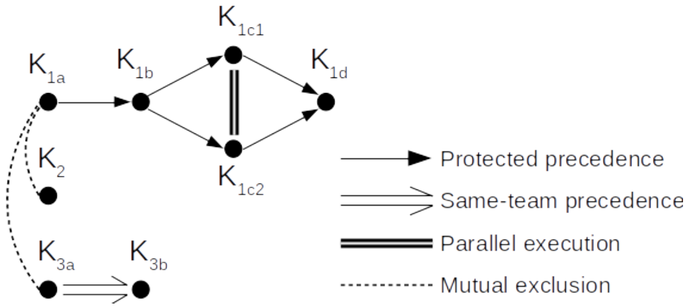


Figure 35: Relationships between tasks of the motivational problem.

Solely with the MILP model to be described in Section 5.3, this motivational problem could be solved to optimality, almost instantly. The results are shown in Table 13. The objective is 1,944 EUR. Observe that all task relationship constraints are satisfied, and K_{1b} was scheduled in its expected time window, so there was no penalty. All protected precedence relationships were satisfied without closing and opening activities. The more expensive $Team_2$ seems to be only used to reduce load on $Team_1$. For the parallel tasks, both teams are required, and also note that the execution requires 45 min for both teams, since they are synchronized and $Team_1$ is slower.

5.3 MILP model formulation

An MILP model was developed to address the specified mobile workforce management problem. The model was implemented in the GNU MathProg modeling language. A problem instance can be implemented and solved by a single call of an ordinary MILP solver. This is called the **standalone MILP** solution method. The MILP model is now presented. The used symbols are listed in the Nomenclature in Appendix B.

5.3.1 Decision variables

A purely MILP modeling approach like the standalone MILP method can be ineffective for large instances. The focus was on the easier adaptability by the algorithmic framework later presented in Section 5.4, instead of fast solutions by ordinary MILP solvers.

The main novelty of the proposed approach is that a **slot-based** MILP model was developed, in contrast to most literature approaches for VRP and mobile workforce management where MILP models are usually precedence-based. The following slot concepts are introduced, illustrated in Figure 36.

- A **job slot** is a placeholder for task assignment. Each team m has a predefined sequence of its own job slots. A job slot may have a single task assigned, or it can be unused which means no assignment.

Table 13: Optimal timetables of the teams in the motivational problem.

Site	From	To	Action of $Team_1$
D	08:02	08:12	Packing.
D	08:12	08:36	Move from D to S_3 (30 km).
S_3	08:36	08:46	Unpacking.
S_3	08:46	09:31	Execute task K_{3a}.
S_3	09:31	10:16	Execute task K_{3b}.
S_3	10:16	10:26	Packing.
S_3	10:26	10:50	Move from S_3 to S_1 (30 km).
S_1	10:50	11:00	Unpacking.
S_1	11:00	11:45	Execute task K_{1c2} (parallel with K_{1c1}).
S_1	11:45	12:30	Execute task K_{1d}.
S_1	12:30	12:40	Packing.
S_1	12:40	12:56	Move from S_1 to S_2 (20 km).
S_2	12:56	13:06	Unpacking.
S_2	13:06	13:51	Execute task K_2.
S_2	13:51	14:01	Packing.
S_2	14:01	14:17	Moving from S_2 to D (20 km).
D	14:17	14:27	Unpacking. (End of workday.)

Site	From	To	Action of $Team_2$
D	09:16	09:26	Packing.
D	09:26	09:50	Move from D to S_1 (20 km).
S_1	09:50	10:00	Unpacking.
S_1	10:00	10:30	Execute task K_{1a}.
S_1	10:30	11:00	Execute task K_{1b}.
S_1	11:00	11:30	Execute task K_{1c1} (parallel with K_{1c2}).
S_1	11:30	11:45	Prolong execution by 15 minutes.
S_1	11:45	11:55	Packing.
S_1	11:55	12:19	Move from S_1 to D (20 km).
D	12:19	12:29	Unpacking. (End of workday.)

- A set of **travelling slots** is introduced for each team. There is a single travelling slot before the first, after the last, and between two consecutive job slots of the team. Travelling slots are used to represent the movement of a team between sites.

- The concept of **site slots** is also used, which mean points in the schedule of a team when its position is in question. Precisely, the beginning and the end of the workday, and each job slot is a site slot. The purpose of site slots is convenient modeling, as with these definitions, each job slot is surrounded by two travelling slots, and each travelling slot is surrounded by two site slots.

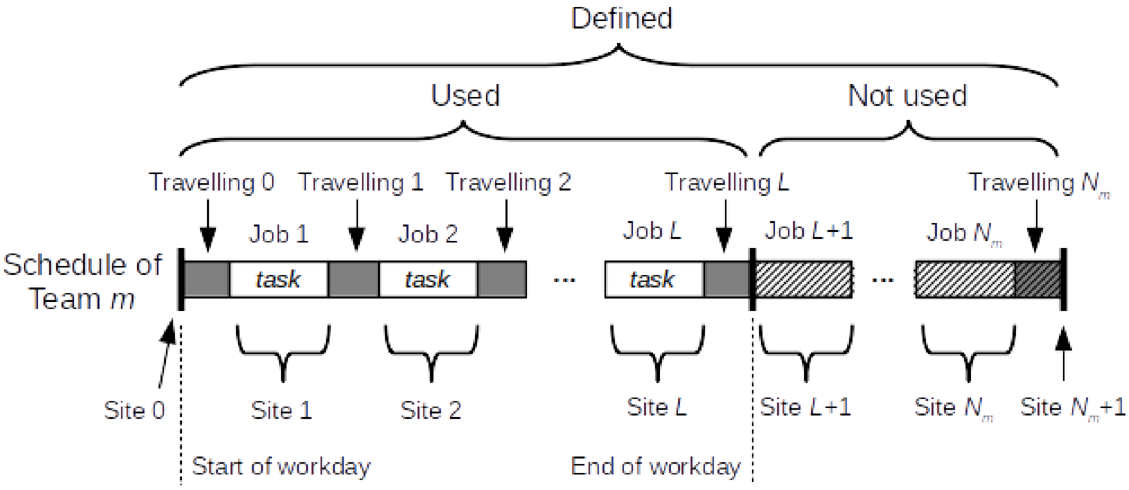


Figure 36: Slots used in the MILP model. Team m has N_m predefined job slots, and in the depicted assignment it executes $L \leq N_m$ tasks.

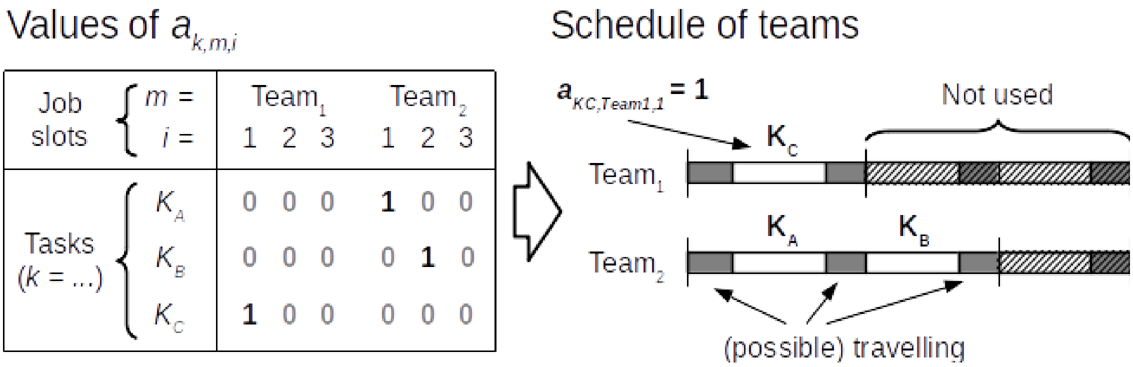
Let M, K, J^{slots} denote the set of teams, tasks and job slots, respectively. Job slots are identified by an ordered pair (m, i), for each $m \in M$ and index i. The numbering denotes order in the job slot sequence of team m, starts with $i = 1$, and ends with $i = N_m$, the total number of job slots of m.

In a feasible solution, the number L of tasks a team m actually executes ranges between 0 and N_m. For a given L, always the first L job slots have tasks assigned, the rest are unused. Job slots do not only define the assignment of tasks to teams, but also the order of execution of tasks assigned to the same team.

N_m is a model parameter and must be decided a priori. Choosing values can be done empirically. $|K| \leq \sum_{m \in M} N_m$ is needed for feasibility, and $|K| = N_m$ guarantees including all theoretically possible solutions, although choosing such a large N_m would result in a very large model.

The main binary decision variables in the MILP model are $a_{k,m,i}$, for each $k \in K$ and $(m, i) \in J^{slots}$, and $a_{k,m,i} = 1$ tells if task k is assigned to job slot (m, i).

An example shown in Figure 37 demonstrates the meaning of $a_{k,m,i}$. Teams $Team_1$ and $Team_2$ must execute tasks K_A, K_B and K_C. The number of predefined job slots is 3 for each team. In the assignment shown, $Team_1$ uses 1 and $Team_2$ uses 2 job slots. Of the 18 binary variables in total, $a_{K_A,Team_2,1}$, $a_{K_B,Team_2,2}$ and $a_{K_C,Team_1,1}$ are the three which are set to 1.


Values of $a_{k,m,i}$

Job slots	m =	$Team_1$			$Team_2$		
	i =	1	2	3	1	2	3
Tasks (k = ...)	K_A	0	0	0	**1**	0	0
	K_B	0	0	0	0	**1**	0
	K_C	**1**	0	0	0	0	0

Schedule of teams

Figure 37: Example usage of the $a_{k,m,i}$ core decision variables.

5.3.2 Allocation constraints

Allocation constraints ensure the basic logic for the assignment of tasks to teams, and also decide the exact site of each team during its schedule.

Task assignment

Task k is assigned to team m (indicated by $a^{task}_{k,m}$) if and only if k is assigned to any job slot (m, i) of m. Since $a^{task}_{k,m}$ is binary, Constraint (5.2) also ensures that any task k is assigned to at most one job slot (m, i) of any team m.

$$a^{task}_{k,m} = \sum_{(m,i)\in J^{slots}} a_{k,m,i} \quad \forall \; k \in K, m \in M \tag{5.2}$$

Any task k is assigned to exactly one team, which is ensured by Constraint (5.3).

$$1 = \sum_{m\in M} a^{task}_{k,m} \quad \forall \; k \in K \tag{5.3}$$

Job slots are used consecutively starting from $i = 1$ without skipping. In other words, a job slot (m, i) can only be used if the previous one, $(m, i - 1)$ is also used, if exists. This is expressed in Constraint (5.4).

$$\sum_{k\in K} a_{k,m,i-1} \geq \sum_{k\in K} a_{k,m,i} \quad \forall \;\; (m, i) \in J^{slots} : i \neq 1 \tag{5.4}$$

Position of teams

Binary variable $b^{present}_{m,i,s}$ indicates whether team m is at site s during its site slot $(m,i) \in X^{slots}$. These are unambiguously calculated by the following constraints.

First, Constraint (5.5) states that any team m must be at exactly one site at any time during its schedule, that is, in each of its site slots $(m,i) \in X^{slots}$. Note that $S^{start}_m = S^{end}_m$ is further assumed in the model.

$$\sum_{s \in S^{tasksites} \cup \left\{S^{start}_m, S^{end}_m\right\}} b^{present}_{m,i,s} = 1 \quad \forall \ (m,i) \in X^{slots} \tag{5.5}$$

In the first site slot ($i = 0$), each team m is at its starting depot.

$$b^{present}_{m,0,S^{start}_m} = 1 \quad \forall \ m \in M \tag{5.6}$$

In the last site slot ($i = N_m + 1$), each team m is at its final depot.

$$b^{present}_{m,N_m+1,S^{end}_m} = 1 \quad \forall \ m \in M \tag{5.7}$$

Finally, Constraint (5.8) ensures that in any job slot (m,i), a team m is at a task site s if and only if a task k with $S^{task}_k = s$ is assigned to (m,i).

$$b^{present}_{m,i,s} = \sum_{k \in K : S^{task}_k = s} a_{k,m,i} \quad \forall \ (m,i) \in J^{slots}, s \in S^{tasksites} \tag{5.8}$$

5.3.3 Travelling and continuity constraints

The following set of constraints expresses travelling rules, the connection between subsequent slots, and some global limits for teams.

Changing sites

Variable b^{sch}_{m,i,s_1,s_2} indicates whether team m moves from site s_1 to s_2 during its travelling slot $(m,i) \in T^{slots}$, and $b^{travel,move}_{m,i}$ indicates whether there is a changing in position at all. These are calculated in the following constraints.

Constraint (5.9) states that for any travelling slot $(m,i) \in T^{slots}$, there is a movement from a site s_1 to a different site s_2 if m is at s_1 in site slot $(m,i) \in X^{slots}$, and m is at s_2 in site slot $(m,i+1) \in X^{slots}$.

$$b^{sch}_{m,i,s_1,s_2} \geq b^{present}_{m,i,s_1} + b^{present}_{m,i+1,s_2} - 1 \quad \forall \ (m,i) \in T^{slots}, s_1, s_2 \in S : s_1 \neq s_2 \tag{5.9}$$

Movement occurs in a travelling slot $(m,i) \in T^{slots}$ if and only if there are two sites s_1 and s_2 for which travelling occurs from s_1 to s_2, this is captured in Constraint (5.10). Note that there can only be one such pair.

$$b_{m,i}^{travel,move} = \sum_{s_1,s_2 \in S: s_1 \neq s_2} b_{m,i,s_1,s_2}^{sch} \quad \forall \ (m,i) \in T^{slots} \tag{5.10}$$

Distance moved in a travelling slot $(m,i) \in T^{slots}$ is calculated in a similar manner, taking into account site distances, by Constraint (5.11).

$$d_{m,i} = \sum_{s_1,s_2 \in S: s_1 \neq s_2} b_{m,i,s_1,s_2}^{sch} D_{s_1,s_2} \quad \forall \ (m,i) \in T^{slots} \tag{5.11}$$

Slot continuity

Constraint (5.12) ensures that any job slot $(m,i) \in J^{slots}$ has a nonnegative length, expressed by the ending time of the previous, and starting time of the next travelling slot.

$$t_{m,i-1}^{travel,end} \leq t_{m,i}^{travel,start} \quad \forall \ (m,i) \in J^{slots} \tag{5.12}$$

The nonnegative length of a travelling slot $(m,i) \in T^{slots}$ is ensured by its calculation formula in Constraint (5.13). Travelling slot $(m,i) \in T^{slots}$ consists of the travelling time of the team m depending on the $d_{m,i}$ distance and V_m travelling speed, the packing and unpacking times, and idle times. Note that if there is no movement, only the idle time can be nonzero.

$$t_{m,i}^{travel,end} - t_{m,i}^{travel,start} = \frac{d_{m,i}}{V_m} + b_{m,i}^{travel,move} \left(U_m^{pack} + U_m^{unpack} \right) + u_{m,i}^{idle} \quad \forall \ (m,i) \in T^{slots} \tag{5.13}$$

Global team limitations

For each team m, the total travelling time is limited by $U_m^{travel,max}$.

$$\sum_{(m,i) \in T^{slots}} \left(t_{m,i}^{travel,end} - t_{m,i}^{travel,start} \right) \leq U_m^{travel,max} \quad \forall \ m \in M \tag{5.14}$$

The total working time for a team m is limited by $U_m^{work,max}$.

$$t_{m,N_m}^{travel,end} - t_{m,0}^{travel,start} \leq U_m^{work,max} \quad \forall \ m \in M \tag{5.15}$$

The total distance travelled by team m is limited by $D_m^{travel,max}$.

$$\sum_{(m,i) \in T^{slots}} d_{m,i} \leq D_m^{travel,max} \quad \forall \ m \in M \tag{5.16}$$

5.3.4 Execution constraints

Several activities take place in a job slot before, after or during a task is executed (see Figure 38). The exact timings of these activities are calculated by the following set of constraints, all formulated for each task $k \in K$. Time windows are also addressed.

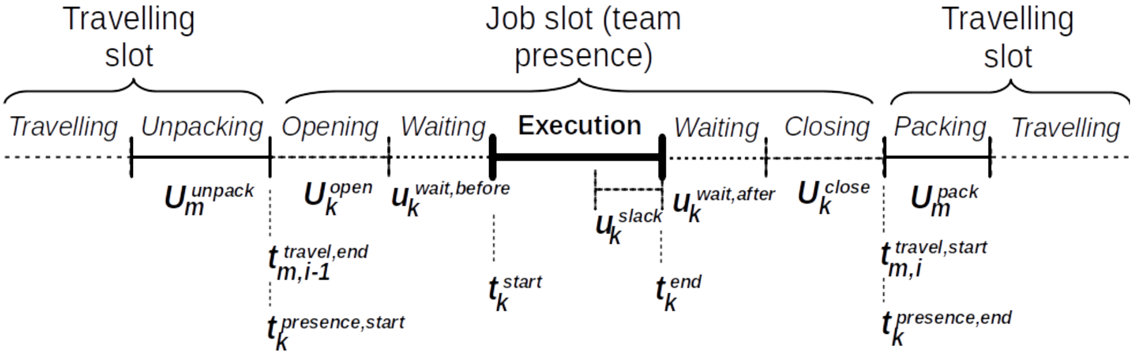


Figure 38: Variables denoting time points and durations of various activities.

Job slot sequencing

The start of presence of the executing team for any task k, indicated by $t_k^{presence,start}$, coincides with the start of the job slot the task is assigned to. Constraint (5.17) expresses this as a big-M constraint, with two inequalities.

$$\begin{aligned} t_k^{presence,start} - t_{m,i-1}^{travel,end} &\geq -U^{workday}\left(1 - a_{k,m,i}\right) \\ t_k^{presence,start} - t_{m,i-1}^{travel,end} &\leq +U^{workday}\left(1 - a_{k,m,i}\right) \end{aligned} \quad \forall\ k \in K, (m,i) \in J^{slots} \tag{5.17}$$

In a similar manner, the end of presence of the executing team for any task k, indicated by $t_k^{presence,end}$, coincides with the end of the job slot the task is assigned to.

$$\begin{aligned} t_k^{presence,end} - t_{m,i}^{travel,start} &\geq -U^{workday}\left(1 - a_{k,m,i}\right) \\ t_k^{presence,end} - t_{m,i}^{travel,start} &\leq +U^{workday}\left(1 - a_{k,m,i}\right) \end{aligned} \quad \forall\ k \in K, (m,i) \in J^{slots} \tag{5.18}$$

In the presence of the executing team until the actual starting of the execution of a task k, indicated by t_k^{start}, nonnegative site opening and waiting times may take place. These additional events are due to possible task relationships and are discussed later.

$$t_k^{presence,start} + p_k^{open} U_k^{open} + u_k^{wait,before} = t_k^{start} \quad \forall\ k \in K \tag{5.19}$$

Similarly, after the actual end of execution and the end of presence of the executing team for a task k, waiting and site closing times may also take place.

$$t_k^{end} + u_k^{wait,after} + p_k^{close} U_k^{close} = t_k^{presence,end} \quad \forall\ k \in K \tag{5.20}$$

Finally, the net execution time of a task k depends on the executing team. If k is in a parallel execution relation, slack time may be added.

$$t_k^{end} - t_k^{start} = u_k^{slack} + \sum_{m \in M} a_{k,m}^{task} U_{k,m}^{exec} \quad \forall\ k \in K \tag{5.21}$$

Time windows

Two kinds of time windows are supported by the model. Any task k must fit in its absolute time window $\left[T_k^{earliest}, T_k^{latest}\right]$, but may be started earlier or finished later than its expected time window $\left[T_k^{expected,start}, T_k^{expected,end}\right]$ in exchange for a penalty cost proportional to the extent of earliness or lateness.

Absolute time windows are implemented by the following two constraints.

$$T_k^{earliest} \leq t_k^{start} \quad \forall\ k \in K \tag{5.22}$$

$$T_k^{latest} \geq t_k^{end} \quad \forall\ k \in K \tag{5.23}$$

The penalty for starting the task k earlier than its expected time window is denoted by variable $c_k^{pen,early}$. Note that this penalty cannot be negative, but is optimized to zero if the task is started in time. The cost factor is $C_k^{earliness}$.

$$\left(T_k^{expected,start} - t_k^{start}\right) C_k^{earliness} \leq c_k^{pen,early} \quad \forall\ k \in K \tag{5.24}$$

Similarly, the penalty for finishing a task k later than its expected time window is denoted by variable $c_k^{pen,late}$. Again, this value is never negative, but is optimized to zero if the task is finished in time. The cost factor is $C_k^{lateness}$.

$$\left(t_k^{end} - T_k^{expected,end}\right) C_k^{lateness} \leq c_k^{pen,late} \quad \forall\ k \in K \tag{5.25}$$

Neither time window is mandatory. To effectively omit the absolute time window, $T_k^{earliest} = T^{day,start}$ and $T_k^{latest} = T^{day,end}$ can be set. To omit the expected time window, $T_k^{expected,start} = T_k^{earliest}$ and $T_k^{expected,end} = T_k^{latest}$ can be set.

5.3.5 Resource constraints

Two types of resources are managed by the model: consumables, which are used up at each task, and tools, which are needed by tasks but are kept.

In each job slot (m, i), the amount of resource r team m needs for task execution is denoted by $q_{r,m,i}^{req}$. To calculate $q_{r,m,i}^{req}$, resource requirements $Q_{r,k,m}^{req}$ for each possibly executed task k are needed. From the values of $Q_{r,k,m}^{req}$, exactly one according to task assignment is selected in Constraint (5.26), or none if the job slot is unused.

$$q^{req}_{r,m,i} = \sum_{k \in K} a_{k,m,i} Q^{req}_{r,k,m} \quad \forall\ r \in R, (m,i) \in J^{slots} \tag{5.26}$$

For any consumable resource r, the amount a team m utilizes is calculated as the sum of the required amounts for all job slots of m.

$$q^{carry}_{r,m} = \sum_{(m,i) \in J^{slots}} q^{req}_{r,m,i} \quad \forall\ r \in R^{cons}, m \in M \tag{5.27}$$

In contrast, for any tool resource r, the amount a team m utilizes is calculated as the maximum of the required amounts for all job slots of m. This value, denoted by $q^{carry}_{r,m}$ is optimized to the actual minimum, which is expressed in Constraint (5.28).

$$q^{carry}_{r,m} \geq q^{req}_{r,m,i} \quad \forall\ r \in R^{tool}, (m,i) \in J^{slots} \tag{5.28}$$

Each team m has a capacity limit $Q^{max}_{r,m}$ for each resource r.

$$q^{carry}_{r,m} \leq Q^{max}_{r,m} \quad \forall\ r \in R, m \in M \tag{5.29}$$

There is also a total availability Q^{cap}_{r} individually for each resource r, which can be shared among teams.

$$\sum_{m \in M} q^{carry}_{r,m} \leq Q^{cap}_{r} \quad \forall\ r \in R \tag{5.30}$$

5.3.6 Task relationship constraints

Each relationship is described by an ordered pair (k_1, k_2) of tasks, although the order is not needed for all relationship types. The model considers precedence relationships (P^{prec}), which have three types: free precedence (P^{free}), same-team precedence (P^{same}) and protected precedence (P^{prot}). There are also mutual exclusion (P^{mtx}) and parallel execution (P^{par}) relationships.

Free and same-team precedence

For any precedence relationship $(k_1, k_2) \in P^{prec}$, task k_1 must be finished before k_2 is started. This is expressed in Constraint (5.31). Note that k_2 can be done by a different team, or the same one in a later job slot. This constraint is in effect for all three types of precedence relationships. For free precedence, this is the only constraint needed.

$$t^{end}_{k_1} \leq t^{start}_{k_2} \quad \forall\ (k_1, k_2) \in P^{prec} = P^{free} \cup P^{same} \cup P^{prot} \tag{5.31}$$

For a same-team precedence relationship $(k_1, k_2) \in P^{same}$, Constraint (5.32) guarantees that tasks k_1 are k_2 are also executed by the same team.

$$a^{task}_{k_1,m} = a^{task}_{k_2,m} \quad \forall\ (k_1, k_2) \in P^{same}, m \in M \tag{5.32}$$

Protected precedence

Protected precedence relationships are more complex. For any $(k_1, k_2) \in P^{prot}$, Constraint (5.31) is imposed to ensure that k_1 is executed before k_2, but another set of constraints is also required. In this description, the first team is the team executing k_1, and the second team is the team executing k_2.

The intention of protected precedence is to model the case when leaving a site unattended between two tasks would be hazardous. There are two options to avoid this. The first option is that the first team may wait until the second one arrives. Note that this automatically happens if k_2 is executed by the same team as k_1, directly afterwards. The second option is that the first team secures the site by a closing activity, then leaves. The second team must then perform an opening operation when arrives. The model also allows the sites of the tasks, $S^{task}_{k_1}$ and $S^{task}_{k_2}$ to be different, although this did not find a viable use case for this.

The solution to choose between the two options is that a binary decision variable $p^{prot}_{k_1,k_2}$ is introduced. The value $p^{prot}_{k_1,k_2} = 0$ indicates that the first option, waiting is chosen. The value $p^{prot}_{k_1,k_2} = 1$ indicates that the second option, closing and opening are chosen.

In case of $p^{prot}_{k_1,k_2} = 0$, Constraint (5.33) ensures that the first team waits the second one. This is a big-M constraint. Note that the teams themselves are not mentioned, $t^{presence,end}_{k_1}$ denotes the time point when the first team leaves and $t^{presence,start}_{k_2}$ denotes when the second team arrives.

$$t^{presence,start}_{k_2} - t^{presence,end}_{k_1} \leq U^{workday} p^{prot}_{k_1,k_2} \quad \forall \ (k_1, k_2) \in P^{prot} \tag{5.33}$$

Note that $u^{wait,before}_{k}$ from Constraint (5.19) and $u^{wait,after}_{k}$ from Constraint (5.20) are the variables representing the waiting times taking place. Both are nonnegative and count towards the presence of the executing team at the task site.

In case of $p^{prot}_{k_1,k_2} = 1$, the binary variables $p^{close}_{k_1}$ and $p^{open}_{k_2}$ indicate whether there is a closing activity after k_1 and an opening activity before k_2, are set to 1.

$$p^{close}_{k_1} \geq p^{prot}_{k_1,k_2} \quad \forall \ (k_1, k_2) \in P^{prot} \tag{5.34}$$

$$p^{open}_{k_2} \geq p^{prot}_{k_1,k_2} \quad \forall \ (k_1, k_2) \in P^{prot} \tag{5.35}$$

The following constraints also ensure that the closing and opening activities can take place only if there is actually a relevant protected precedence relationship with this option chosen.

$$p^{close}_{k_1} \leq \sum_{(k_1,k_2) \in P^{prot}} p^{prot}_{k_1,k_2} \quad \forall \ k_1 \in K \tag{5.36}$$

$$p^{open}_{k_2} \leq \sum_{(k_1,k_2) \in P^{prot}} p^{prot}_{k_1,k_2} \quad \forall \ k_2 \in K \tag{5.37}$$

Mutual exclusion

A mutual exclusion relationship $(k_1, k_2) \in P^{mtx}$ means that k_1 and k_2 cannot be in progress at the same time. The order of k_1 and k_2 is not important.

A binary decision variable $p^{mtx}_{k_1,k_2}$ is introduced for the two possible ways this relationship can be satisfied. The first option, indicated by $p^{mtx}_{k_1,k_2} = 0$, is that k_2 must be finished before k_1 is started. The second option, indicated by $p^{mtx}_{k_1,k_2} = 1$, is the opposite, when k_1 is finished before k_2 is started. Two Big-M constraints are formulated for this purpose. Constraint (5.38) covers the first option, and Constraint (5.39) covers the second option.

$$t^{start}_{k_1} - t^{end}_{k_2} \geq -U^{workday} p^{mtx}_{k_1,k_2} \quad \forall \ (k_1, k_2) \in P^{mtx} \tag{5.38}$$

$$t^{start}_{k_2} - t^{end}_{k_1} \geq -U^{workday} \left(1 - p^{mtx}_{k_1,k_2}\right) \quad \forall \ (k_1, k_2) \in P^{mtx} \tag{5.39}$$

Parallel execution

The parallel execution relationship models situations where two teams must simultaneously execute two different tasks in cooperation. In the modeling point of view, for any parallel execution relationship $(k_1, k_2) \in P^{par}$, it must be ensured that the starting and ending times of k_1 and k_2 coincide, as follows.

$$t^{start}_{k_1} = t^{start}_{k_2} \quad \forall \ (k_1, k_2) \in P^{par} \tag{5.40}$$

$$t^{end}_{k_1} = t^{end}_{k_2} \quad \forall \ (k_1, k_2) \in P^{par} \tag{5.41}$$

Note that the individual net execution times of the two tasks is determined by the parameter $U^{exec}_{k,m}$, and therefore generally end up being distinct for k_1 and k_2. In this case, the faster team adapts to the slower one. This is achieved by the u^{slack}_k variable introduced as an optional increasing term into the formula of the net execution time, in Constraint (5.21). In short, the individual net execution times can be prolonged to be equal. Note that optimization also seeks to minimize the slack time, as it counts towards the working time of the executing team.

5.3.7 Objective function

The different cost components are listed here. Travelling costs are calculated for each team by its distances travelled, moving speed and cost factor.

$$c^{travel} = \sum_{(m,i) \in T^{slots}} \frac{d_{m,i}}{V_m} C^{travel}_m \tag{5.42}$$

Packing costs are coming from packing and unpacking activities from all travelling slots where moving actually takes place. This also applies to moving out from or back into the depot.

$$c^{packing} = \sum_{(m,i)\in T^{slots}} b^{travel,move}_{m,i} \left(C^{pack}_{m} + C^{unpack}_{m}\right) \tag{5.43}$$

Time window costs are composed of earliness and lateness penalties calculated for expected time windows.

$$c^{tw} = \sum_{k\in K} \left(c^{pen,early}_{k} + c^{pen,late}_{k}\right) \tag{5.44}$$

Execution costs are based solely on the team each task is assigned to.

$$c^{exec} = \sum_{k\in K} \sum_{(m,i)\in J^{slots}} a_{k,m,i} C^{exec}_{k,m} \tag{5.45}$$

Resource costs are derived from the total amounts utilized by teams.

$$c^{res} = \sum_{r\in R} \sum_{m\in M} q^{carry}_{r,m} C^{res}_{r} \tag{5.46}$$

Opening and closing costs are incurred for opening and closing activities performed due to protected precedence relationships.

$$c^{opcl} = \sum_{(k_1,k_2)\in P^{prot}} p^{prot}_{k_1,k_2} C^{opcl}_{k_1,k_2} \tag{5.47}$$

Working time costs for each team are proportional to the total time the team spends in duty.

$$c^{work} = \sum_{m\in M} \left(t^{travel,end}_{m,N_m} - t^{travel,start}_{m,0}\right) C^{work}_{m} \tag{5.48}$$

The objective is the total of the aforementioned components.

$$\text{minimize: } c^{total} = c^{travel} + c^{packing} + c^{tw} + c^{exec} + c^{res} + c^{opcl} + c^{work} \tag{5.49}$$

5.4 Algorithmic framework

Although the standalone MILP solution method works well for small examples, it can quickly become computationally too difficult. For this reason, an algorithmic framework was also developed which offers heuristic solutions for larger problem instances in an acceptable amount of time. Note that some problem features may interfere with the algorithm, as detailed later.

5.4.1 Algorithm description

For the sake of description, a **schedule** of a subset of tasks is defined as a proper assignment of these tasks to job slots of teams according to assignment rules. Job slots of the same team m are used from the first slot $i = 1$ without skipping an index, this is the same logic as for the MILP model. Therefore, a schedule determines the assignment of a subset of tasks to teams and the order each team performs the task assigned to it, but nothing more specific. Particularly, a schedule does not determine exact timings.

The algorithmic framework obtains a schedule incrementally, introducing new tasks one by one and assigning them to teams. This can be concluded in five steps.

1. Start from the initial schedule with no tasks.
2. Choose a new task not in the existing schedule.
3. Put the new task into the existing schedule obtaining a new schedule.
4. Update decisions (timings, resource and cost calculations, and others) according to the new schedule.
5. Repeat steps 2-4 until all tasks are schedule with all tasks is obtained.

Steps 2-4 together are called an **iteration** of the algorithm. An iteration starts with a given schedule of tasks and ends with a new schedule with one more task.

The key idea of the algorithm is that an iteration is done by solving a single instance of a **modified MILP model**, obtained from the standalone MILP model. The key differences between the two models are the following.

- The existing schedule is an input for the modified MILP model, and is used as a restriction on possible assignments.
- It is no longer required to assign all tasks. Instead, all tasks from the existing schedule must be executed, and a single new, unscheduled task is to be selected and executed.
- The result of the model is a new schedule with one more tasks than the existing schedule, for which the total cost is minimal.

The existing schedule is maintained in a way that the assignment and relative order of the already scheduled tasks cannot change. It is allowed for new tasks to be inserted in between already scheduled tasks, pushing some scheduled tasks forward by a single job slot. The complete iteration is illustrated in Figure 39.

Although the modified MILP model consists of more variables and constraints, its search space is significantly reduced so it can be solved in an acceptable amount of time. The algorithm solves the modified MILP model $|K|$ times, adding a single

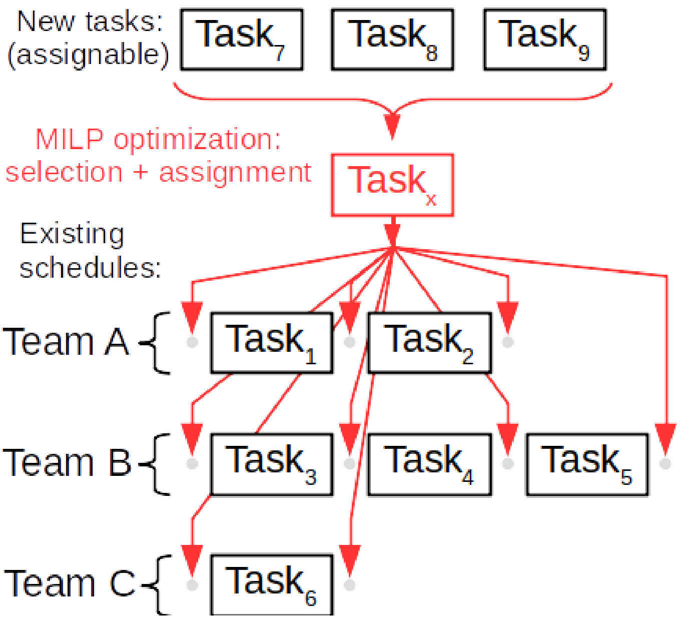


Figure 39: Decision scheme of a single iteration of the solution algorithm.

task each time. Since all other constraints are also included, the last solution of the modified MILP model eventually schedules all tasks, and is therefore a feasible solution for the original problem.

The only feature of the standalone MILP method which is not supported by the algorithmic framework are task relationships. The reason is that constraints, including relationship constraints of unscheduled tasks are simply ignored. Therefore, early decisions on tasks may easily turn out to be too restrictive. The efficient management of relationships in the algorithmic framework is subject to future research.

5.4.2 The modified MILP model

At the beginning of an iteration, the set of tasks, K is divided into the set of tasks already scheduled, K^{done}, and those not scheduled yet, K^{rem}. Each task $k \in K^{done}$ already scheduled has a job slot $H_k^{slot} \in J^{slots}$ it is currently assigned to. The predefined number of job slots N_m for team m is chosen in such a way that each team has as many job slots as tasks already assigned to it in the existing schedule, plus one. In this way, the newly selected task can be assigned to any team. These conclude the additional input data required by the modified MILP model. Note that slot sets J^{slots}, T^{slots} and X^{slots} are also updated for each iteration based on the current values of N_m.

The following decisions are made in the model.

- A single remaining task $k \in K^{rem}$ is selected, which is going to be included in the new schedule. This decision is indicated by the new binary variable x_k^{task}, for each $k \in K^{rem}$, for which the value 1 means that the task k is selected.

- A single team is selected to which the selected task is assigned to. This decision is represented by the new binary variable x_m^{team}, which is 1 when team m is selected. The schedules of other teams remain unchanged.
- A single job slot $(m, i) \in J^{slots}$ is selected where the selected task k is inserted into the schedule of the selected team m. The new binary variable $x_{m,i}^{slot}$ represents this decision, a value of 1 means that the particular (m, i) is chosen.

Note that $x_{m,i}^{slot}$ determines x_m^{team}, but for modeling purposes it is easier to introduce both. Another unambiguously determined, new binary variable $y_{m,i}$ is introduced, which takes the value 1 if and only if the new task is inserted before travelling slot $(m, i) \in T^{slots}$ into the schedule of team m.

The key decision variables are illustrated in Figure 40. Team m already has a schedule with tasks K_1, K_5 and K_2 in this order. In the next iteration of the algorithm, a new task K_4 is selected, and is inserted in between K_5 and K_2. Note that other teams may also be present.

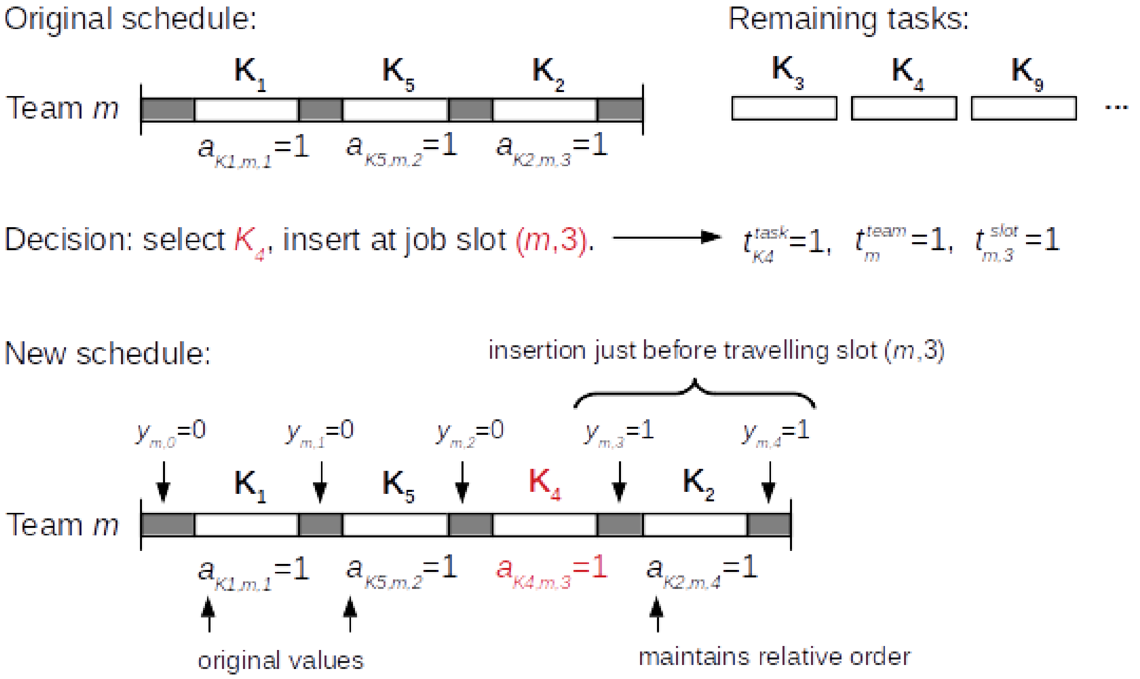


Figure 40: Example usage of decision variables in the algorithmic framework.

Constraint (5.3) ensured that any task k is assigned to exactly one team. This is the only constraint dropped in the modified MILP model. The other parts of the original model, the existing schedule as input, and the newly introduced variables x_k^{task}, x_m^{team}, $x_{m,i}^{slot}$ and $y_{m,i}$ together should determine the new schedule. To achieve this, a new set of constraints is needed, presented below.

No task can be inserted before the first travelling slot $(m, 0)$ of any team m.

$$y_{m,0} = 0 \quad \forall\ m \in M \tag{5.50}$$

Insertion of the selected task before a travelling slot $(m, i) \in T^{slots}$ is possible in two mutually exclusive ways. Either it can be inserted just before, to job slot $(m, i) \in J^{slots}$, or it can be inserted even earlier, before the previous travelling slot $(m, i-1) \in T^{slots}$. Using this rule, Constraint (5.51) completes the definition of all $y_{m,i}$ based on the other variables.

$$y_{m,i} = y_{m,i-1} + x^{slot}_{m,i} \quad \forall\ \ (m, i) \in T^{slots} : i \neq 0 \tag{5.51}$$

Inserting the selected task before any travelling slot $(m, i) \in T^{slots}$ is only possible if team m is also selected.

$$x^{team}_{m} \geq y_{m,i} \quad \forall\ \ (m, i) \in T^{slots} \tag{5.52}$$

A new task $k \in K^{rem}$ is assigned to a job slot $(m, i) \in J^{slots}$ if and only if k is the selected task, and (m, i) is the selected job slot. The connection of the new decision variables with the original $a_{k,m,i}$ is established in the following three constraints.

$$a_{k,m,i} \leq x^{task}_{k} \quad \forall\ k \in K^{rem}, (m, i) \in J^{slots} \tag{5.53}$$

$$a_{k,m,i} \leq x^{slot}_{m,i} \quad \forall\ k \in K^{rem}, (m, i) \in J^{slots} \tag{5.54}$$

$$a_{k,m,i} \geq x^{task}_{k} + x^{slot}_{m,i} - 1 \quad \forall\ k \in K^{rem}, (m, i) \in J^{slots} \tag{5.55}$$

Any task $k \in K^{done}$ already scheduled must be scheduled again, so they are assigned to exactly one team. But any task $k \in K^{rem}$ not scheduled yet is scheduled if and only if it is selected.

$$1 = \sum_{m \in M} a^{task}_{k,m} \quad \forall\ k \in K^{done} \tag{5.56}$$

$$x^{task}_{k} = \sum_{m \in M} a^{task}_{k,m} \quad \forall\ k \in K^{rem} \tag{5.57}$$

It is also ensured by the following constrains that exactly one task, one team and one job slot is selected in total.

$$1 = \sum_{k \in K^{rem}} x^{task}_{k} \tag{5.58}$$

$$1 = \sum_{m \in M} x^{team}_{m} \tag{5.59}$$

$$1 = \sum_{(m,i) \in J^{slots}} x^{slot}_{m,i} \tag{5.60}$$

Finally, the exact positions of already scheduled tasks $k \in K^{done}$ are also fixed. Suppose that the original job slot for k was $H^{slot}_k = (m,i)$. Then there are two cases. First, if the selected task was inserted before travelling slot $(m,i) \in T^{slots}$, then the new job slot of k is $(m, i+1)$. Second, if the selected task was not inserted before travelling slot $(m,i) \in T^{slots}$, then the job slot of k remains the same, (m,i).

$$y_{m,i} = a_{k,m,i+1} \quad \forall\ k \in K^{done} : (m,i) = H^{slot}_k \tag{5.61}$$

$$1 - y_{m,i} = a_{k,m,i} \quad \forall\ k \in K^{done} : (m,i) = H^{slot}_k \tag{5.62}$$

For already scheduled tasks $k \in K^{done}$, other assignment variables can be explicitly set to zero. Although Constraint (5.63) is redundant with the previous ones, it may help in MILP model preprocessing.

$$0 = a_{k,m,j} \quad \forall\ k \in K^{done}, (m,j) \in J^{slots} : (m,i) = H^{slot}_k, j \notin \{i, i+1\} \tag{5.63}$$

5.5 Computational results

To demonstrate the effectiveness of the proposed standalone MILP method and the algorithmic framework, and to investigate how certain model elements affect performance, several series of tests were performed.

All problem data and results to be presented here are available as supplementary material in the main publication [S5], and also on the web [112]. This material includes the implementation of both the MILP model and the algorithmic framework, and also the testing procedure itself in an executable format.

Note that a motivational problem was presented in Subsection 5.2.2, with its full solution using the standalone MILP method. In this section, the focus is rather on the obtained objective values, running times and some other statistics for the two solution methods, but the solutions themselves are not represented in detail. Also note that these tests are intended to give a glance, but are not exhaustive.

5.5.1 Overview of standalone MILP model testing

Three test series were performed with the standalone MILP method, each trying to investigate the impact of a particular property of the problem instance on solver performance. The three properties are the following.

- Number of different task sites, $\left|S^{tasksites}\right|$.

- Number of task relationships.
- Number of predefined job slots, which is the parameter N_m.

A set of mobile workforce management problem instances were constructed in the following way. The basis is a single problem instance called the **main problem instance**, with the following properties. Full details and description can be found in the supplementary material.

- $|M| = 3$ teams must execute $|K| = 18$ tasks spread over $\left|S^{tasksites}\right| = 4$ sites.
- All teams have own constant travelling and execution times and costs, but all have $N_m = 6$ predefined job slots.
- Task have absolute and expected time windows, which slightly reduce the workday starting at 08:00 and ending at 16:00.
- There is a single consumable and a single tool resource.
- Five of the tasks mutually exclude each other, the others have all the supported other kinds of relationships.

A set of test instances for each test series was obtained by described, slight modifications of the main problem instance, according to the property in focus, resulting in a series of similar problem instances.

The MILP model was implemented in the GNU MathProg modeling language. Solutions for all instances were obtained by the Gurobi 8.1 commercial MILP solver, on a desktop PC with Ubuntu 18.04.1 LTS, Intel i7-4770 3.40 GHz CPU and 16 GB RAM. The time limit was one hour per test case.

The data obtained and presented from the test series are the optimal (or best) objective reported (in EUR), and the running time of the solver. The number of constraints (rows), variables (columns) are also shown, as well as the number of integer (binary) variables. Note that these latter data should be interpreted with caution, as preprocessing steps of an MILP solver can greatly reduce these dimensions, and also there are some strong knapsack-type constraints for the binary decision variables which can shrink model complexity well.

5.5.2 Number of task sites

In the first series, 9 problem instances were constructed as follows.

- Data are the same as for the main problem instance, except for task sites.
- Values of 3, 4 and 5 for $\left|S^{tasksites}\right|$ was considered. Note that the main problem instance has 4 task sites.

Table 14: Effect of site count on solver performance for the standalone MILP model. In all instances, the model has 3096 rows, 1794 columns and 337 integer variables.

Sites	Variation	Solver runtime	Gap	Objective
3	#1	593.33 s	-	14,482 EUR
3	#2	1,048.95 s	-	15,020 EUR
3	#3	1,015.77 s	-	15,018 EUR
4	#1 (main)	285.45 s	-	13,842 EUR
4	#2	2,348.23 s	-	14,834 EUR
4	#3	939.03 s	-	14,844 EUR
5	#1	977.29 s	-	14,338 EUR
5	#2	3,600.00 s	14.44%	16,286 EUR
5	#3	3,600.00 s	13.00%	16,074 EUR

- For each value of $\left|S^{tasksites}\right|$, three variations were constructed, differing only in site distribution, but the occurrence of sites among tasks remains well-balanced, which means that the difference is at most 1. The main problem instance is one of the variations.

Results obtained for these 9 problem instances are shown in Table 14.

It can be seen that site count has a huge impact on performance. The more sites there are, the more difficult the model becomes to solve, although the objective only changes slightly. The variations turned out to be more difficult than the main problem instance, this was preserved even after changing the site count. Two cases did not finish in one hour. However, a gap between the best found and the possible best objectives can be given in such cases. It is also apparent that model dimensions like row, column and integer variable counts are the same. This is due to the fact that the unused task sites are still present in all problem instances, just their corresponding variables are unused if a site does not appear among tasks, resulting in a great effective reduction in complexity.

Overall, this slot-based standalone MILP approach seems effective, but only when the number of different sites is small. The model works best for scenarios with only a few sites and many tasks shared among these.

5.5.3 Number of task relationships

Relationships between tasks restrict the search space, although introduce more constraints and even variables in some cases, so their overall effect is in question.

In the second series, the effect of mutual exclusion and free precedence was probed. The instance set was constructed as follows.

- From the main problem instance, which is also included in the set, 1, 2 and 3 of the existing free precedence relationships were excluded one by one, this resulted in 4 test instances.

Table 15: Effect of free precedence and mutual exclusion (mutex) relationships on solver performance for the standalone MILP model. In all instances, the objective was 13,842 EUR.

Problem instance	Rows	Columns	Integers	Solver runtime
main instance	3096	1794	337	285.45 s
-1 free precedence	3095	1794	337	307.65 s
-2 free precedences	3094	1794	337	243.81 s
-3 free precedences	3093	1794	337	199.77 s
-1 mutex	3094	1793	336	119.57 s
-2 mutexes	3092	1792	335	208.87 s
-3 mutexes	3090	1791	334	125.90 s
-4 mutexes	3088	1790	333	258.38 s
-5 mutexes	3086	1789	332	236.65 s
-6 mutexes	3084	1788	331	242.69 s
-7 mutexes	3082	1787	330	392.80 s
-8 mutexes	3080	1786	329	165.05 s
-9 mutexes	3078	1785	328	256.34 s
-10 mutexes	3076	1784	327	338.31 s

- From the main problem instance, 1 to 10 of the existing mutual exclusion relationships were excluded, this resulted in 10 more test instances.

Results obtained for these 14 problem instances are shown in Table 15.

The objective was the same in all cases. Row, column and integer variable count only differs slightly, as expected by the removal of the specific constraints and variables regarding the excluded relationships.

Removing the free precedence relationships seemingly made the model easier to solve, except for the first removal. Removing mutual exclusion relationships one by one made solver running times lower and higher. The were no differences in magnitude in either case.

It seems like such a small test series is inadequate to show general tendencies regarding these task relationships. The actual effect of a relationship constraint might be difficult to predict, and depend on the instance as well.

5.5.4 Number of predefined job slots

One drawback of the standalone MILP approach is that the number N_m of predefined job slots for each team m must be given as a parameter. Low values may exclude valuable solutions from the search space, but high values may increase the model size and its complexity.

The smallest values for N_m which theoretically guarantee the globally optimal solution in all edge cases are very high, and it may be tedious to strengthen this

Table 16: Effect of predefined job slot count and task count on solver performance for the standalone MILP model.

Problem instance	Rows	Columns	Integers	Solver runtime	Objective
motivational problem	1208	669	135	0.73 s	1,944 EUR
main problem instance	3096	1794	337	285.45 s	13,842 EUR
-1 task	2992	1762	319	93.77 s	13,507 EUR
-2 tasks	2889	1730	301	91.65 s	13,202 EUR
-3 tasks	2785	1698	283	35.73 s	12,815 EUR
-3 tasks, -1 job slot	2653	1627	268	21.74 s	12,815 EUR
-3 tasks, -2 job slots	2521	1556	253	24.54 s	12,963 EUR
-3 tasks, -3 job slots	2389	1485	238	13.40 s	13,133 EUR
+1 job slot	3243	1868	355	225.96 s	13,664 EUR
+2 job slots	3390	1942	373	479.66 s	13,664 EUR
+3 job slots	3537	2016	391	725.77 s	13,664 EUR

bound for a specific problem. N_m can be chosen empirically.

In this third series, task numbers as well as N_m are changed to obtain the instance set, as follows.

- The motivational problem that was shown in full detail is also included to illustrate problem sizes. Note that for the motivational problem, $|M| = 2$, $|K| = 8$, and for both teams $N_m = 8$.
- The main problem instance is also included. Note that for the main problem instance, $|M| = 3$, $|K| = 18$ and for all three teams $N_m = 6$.
- First, 1 to 3 tasks were excluded from the main problem instance to obtain 3 new instances.
- From the last instance where 3 tasks are missing, 1 to 3 job slots were further removed by decreasing one of N_m for each team by 1. This resulted in 3 new instances, the last one having $|K| = 15$ tasks and $N_m = 5$ for each team.
- Finally, based on the main problem instance again, 1 to 3 job slots were added, giving 3 new instances, the last one having $N_m = 7$ for each team.

Results for these 11 instances altogether are shown in Table 16.

Decreasing the task count decreases the running time as well. It is not so obvious that decreasing the number of job slots as well has the same effect, although it is expected to be. The objective becomes worse by removing job slots, this indicates that decreasing N_m actually excludes valuable solutions.

Adding the first job slot results in a better solution in less computational time. But adding further job slots does not decrease the objective, indicating that there

Table 17: Comparison of the standalone MILP solution and the algorithmic framework. Objective values are in EUR.

Task	standalone MILP			algorithm		Difference of
count	Objective	Runtime	Gap	Objective	Runtime	objectives
5	1,604.13	0.31 s	-	1,648.40	0.34 s	2.76%
6	1,812.05	0.63 s	-	1,823.87	0.41 s	0.65%
7	2,192.30	1.39 s	-	2,192.30	0.53 s	0
8	2,151.56	48.07 s	-	2,269.55	0.75 s	5.48%
9	2,416.80	203.79 s	-	2,518.08	0.79 s	4.19%
10	2,467.14	2,600.96 s	-	2,607.26	1.31 s	5.68%
11	2,714.46	3,600.00 s	13.39%	2,875.99	1.59 s	5.95% – 20.14%
12	2,933.69	3,600.00 s	15.75%	3,130.61	1.88 s	6.71% – 23.52%
13	2,934.91	3,600.00 s	20.10%	3,076.68	2.09 s	4.83% – 25.90%
14	3,170.23	3,600.00 s	21.79%	3,385.92	2.10 s	6.80% – 30.08%
15	3,165.45	3,600.00 s	22.77%	3,378.81	2.69 s	6.74% – 31.05%

are no better solutions if other teams are also allowed to execute more tasks. The problem size and the running time, however, increases significantly, as expected. The last running time of 12 min may suggest that this main problem instance is roughly the practical limit for this approach. Larger problems could still be managed, and MILP solvers can report useful suboptimal solutions and provide a lower bound for the optimum.

5.5.5 Testing the algorithmic framework

To demonstrate the usability of the algorithmic framework relying on the modified MILP model, and also a comparison with the standalone MILP approach, a different series of tests was conducted. Instances were randomly generated for $|M| = 3$ teams situated at different depots and $\left|S^{tasksites}\right| = 6$ sites, with gradually increased task count and corresponding N_m, which is later explained. For a fair comparison, smaller problems were actually a subset of larger ones in terms of problem data.

The formula shown in Equation (5.64) was used to determine N_m for the standalone MILP method. Note that in this way, there are at least 20% more job slots in total than the number of tasks.

$$N_m = \left\lfloor 1.2\frac{|K|}{|M|} + 1 \right\rfloor \quad \forall\ m \in M \tag{5.64}$$

The algorithmic framework uses the same Gurobi MILP solver for running the modified MILP model $|K|$ times. This gives the most of the running time of the framework. File parsing can also be significant for smaller problems.

The results of this test series for 5 to 15 tasks is given in Table 17.

$|K| = 15$ was the largest instance for which the standalone MILP model was used. Running times increased rapidly with task count, from $|K| = 11$ the one hour time limit was exceeded, and the gap also started to become worse. Nevertheless, the Gurobi commercial MILP solver still managed to find suboptimal solutions.

The algorithmic framework finished very fast for the same problem sizes, in 2.69 s for $|K| = 15$. Below $|K| = 11$, the reported objective was at most 5.68% worse than the optimum found by the standalone MILP method. Notably, for $|K| = 7$ the algorithm succeeded to find the same solution as the standalone MILP method.

From $|K| = 11$, the solutions obtained by the algorithm are similarly worse as for smaller $|K|$, by at most 6.80%. Note that in this case, if the MILP solver had been allowed to complete the solution of the standalone MILP model, this gap could theoretically be much higher, although this is impractical due to time requirements.

The series was continued for the algorithmic framework only, the results are shown in Table 18 with jumps of five tasks until $|K| = 130$ where the one hour time limit was exceeded.

The running time gracefully worsens as $|K|$ increases and the size of the problem explodes. The algorithmic framework finished in an hour for a magnitude larger number of tasks, 125 compared to 10.

There was a special case for $|K| = 80$, where the very last task could not be scheduled in a feasible way, so the final solution only executed 79 tasks. This is possible due to the heuristic nature. Decisions made early may make the continuation of the algorithm infeasible, and an existing schedule might be impossible to continue and may need to be rescheduled. This is more likely when relationships between tasks are allowed, but may happen due to other components like absolute time windows, too large distances, too few job slots, or other parameters.

5.5.6 Concluding results

Both the standalone MILP method and the algorithmic framework are capable of solving mobile workforce management problems. The problem specification supports a range of parameters including packing and unpacking times, time windows, resource management and task relationship constraints.

The standalone MILP method can reach optimal solutions for small problem instances. The advantage of this approach is that valuable suboptimal results are obtained even before the optimum is found or proven, depending on the MILP solver. The disadvantage of the approach is that the number of predefined job slots N_m must be provided as a parameter, and solver performance is very sensitive to changes in N_m. Also, the model works best for a small number of task sites.

The algorithmic framework is applicable on much larger instances and in the tests performed, provided objective values not significantly worse than the ones reported by the standalone MILP method, although this result can be specific to this study. The values of N_m are also not required, but managed by the algorithm. The downside of the algorithm is the heuristic nature which does not guarantee the

Table 18: Performance of the algorithmic framework on larger instances. Note that for $|K| = 80$, the algorithm failed to finish.

Task count	Integers in MILP model	Objective	Runtime
20	466	3,961.96 EUR	4.87 s
25	706	4,740.18 EUR	8.42 s
30	996	5,434.43 EUR	13.66 s
35	1336	6,131.49 EUR	20.59 s
40	1726	6,682.10 EUR	31.74 s
45	2166	7,254.24 EUR	44.82 s
50	2656	7,960.95 EUR	68.37 s
55	3196	9,296.89 EUR	119.25 s
60	3786	9,244.28 EUR	186.19 s
65	4426	9,703.81 EUR	182.64 s
70	5116	10,626.84 EUR	212.01 s
75	5856	11,522.22 EUR	429.95 s
80	*	*	541.33 s
85	7486	12,741.65 EUR	802.32 s
90	8376	13,032.99 EUR	794.29 s
95	9316	13,892.74 EUR	1,008.72 s
100	10306	14,056.89 EUR	1,265.55 s
105	11346	14,776.31 EUR	1,475.97 s
110	12436	15,753.90 EUR	1,884.86 s
115	13576	15,713.98 EUR	2,236.08 s
120	14766	16,363.13 EUR	2,354.14 s
125	16006	17,057.40 EUR	3,054.47 s
130	17296	17,387.84 EUR	4,151.19 s

optimal solution, may fail in some cases, and in its current form, unlike an MILP solver, it cannot report suboptimal solutions or lower bounds before completion.

The performed tests can provide a glance at how these approaches work, although more would be needed to measure their capabilities and make more general claims. There are other strategies for choosing binary decision variables in the MILP model which might worth trying. The algorithmic framework also has possibilities for development. More sophisticated methods to avoid infeasible solutions even for task relationships may be an option. The implementation can also be optimized, as in the current form, the GNU MathProg language was used for the MILP model, but management through an MILP solver API could be much faster.

5.6 Thesis summary

Thesis 3. A new method was developed and tested for solving mobile workforce management problems. This approach can handle several problem characteristics at once. The method is based on an MILP model which can be solved either in a standalone way or as part of an algorithmic framework which is capable of providing heuristic solutions for larger problem instances. The capabilities and limitations of both methods were investigated through a set of tests.

Related publications: [S1], [S5].

T3.1. An MILP model was developed which solves the mobile workforce management problem specification. The novelty is that a slot-based modeling technique was used, and a range of problem characteristics are covered including packing and unpacking times, time windows, resource and task relationship constraints.

T3.2. An algorithmic framework was developed which uses a modified version of the MILP model to heuristically add tasks to the existing schedule until a final solution is obtained. This method does not guarantee the optimal solution unlike the standalone MILP approach, and has other limitations, but succeeds in providing feasible solutions near the optimal in an acceptable amount of time, even for large problem instances.

Summary of accomplishments

In the recent years, I took part in multiple projects, different in goals and tools used. My jobs in these projects resulted in new optimization models for particular problems, general modeling techniques and case study evaluations.

The first presented result, from Chapter 3, involved the optimization of the energy supply of a manufacturing plant. This problem motivated the development of a new P-Graph model, which has two novelties. First, the pelletizer and biogas plant equipment units were modeled in a special way, later called the flexible input scheme. Second, a multi-period version of the P-Graph model was provided, which separates the mid-year and winter periods. Both features aimed at obtaining a more precise model of the real-world situation.

The pelletizer and biogas plant models motivated a generalization of the method. This resulted in a general modeling technique of operations with flexible inputs, presented in Chapter 4, which allows independent inputs and arbitrary linear constraints. This approach has an advantage of relying solely on the software tools and solution algorithms of the P-Graph framework. A case study demonstration involving biomass-based energy production in a rural region was also given, for which the developed P-Graph model is to be published in the future.

The mobile workforce management model and solution algorithm, which were shown in Chapter 5, are the results of another research. This was not motivated by a particular real-world problem. Instead, the intention was to provide a governing approach for a range of different mobile workforce management problem features. A slot-based MILP model was formulated, followed by an algorithmic framework for larger problem instances. The key novelty of the algorithm is that the MILP model itself is used by a greedy heuristic to construct a schedule.

Overall, these examples show that optimization is a topic where new results can be obtained at different levels and with different tools. There are possible future research directions based on these results as well. Even a particular model as provided for the manufacturing plant can be reused in a similar case study with relevant data. The P-Graph approach makes it convenient to introduce other technologies. The flexible operation model can be used in future P-Graph models, may become supported by software implementation, and other, similar P-Graph „design patterns” could be invented. The mobile workforce management algorithm could also be improved in many different ways to be faster, to consider more options, and avoid bad decisions.

Related publications

[S1] **A. Éles**, H. Cabezas, and I. Heckl, “Heuristic algorithm utilizing mixed-integer linear programming to schedule mobile workforce,” *Chemical Engineering Transactions*, vol. 70, pp. 895–900, 2018.

[S2] **A. Éles**, L. Halász, I. Heckl, and H. Cabezas, “Energy consumption optimization of a manufacturing plant by the application of the p-graph framework,” *Chemical Engineering Transactions*, vol. 70, pp. 1783–1788, 2018.

[S3] **A. Éles**, L. Halász, I. Heckl, and H. Cabezas, “Evaluation of the energy supply options of a manufacturing plant by the application of the p-graph framework,” *Energies*, vol. 12, no. 8, 2019, **IF: 2.707**.

[S4] **A. Éles**, I. Heckl, and H. Cabezas, “Modeling technique in the p-graph framework for operating units with flexible input ratios,” *Central European Journal of Operations Research*, vol. 29, no. 2, pp. 463–489, 2020, **IF: 2.345**.

[S5] **A. Éles**, I. Heckl, and H. Cabezas, “New general mixed-integer linear programming model for mobile workforce management,” *Optimization and Engineering*, vol. 23, pp. 479–525, 2022, **IF: 2.760**.

[S6] **A. Éles**, I. Heckl, and H. Cabezas, “Modeling renewable energy systems in rural areas with flexible operating units,” *Chemical Engineering Transactions*, vol. 88, pp. 643–648, 2021.

Other publications

[E1] **A. Éles** and I. Heckl, "Application of the maximal bipartite matching algorithm to schedule medical appointments with quotas," in *Proceedings of the Pannonian Conference on Advances in Information Technology (PCIT'2019)*, I. Vassányi, Ed., ISBN 978-963-127-8, University of Pannonia, Veszprém, Hungary, 2019, pp. 1–7.

[E2] **A. Éles** and I. Heckl, "Solving an extended line balancing optimization problem using dynamic programming," in *Proceedings of the Pannonian Conference on Advances in Information Technology (PCIT'2020)*, I. Vassányi, Ed., ISBN 978-963-396-144-5, University of Pannonia, Veszprém, Hungary, 2020, pp. 31–37.

[E3] **A. Éles** and I. Heckl, "Dynamic programming approaches for line balancing problems," in *Short papers of VOCAL 2022 (9th VOCAL Optimization Conference: Advanced Algorithms)*, ISBN 978-615-01-5987-4, Hungarian Operations Research Society, Budapest, Hungary, 2022, pp. 30–35.

References

[1] US Census Bureau. "U.s. and world population clock," [Online]. Available at: https://www.census.gov/popclock/ Accessed: 02/09/2022.

[2] J. Lan, A. Malik, M. Lenzen, D. McBain, and K. Kanemoto, "A structural decomposition analysis of global energy footprints," *Applied Energy*, vol. 163, pp. 436–451, 2016.

[3] Worldwatch Institute. "The state of consumption today." (2010), [Online]. Available at: http://www.worldwatch.org/node/810 Accessed: 04/03/2017.

[4] S. W. Running, "A measurable planetary boundary for the biosphere," *Science*, vol. 337, no. 6101, pp. 1458–1459, 2012.

[5] International Energy Agency. "Electricity statistics," [Online]. Available at: https://www.iea.org/statistics/electricity/ Accessed: 02/20/2019.

[6] F. Friedler, K. Tarjan, Y. W. Huang, and L. T. Fan, "Graph-theoretic approach to process synthesis: Axioms and theorems," *Chemical Engineering Science*, vol. 47, pp. 1973–1988, 1992.

[7] F. Friedler, B. J. Varga, E. Fehér, and L. T. Fan, "State of the art in global optimization," in C. A. Floudas and P. M. Pardalos, Eds. Dordrecht: Kluwer Academic Publishers, 1996, ch. Combinatorially Accelerated Branch-and-Bound Method for Solving the MIP Model of Process Network Synthesis, pp. 609–626.

[8] "P-graph official website," [Online]. Available at: https://p-graph.org Accessed: 02/09/2022.

[9] F. Friedler, K. Tarjan, Y. W. Huang, and L. T. Fan, "Graph-theoretic approach to process synthesis: Polynomial algorithm for the maximal structure generation," *Computers & Chemical Engineering*, vol. 17, no. 9, pp. 929–942, 1993.

[10] F. Friedler, J. B. Varga, and L. T. Fan, "Decision-mapping: A tool for consistent and complete decisions in process synthesis," *Chemical Engineering Science*, vol. 50, no. 11, pp. 1755–1768, 1995.

[11] V. Varga, I. Heckl, F. Friedler, and L. T. Fan, “Pns solutions: A p-graph based programming framework for process network synthesis,” *Chemical Engineering Transactions*, vol. 21, pp. 1387–1392, 2010.

[12] M. A. B. Promentilla, R. I. G. Lucas, K. B. Aviso, and R. R. Tan, “Problem-based learning of process systems engineering and process integration concepts with metacognitive strategies: The case of p-graphs for polygeneration systems,” *Applied Thermal Engineering*, vol. 127, pp. 1317–1325, 2017.

[13] H. L. Lam, R. R. Tan, and K. B. Aviso, “Implementation of p-graph modules in undergraduate chemical engineering degree programs: Experiences in malaysia and the philippines,” *Journal of Cleaner Production*, vol. 136, pp. 254–265, 2016, PRES’15: Cleaner energy planning, management and technologies.

[14] A. Bartos and B. Bertok, “Parameter tuning for a cooperative parallel implementation of process-network synthesis algorithms,” *Central European Journal of Operations Research*, vol. 27, pp. 551–572, 2019.

[15] J. Klemes and P. Varbanov, “Spreading the message: P-graph enhancements: Implementations and applications,” *Chemical Engineering Transactions*, vol. 45, pp. 1333–1338, 2015.

[16] K. Kalauz, Z. Süle, B. Bertok, F. Friedler, and L. T. Fan, “Extending process-network synthesis algorithms with time bounds for supply network design,” *Chemical Engineering Transactions*, vol. 29, pp. 259–264, 2012.

[17] M. Frits and B. Bertok, “Process scheduling by synthesizing time constrained process-networks,” *Computer Aided Chemical Engineering*, vol. 33, pp. 1345–1350, 2014.

[18] 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*, vol. 13, no. 4, pp. 637–642, 2011.

[19] A. Szlama, I. Heckl, and H. Cabezas, “Optimal design of renewable energy systems with flexible inputs and outputs using the p-graph framework,” *AIChE Journal*, vol. 62, pp. 1143–1153, 2016.

[20] K. B. Aviso, K. D. Yu, and R. R. Tan, “P-graph model of economic networks with partial substitution,” *Chemical Engineering Transactions*, vol. 88, pp. 61–66, 2021.

[21] I. Heckl, L. Halász, A. Szlama, H. Cabezas, and F. Friedler, “Modeling multi-period operations using the p-graph methodology,” *Computer Aided Chemical Engineering*, vol. 33, pp. 979–984, 2014.

[22] R. R. Tan and K. B. Aviso, “An extended p-graph approach to process network synthesis for multi-period operations,” *Computers & Chemical Engineering*, vol. 85, pp. 40–42, 2016.

[23] B. Bertók and A. Bartos, “Algorithmic process synthesis and optimisation for multiple time periods including waste treatment: Latest developments in p-graph studio software,” *Chemical Engineering Transactions*, vol. 70, pp. 97–102, 2018.

[24] F. Friedler, Á. Orosz, and J. P. Losada, *P-graphs for Process Systems Engineering*, 1st ed. Springer, Cham, 2022.

[25] H. L. Lam, “Extended p-graph applications in supply chain and process network synthesis,” *Current Opinion in Chemical Engineering*, vol. 4, no. 2, pp. 475–486, 2013.

[26] H. Cabezas, A. Argoti, F. Friedler, P. Mizsey, and J. Pimentel, “Design and engineering of sustainable process systems and supply chains by the p-graph framework,” *Environmental Progress & Sustainable Energy*, vol. 37, pp. 624–636, 2018.

[27] M. Ebrahim and K. Al, “Pinch technology: An efficient tool for chemical-plant energy and capital-cost saving,” *Applied Energy*, vol. 65, no. 1, pp. 45–49, 2000.

[28] R. R. Tan and D. C. Y. Foo, “Pinch analysis approach to carbon-constrained energy sector planning,” *Energy*, vol. 32, no. 8, pp. 1422–1429, 2007.

[29] R. R. Tan, K. B. Aviso, and D. C. Y. Foo, “P-graph approach to carbon-constrained energy planning problems,” in *26th European Symposium on Computer Aided Process Engineering*, ser. Computer Aided Chemical Engineering, Z. Kravanja and M. Bogataj, Eds., vol. 38, Elsevier, 2016, pp. 2385–2390.

[30] B. S. How, B. H. Hong, H. L. Lam, and F. Friedler, “Synthesis of multiple biomass corridor via decomposition approach: A p-graph application,” *Journal of Cleaner Production*, vol. 130, pp. 45–57, 2016, Special Volume: SDEWES 2014 - Sustainable Development of Energy, Water and Environment Systems.

[31] M. J. Atkins, T. G. Walmsley, B. H. Y. Ong, M. R. W. Walmsley, and J. R. Neale, “Application of p-graph techniques for efficient use of wood processing residues in biorefineries,” *Chemical Engineering Transactions*, vol. 52, pp. 499–504, 2016.

[32] U. Safder, J. Y. Lim, B. S. How, P. Ifaei, S. Heo, and C. Yoo, “Optimal configuration and economic analysis of pro-retrofitted industrial networks for sustainable energy production and material recovery considering uncertainties: Bioethanol and sugar mill case study,” *Renewable Energy*, vol. 182, pp. 797–816, 2022.

[33] H. L. Lam, P. S. Varbanov, and J. J. Klemeš, “Optimisation of regional energy supply chains utilising renewables: P-graph approach,” *Computers & Chemical Engineering*, vol. 34, no. 5, pp. 782–792, 2010, Selected Paper of Symposium ESCAPE 19, June 14-17, 2009, Krakow, Poland.

[34] H. Cabezas, I. Heckl, B. Bertok, and F. Friedler, “Use the p-graph framework to design supply chains for sustainability,” *Chemical Engineering Progress*, vol. 111, no. 1, pp. 41–47, 2015.

[35] L. Vance, I. Heckl, B. Bertok, H. Cabezas, and F. Friedler, “Designing sustainable energy supply chains by the p-graph method for minimal cost, environmental burden, energy resources input,” *Journal of Cleaner Production*, vol. 94, pp. 144–154, 2015.

[36] H. L. Lam, K. H. Chong, T. K. Tan, G. D. Ponniah, Y. T. Tin, and B. S. How, “Debottlenecking of the integrated biomass network with sustainability index,” *Chemical Engineering Transactions*, vol. 61, pp. 1615–1620, 2017.

[37] B. S. How, T. T. Yeoh, T. K. Tan, K. H. Chong, D. Ganga, and H. L. Lam, “Debottlenecking of sustainability performance for integrated biomass supply chain: P-graph approach,” *Journal of Cleaner Production*, vol. 193, pp. 720–733, 2018.

[38] N. Niemetz, K.-H. Kettl, M. Szerencsits, and M. Narodoslawsky, “Economic and ecological potential assessment for biogas production based on intercrops,” English, in *Biogas*, 1st ed. Croatia: InTech, 2012, pp. 173–190.

[39] I. Heckl, F. Friedler, and L. T. Fan, “Solution of separation-network synthesis problems by the p-graph methodology,” *Computers & Chemical Engineering*, vol. 34, no. 5, pp. 700–706, 2010, Selected Paper of Symposium ESCAPE 19, June 14-17, 2009, Krakow, Poland.

[40] A. Bartos and B. Bertok, “Production line balancing by p-graphs,” *Optimization and Engineering*, vol. 21, pp. 567–584, 2020.

[41] K. B. Aviso, A. S. F. Chiu, F. P. A. Demeterio, R. I. G. Lucas, M.-L. Tseng, and R. R. Tan, “Optimal human resource planning with p-graph for universities undergoing transition,” *Journal of Cleaner Production*, vol. 224, pp. 811–822, 2019.

[42] J. S. Okusa, J. C. R. Dulatre, V. R. F. Madria, K. B. Aviso, and R. R. Tan, “P-graph approach to optimization of polygeneration systems under uncertainty,” in *Proceedings of the DLSU Research Congress*, Manila, Philippines, March 7-9. Vol. 4, 2016.

[43] R. R. Tan, C. D. Cayamanda, and K. B. Aviso, “P-graph approach to optimal operational adjustment in polygeneration plants under conditions of process inoperability,” *Applied Energy*, vol. 135, pp. 402–406, 2014.

[44] A. Orosz, Z. Kovacs, and F. Friedler, "Processing systems synthesis with embedded reliability consideration," in *28th European Symposium on Computer Aided Process Engineering*, ser. Computer Aided Chemical Engineering, A. Friedl, J. J. Klemeš, S. Radl, P. S. Varbanov, and T. Wallek, Eds., vol. 43, Elsevier, 2018, pp. 869–874.

[45] Z. Süle, J. Baumgartner, G. Dörgő, and J. Abonyi, "P-graph-based multi-objective risk analysis and redundancy allocation in safety-critical energy systems," *Energy*, vol. 179, pp. 989–1003, 2019.

[46] M. F. D. Benjamin, "Multi-disruption criticality analysis in bioenergy-based eco-industrial parks via the p-graph approach," *Journal of Cleaner Production*, vol. 186, pp. 325–334, 2018.

[47] M. F. Benjamin, K. Aviso, B. A. Belmonte, and R. R. Tan, "Optimal operations of a bioenergy park under capacity disruptions via the p-graph method," *Chemical Engineering Transactions*, vol. 89, pp. 139–144, 2021.

[48] R. T. L. Ng, R. R. Tan, and M. H. Hassim, "P-graph methodology for bi-objective optimisation of bioenergy supply chains: Economic and safety perspectives," *Chemical Engineering Transactions*, vol. 45, pp. 1357–1362, 2015.

[49] K. B. Aviso and R. R. Tan, "Fuzzy p-graph for optimal synthesis of cogeneration and trigeneration systems," *Energy*, vol. 154, pp. 258–268, 2018.

[50] K. B. Aviso, B. A. Belmonte, M. F. D. Benjamin, *et al.*, "Synthesis of optimal and near-optimal biochar-based carbon management networks with p-graph," *Journal of Cleaner Production*, vol. 214, pp. 893–901, 2019.

[51] R. R. Tan, K. B. Aviso, and D. C. Y. Foo, "P-graph and monte carlo simulation approach to planning carbon management networks," *Computers & Chemical Engineering*, vol. 106, pp. 872–882, 2017, ESCAPE-26.

[52] I. Heckl, L. Halász, A. Szlama, H. Cabezas, and F. Friedler, "Process synthesis involving multi-period operations by the p-graph framework," *Computers & Chemical Engineering*, vol. 83, pp. 157–164, 2015.

[53] K. B. Aviso, J.-Y. Lee, J. C. Dulatre, V. R. Madria, J. Okusa, and R. R. Tan, "A p-graph model for multi-period optimization of sustainable energy systems," *Journal of Cleaner Production*, vol. 161, pp. 1338–1351, 2017.

[54] I. H. Osman and C. N. Potts, "Simulated annealing for permutation flow-shop scheduling," *Omega*, vol. 17, no. 6, pp. 551–557, 1989.

[55] Y. Li and Y. Chen, "A genetic algorithm for job-shop scheduling," *Journal of Software*, vol. 5, no. 3, pp. 269–274, 2010.

[56] W. H. M. Raaymakers and J. A. Hoogeveen, "Scheduling multipurpose batch process industries with no-wait restrictions by simulated annealing," *European Journal of Operational Research*, vol. 126, no. 1, pp. 131–151, 2000.

[57] C. Bierwirth and D. C. Mattfeld, “Production scheduling and rescheduling with genetic algorithms,” *Evolutionary Computation*, vol. 7, no. 1, pp. 1–17, 1999.

[58] A. Sadegheih, “Scheduling problem using genetic algorithm, simulated annealing and the effects of parameter values on {ga} performance,” *Applied Mathematical Modelling*, vol. 30, no. 2, pp. 147–154, 2006.

[59] E. Sanmartí, L. Puigjaner, T. Holczinger, and F. Friedler, “Combinatorial framework for effective scheduling of multipurpose batch plants,” *AIChE Journal*, vol. 48, no. 11, pp. 2557–2570, 2002.

[60] J. Romero, L. Puigjaner, T. Holczinger, and F. Friedler, “Scheduling intermediate storage multipurpose batch plants using the s-graph,” *AIChE Journal*, vol. 50, no. 2, pp. 403–417, 2004.

[61] M. Hegyhati, T. Holczinger, A. Szoldatics, and F. Friedler, “Combinatorial approach to address batch scheduling problems with limited storage time,” *Chemical Engineering Transactions*, vol. 25, pp. 495–500, 2011.

[62] M. Hegyháti and F. Friedler, “Combinatorial algorithms of the s-graph framework for batch scheduling,” *Industrial & Engineering Chemistry Research*, vol. 50, no. 9, pp. 5169–5174, 2011.

[63] M. Hegyháti, T. Majozi, T. Holczinger, and F. Friedler, “Practical infeasibility of cross-transfer in batch plants with complex recipes: S-graph vs milp methods,” *Chemical Engineering Science*, vol. 64, no. 3, pp. 605–610, 2009.

[64] J. M. Laínez, M. Hegyháti, F. Friedler, and L. Puigjaner, “Using s-graph to address uncertainty in batch plants,” *Clean Technologies and Environmental Policy*, vol. 12, no. 2, pp. 105–115, 2010.

[65] J. M. Laínez-Aguirre and L. Puigjaner, “Using s-graph to address exogenous uncertainty in processes scheduling,” in *Advances in Integrated and Sustainable Supply Chain Planning: Concepts, Methods, Tools and Solution Approaches toward a Platform for Industrial Practice*. Cham: Springer International Publishing, 2015, pp. 197–214.

[66] T. Holczinger, O. Ősz, and M. Hegyháti, “Scheduling approach for on-site jobs of service providers,” *Flexible Services and Manufacturing Journal*, vol. 32, no. 4, pp. 913–948, 2020.

[67] C. A. Méndez, J. Cerdá, I. E. Grossmann, I. Harjunkoski, and M. Fahl, “State-of-the-art review of optimization methods for short-term scheduling of batch processes,” *Computers & Chemical Engineering*, vol. 30, no. 6, pp. 913–946, 2006.

[68] N. V. Sahinidis and I. E. Grossmann, “Reformulation of multiperiod milp models for planning and scheduling of chemical processes,” *Computers & Chemical Engineering*, vol. 15, no. 4, pp. 255–272, 1991.

[69] B. Brunaud, H. D. Perez, S. Amaran, S. Bury, J. Wassick, and I. E. Grossmann, “Batch scheduling with quality-based changeovers,” *Computers & Chemical Engineering*, vol. 132, p. 106 617, 2020.

[70] E. Kondili, C. C. Pantelides, and R. W. H. Sargent, “A general algorithm for short-term scheduling of batch operations–i. milp formulation,” *Computers & Chemical Engineering*, vol. 17, no. 2, pp. 211–227, 1993.

[71] C. T. Maravelias and I. E. Grossmann, “New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants,” *Industrial & Engineering Chemistry Research*, vol. 42, no. 13, pp. 3056–3074, 2003.

[72] M. G. Ierapetritou and C. A. Floudas, “Effective continuous-time formulation for short-term scheduling. 1. multipurpose batch processes,” *Industrial & Engineering Chemistry Research*, vol. 37, no. 11, pp. 4341–4359, 1998.

[73] J. M. Pinto and I. E. Grossmann, “A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants,” *Industrial & Engineering Chemistry Research*, vol. 34, no. 9, pp. 3037–3051, 1995.

[74] C. A. Mendez and J. Cerda, “An milp continuous-time framework for short-term scheduling of multipurpose batch processes under different operation strategies,” *Optimization and Engineering*, vol. 4, no. 1-2, pp. 7–22, 2003.

[75] S. B. Kim, H. K. Lee, I. B. Lee, E. S. Lee, and B. Lee, “Scheduling of non-sequential multipurpose batch processes under finite intermediate storage policy,” *Computers & Chemical Engineering*, vol. 24, no. 2-7, pp. 1603–1610, 2000.

[76] Z. Bradac, V. Kaczmarczyk, and P. Fiedler, “Optimal scheduling of domestic appliances via milp,” *Energies*, vol. 8, no. 1, pp. 217–232, 2015.

[77] T. Vidal, G. Laporte, and P. Matl, “A concise guide to existing and emerging vehicle routing problem variants,” *European Journal of Operational Research*, vol. 286, no. 2, pp. 401–416, 2020.

[78] R. Moghdani, K. Salimifard, E. Demir, and A. Benyettou, “The green vehicle routing problem: A systematic literature review,” *Journal of Cleaner Production*, vol. 279, p. 123 691, 2021.

[79] R. V. Kulkarni and P. R. Bhave, “Integer programming formulations of vehicle routing problems,” *European Journal of Operational Research*, vol. 20, no. 1, pp. 58–67, 1985.

[80] L. Costa, C. Contardo, and G. Desaulniers, “Exact branch-price-and-cut algorithms for vehicle routing,” *Transportation Science*, vol. 53, no. 4, pp. 946–985, 2019.

[81] V. Maniezzo, M. A. Boschetti, and T. Stützle, *Matheuristics*, M. G. Speranza and J. F. Oliveira, Eds., ser. EURO Advanced Tutorials on Operational Research. Springer, Cham, 2021.

[82] J. D. Camm, M. J. Magazine, S. Kuppusamy, and K. Martin, "The demand weighted vehicle routing problem," *European Journal of Operational Research*, vol. 262, no. 1, pp. 151–162, 2017.

[83] D. M. Chitty and M. L. Hernandez, "A hybrid ant colony optimisation technique for dynamic vehicle routing," in *Genetic and Evolutionary Computation – GECCO 2004*, K. Deb, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 48–59.

[84] H.-K. Chen, C.-F. Hsueh, and M.-S. Chang, "Production scheduling and vehicle routing with time windows for perishable food products," *Computers & Operations Research*, vol. 36, no. 7, pp. 2311–2319, 2009.

[85] H. N. Geismar, G. Laporte, L. Lei, and C. Sriskandarajah, "The integrated production and transportation scheduling problem for a product with a short lifespan," *INFORMS Journal on Computing*, vol. 20, no. 1, pp. 21–33, 2008.

[86] Y. Kergosien, M. Gendreau, and J.-C. Billaut, "A benders decomposition-based heuristic for a production and outbound distribution scheduling problem with strict delivery constraints," *European Journal of Operational Research*, vol. 262, no. 1, pp. 287–298, 2017.

[87] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik*, vol. 4, no. 1, pp. 238–252, 1962.

[88] J. Lee, B.-I. Kim, A. L. Johnson, and K. Lee, "The nuclear medicine production and delivery problem," *European Journal of Operational Research*, vol. 236, no. 2, pp. 461–472, 2014.

[89] F. Ben Abdelaziz, H. Masri, and H. Alaya, "A recourse goal programming approach for airport bus routing problem," *Annals of Operations Research*, vol. 251, no. 1, pp. 383–396, 2017.

[90] Y. Gong, J. Zhang, O. Liu, R. Huang, H. S. Chung, and Y. Shi, "Optimizing the vehicle routing problem with time windows: A discrete particle swarm optimization approach," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 2, pp. 254–267, 2012.

[91] L. Liu, K. Li, and Z. Liu, "A capacitated vehicle routing problem with order available time in e-commerce industry," *Engineering Optimization*, vol. 49, no. 3, pp. 449–465, 2017.

[92] G. Macrina, G. Laporte, F. Guerriero, and L. Di Puglia Pugliese, "An energy-efficient green-vehicle routing problem with mixed vehicle fleet, partial battery recharging and time windows," *European Journal of Operational Research*, vol. 276, no. 3, pp. 971–982, 2019.

[93] J. Paz, M. Granada-Echeverri, and J. W. Escobar, "The multi-depot electric vehicle location routing problem with time windows," *International Journal of Industrial Engineering Computations*, vol. 9, pp. 123–136, 2018.

[94] X. Wang, S. Poikonen, and B. Golden, "The vehicle routing problem with drones: Several worst-case results," *Optimization Letters*, vol. 11, no. 4, pp. 679–697, 2017.

[95] Z. Wang and J.-B. Sheu, "Vehicle routing problem with drones," *Transportation Research Part B: Methodological*, vol. 122, pp. 350–364, 2019.

[96] S. Pelletier, O. Jabali, and G. Laporte, "The electric vehicle routing problem with energy consumption uncertainty," *Transportation Research Part B: Methodological*, vol. 126, pp. 225–255, 2019.

[97] J. A. Castillo-Salazar, D. Landa-Silva, and R. Qu, "Workforce scheduling and routing problems: Literature survey and computational study," *Annals of Operations Research*, vol. 239, no. 1, pp. 39–67, 2016.

[98] J. Sung and B. Jeong, "An adaptive evolutionary algorithm for traveling salesman problem with precedence constraints," *The Scientific World Journal*, vol. 2014, Z. Cui and X. Yang, Eds., p. 313 767, 2014.

[99] A. Goel and F. Meisel, "Workforce routing and scheduling for electricity network maintenance with downtime minimization," *European Journal of Operational Research*, vol. 231, no. 1, pp. 210–228, 2013.

[100] D. L. Pereira, J. C. Alves, and M. C. de Oliveira Moreira, "A multiperiod workforce scheduling and routing problem with dependent tasks," *Computers & Operations Research*, vol. 118, p. 104 930, 2020.

[101] N. Safaei, D. Banjevic, and A. K. S. Jardine, "Simulated annealing," in C. M. Tan, Ed. InTech, 2008, ch. Multi-Objective Simulated Annealing for a Maintenance Workforce Scheduling Problem: A Case Study, pp. 27–48.

[102] A. Goel, V. Gruhn, and T. Richter, "Mobile workforce scheduling problem with multitask-processes," in *Business Process Management Workshops*, F. L. Stefanie Rinderle-M Shazia Sadiq, Ed., Ulm, Germany: Springer, 2009, pp. 81–91.

[103] A. Starkey, H. Hagras, S. Shakya, and G. Owusu, "A multi-objective genetic type-2 fuzzy logic based system for mobile field workforce area optimization," *Information Sciences*, vol. 329, pp. 390–411, 2016.

[104] R. Chimatapu, H. Hagras, A. Starkey, and G. Owusu, "A big-bang big-crunch type-2 fuzzy logic system for generating interpretable models in workforce optimization," in *2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Rio de Janeiro, Brazil: IEEE, 2018, pp. 1–8.

[105] A. J. Starkey, H. Hagras, S. Shakya, and G. Owusu, "A genetic algorithm based system for simultaneous optimisation of workforce skills and teams," *KI - Künstliche Intelligenz*, vol. 32, no. 4, pp. 245–260, 2018.

[106] S. Çakırgil, E. Yücel, and G. Kuyzu, "An integrated solution approach for multi-objective, multi-skill workforce scheduling and routing problems," *Computers & Operations Research*, vol. 118, p. 104 908, 2020.

[107] A. Nemet, J. J. Klemeš, N. Duić, and J. Yan, "Improving sustainability development in energy planning and optimisation," *Applied Energy*, vol. 184, pp. 1241–1245, 2016.

[108] M. R. Saavedra, C. H. O. Fontes, and F. G. M. Freires, "Sustainable and renewable energy supply chain: A system dynamics overview," *Renewable & Sustainable Energy Reviews*, vol. 82, pp. 247–259, 2018.

[109] M. A. Kalaitzidou, M. C. Georgiadis, and G. M. Kopanos, "A general representation for the modeling of energy supply chains," in *26th European Symposium on Computer Aided Process Engineering*, ser. Computer Aided Chemical Engineering, Z. Kravanja and M. Bogataj, Eds., vol. 38, Elsevier, 2016, pp. 781–786.

[110] B. Sharma, R. G. Ingalls, C. L. Jones, and A. Khanchi, "Biomass supply chain design and analysis: Basis, overview, modeling, challenges, and future," *Renewable and Sustainable Energy Reviews*, vol. 24, pp. 608–627, 2013.

[111] "Supplementary materials for the energy consumption optimization case study, 2019," [Online]. Available at: `https://dcs.uni-pannon.hu/files/docs/users/eles/downloads/2019-energy-consumption-case-study-supplementary.7z` Accessed: 03/03/2022.

[112] "Supplementary materials for the mobile workforce management model, 2020," [Online]. Available at: `https://dcs.uni-pannon.hu/files/docs/users/eles/downloads/MWM-2020-supplementary.7z` Accessed: 07/07/2022.

[113] "Supplementary materials for the rural biomass supply chain case study, 2021," [Online]. Available at: `https://dcs.uni-pannon.hu/files/docs/users/eles/downloads/PGraph-MILP-CaseStudy-BadZell-2021.7z` Accessed: 05/05/2022.

Appendix A

Case study for fermenters with flexible inputs

This appendix chapter presents further details of the case study summarized in Section 4.4, and serves as a demonstration for the modeling technique.

The basis is the work published by Niemetz et al. [38] about a sustainable and economical utilization of locally available biomass, near the town of Bad Zell. The authors presented an optimization problem which was formulated as a PNS problem. The problem data and optimal solutions in different scenarios were detailed in the publication and a corresponding technical report.

The case study involved fermenter units which produce biogas from different types of biomass. The published solution relied on a model of fermenters which assumed a fixed input composition. Specifically, there were 8 available mixtures, each determining a fresh matter input ratio for a fermenter. For example, the $50 : 20 : 10 : 20$ mixture meant that 50% of fresh matter input was manure, 20% was intercrops, 10% was grass and 20% was corn silage. The optimization assumed that any fermenter to be built would use one of the given mixtures.

The question was that if we change the fixed input model of fermenters to a flexible input model, how the overall performance of the whole model is affected.

A.1 Problem description

The most important components of the model are shown in Figure A1. The material flow is the following.

- Four different types of biomass are produced by local agriculture: manure, intercrops, grass and corn silage. Each biomass type has a purchase price per unit of fresh matter. These are available for use from 8 different supplier sites.
- Transportation must be arranged for biomass to any of 3 possible processing locations: L_1, L_2 and L_3.

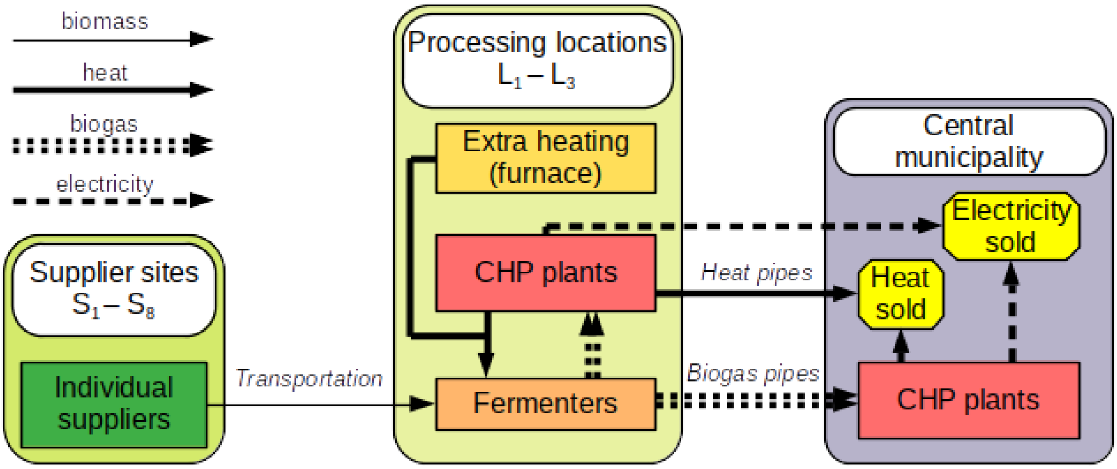


Figure A1: Main components and material flows in the Bad Zell case study.

- At each processing location, multiple fermenters can be built, in sizes 80 kW, 160 kW, 250 kW and 500 kWh. Fermenters require heating, and produce biogas based on input amounts.

- At each processing location and the central town location, CHP plants can be built in the same sizes as fermenters, which produce electricity and heating.

- Supporting infrastructure must be built depending on investment choices: transformer, silo plates, biogas pipes and heat pipes.

- Electricity and heating is sold, assuming no upper bound.

The objective is maximization of yearly profit. The key decisions to be made are the exact amount of biomass to purchase and where to transport it, where to build fermenters and CHP units, and in what sizes and input compositions.

The only loop in the material flow is due to the fact that fermenter heating can partially come from CHP plants. Otherwise, heating of fermenters is done by traditional wood chip furnaces for which only a material cost is assumed.

For a CHP plant, the size means electricity output. For a fermenter, the size means that assuming the total 7800 working hours for the year, it can just feed a CHP plant of the same size with full capacity. For example, an 80 kW fermenter can produce just enough biogas for an 80 kW CHP plant if both operate at full capacity. Note that this connection of fermenters and CHP plants is not at all mandatory, any combinations and sizing of fermenters and CHP plants can coexist provided that there is sufficient input. It is also possible to use a fermenter or a CHP plant below full capacity, although it is discouraged due to investment costs.

There can be different feed-in tariffs for different CHP sizes. In particular, 185 EUR/MWh for the 500 kWh CHP plant, but 205 EUR/MWh for smaller sizes. This may contradict economies of scale. Heat sell price was assumed to be 22.5 EUR/MWh.

The supporting infrastructure must be built in the following conditions.

- A transformer must be built if we want to sell electricity.
- A silo plate must be built per processing location if a fermenter is built there.
- A biogas pipe must be built between a processing location and the central town, if the biogas produced by a fermenter at the processing location is used up by a CHP plant at the central town. Note that only CHP plants are allowed at the central town, fermenters are not.
- A heat pipe must be built between a processing location and the central town, if heating produced by a CHP plant at a processing location is to be sold.

Pipes are not independent: a pipe from the processing location L_3 to the center consist of the pipe from L_3 to L_1, and from L_1 to the center. Therefore, the pipe segment for L_1 is used for transportation from both L_1 and L_3.

Investment costs are associated with fermenters, CHP plants, biogas and heat pipes, silo plates and the transformer. A 15 years long payoff period is assumed for investments. Yearly operational costs are associated with fermenters, CHP plants and silo plates, regardless of usage. Transportation costs have a fixed and a proportional part for distance, for each unit of biomass transported. For heat transfer between sites, a fixed heat loss and an operation cost proportional to the transferred heat are also assumed.

The key point of importance was the choices for fermenters. In the report, 8 fixed input material compositions were considered for the fermenters. These are shown with total raw material availability in Table A1. Any of these 8 fermenters prescribed a fixed input ratio for the four biomass types. Parameters of the fermenters were calculated based on the input composition and the size, resulting in $8 \cdot 4 = 32$ designs in total. Therefore, at each of the three processing locations, any number of fermenters of any of these 32 designs can be built. Note that for CHP plants, there is no input composition distinction, simply any number of CHP plants in any of the four sizes can be built at any of the three processing locations, or in the central town.

There were also input composition restrictions for fermenters. These are implicitly ensured by fixed compositions, but for a model with flexible input compositions, such constraints must be explicitly stated. The assumed constraint was that the composition of manure is at least 30% for each fermenter.

A.2 Model formulations

This case study consisted of the following four phases, to be explained afterwards.

Table A1: The 8 fixed fermenter input compositions considered in the original PNS formulation. For each raw material (biomass) type considered in the model, the total availability is also indicated.

Biomass	Available	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
Manure	15,501 m^3	30%	30%	50%	50%	75%	75%	75%	100%
Intercrops	5,300 t		70%	50%	20%		25%	15%	
Grass	2,820 t				10%			10%	
Corn	2,418 t	70%			20%	25%			

1. Reproduction of the original results, as a MILP model using the fixed input model for fermenters.

2. Calculation of linear estimations of fermenter parameters required by modeling flexible inputs for fermenters.

3. Formulation of the MILP model using flexible inputs for fermenters.

4. Formulation of the PNS model using flexible inputs for fermenters.

The original publication provided data for the problems and the optimal solutions, even in different scenarios. However, the model formulations were unavailable. Therefore, the first phase was the reproduction of the original results.

The reproduction was performed as an MILP model instead of a PNS problem. The motivation of this choice was that the GNU MathProg modeling language had some valuable features for debugging purposes, which was extensively needed by the reproduction process. Also, the goal was to provide a supportive basis for comparison with the PNS problem formulation afterwards.

The optimal solution reported by the MILP model using fixed inputs for fermenters is summarized as follows.

- One 160 kW and one 250 kW CHP plant is built in the central area and one 80 kW plant at processing location L_1. All CHP plants work at full capacity.

- Two 250 kW fermenters are built at L_1, using mixture 4 ($50 : 20 : 10 : 20$) and mixture 7 ($75 : 15 : 10 : 0$). The ratios indicate fresh matter input ratios for the fermenters. Both fermenters work at roughly 98% capacity. A biogas pipe is installed from L_1 to the central town.

- Revenue is 783,510 EUR/y from electricity and 93,015 EUR/y from heating.

- Operating costs are 460,928 EUR/y, investment costs are 2,715,790 EUR, resulting in a total profit of 234,544 EUR/y.

- 100% of manure, 75% of intercrops, 84% of grass and 74% of corn silage was utilized from the total availability. 5 of the 8 main supplier sites contributed with their full availability. Note that this part of a solution can easily be balanced in exchange for additional transportation costs.

It must be noted that there is a difference between this result and the one reported by the original study, where the reported optimal solution was 196,351 EUR/y. Although both solutions use the same fermenter compositions and sizes, they are at different sites L_1 and L_3 in the original reported solution, which also uses two 250 kW CHP plants instead. The main reason behind this discrepancy is probably due to the calculation of transportation costs, which are roughly 24,000 EUR/y more in the original formulation, and were not modeled in the same way in the reproduction.

In other terms, the model reproduction seems accurate. For this reason, this MILP model reproduction from the first phase was used as a basis for comparison with the further MILP and PNS model formulations.

A.3 Parameter estimation

There are two fermenter parameters which are relevant in the modeling point of view and depend on input composition: heating requirement and investment cost. Therefore, fermenter heating and investment costs should appear as variable quantities in the decision problem, ensuring their correct values by constraints. The problem is that the exact requirements are not known for an arbitrary fermenter input composition, only for the given 8 fixed designs. Besides, the calculations resulting in these data were unavailable.

To overcome this issue, the second phase was an estimation for both the heating requirement and the investment cost of the fermenters, as a function of all four input material amounts. The estimation was also required to be a linear function of these amounts, because both the MILP modeling technique and PNS problems with the current ABB algorithm implementation are capable of incorporating only linear components. For this reason, a multiple linear regression was performed, based on the data available for the 32 fixed fermenter designs, as follows. The obtained coefficients are shown in Table A2.

- Investment costs are significantly different due to economies of scale: larger sized fermenters are cheaper. For this reason, the regression was performed for each of the four sizes independently.

- Heating requirements on the other hand seemed to depend linearly on sizing, therefore a single regression was performed for all sizes.

As a result, the heating requirement and the investment cost of the fermenter were expressed as a linear function of input material amounts, without a constant coefficient. Note that the estimations assume a 100% working capacity of fermenters.

Table A2: Coefficients for input biomass amounts to determine fermenter parameters, obtained from multiple linear regression. Coefficients are interpreted as **per unit**, where unit is m^3 for manure and t for the other types of biomass. Heating requirement is expressed in terms of fermenter size in kW.

Biomass	Heating (MW/kW)	Investment cost (EUR)			
		80 kW	160 kW	250 kW	500 kW
Manure	0.041	59.30	55.95	47.67	47.19
Intercrops	0.035	187.12	152.03	122.09	103.17
Grass	0.016	246.84	196.98	153.48	88.59
Corn	0.042	267.47	210.54	178.78	134.87

Table A3: Comparison of the original and estimated problem data, showing the effect of using the linear parameter estimations.

	Data	Fermenter heating	Total investments	Objective
1.	Original	70.36 MWh	2,715,790 EUR	234,544 EUR/y
2.	Estimated	71.81 MWh	2,737,360 EUR	233,033 EUR/y

The estimation data compared to the original 32 fixed input fermenter designs varied by between -7.8% and $+6.5\%$.

The MILP model using fixed inputs for fermenters was also solved with the estimated data, and the resulting optimal solution involved exactly the same investment decisions and transportation amounts. The only differences are due to the changed investment costs and heating requirements, both reflected in the objective only. These are shown in Table A3.

Overall, the linear parameters estimations are accurate for further use.

A.4 MILP model using flexible inputs

In the third phase, the MILP model using flexible inputs for fermenters was developed, and directly compared with the MILP model using fixed inputs. Note that while the MILP model using fixed inputs can be used with the original and estimated data, the MILP model using flexible inputs can only be used with the estimated data. Some important details of the two models are shown below.

In the MILP model using fixed inputs, key decisions about the fermenters are represented by integer variables $u^{ferm}_{k,m,l} \geq 0$, where the index k is for fermenter size, m is for the selected input composition, and l is the processing location. Variable $u^{ferm}_{k,m,l}$ denotes the number of fermenters of the kind described by the indices. This is an integer variable, therefore multiple fermenters of the same kind are allowed, the upper bound was 3 per kind.

In the MILP model with flexible fermenters, a binary variable $v^{ferm}_{k,i,l}$ is used instead. Here, k is fermenter size and l is processing location. The index $i \in I$ is introduced to allow multiple fermenters of the same kind, precisely at most $|I|$. In the model, $I = \{1, 2\}$ was chosen, therefore allowing two identical fermenters per kind. This choice eventually turned out to be sufficient.

Comparing the number of the main decision variables, the index m for $u^{ferm}_{k,m,l}$ introduces a factor of 8 which is the number of fixed input compositions, while the index i for $v^{ferm}_{k,i,l}$ introduces a factor of 2, which can be arbitrarily chosen. Therefore, the flexible model has significantly fewer decision variables.

In the flexible model, heating requirements $h^{req}_{k,i,l}$ for a particular fermenter are calculated based on its total input amounts $w^{in}_{k,i,l,t}$ with the use of coefficient $C^{heat,req}_{k,t}$ obtained from the estimation, where the index t is for the biomass type (see Table A2). This calculation is shown in Equation (A1).

$$h^{req}_{k,i,l} = \sum_t C^{heat,req}_{k,t} \cdot w^{in}_{k,i,l,t} \quad \forall\ k, i, l \tag{A1}$$

Investment costs of fermenters are more problematic, because those should not be scaled down if the fermenter is not working at full capacity. Otherwise, economies of scale could lead to, for example, using a 160 kW fermenter at 50% capacity instead of a 80 kW fermenter at full capacity, which is clearly unwanted.

Modeling an unscalable but input-dependent investment cost precisely would require nonlinear constraints. As a workaround, an upper bound for the investment cost is used instead, which is calculated as follows. The most expensive biomass type in terms of investment cost per biogas produced is manure, due to its low dry matter content. The workaround is the following: the fermenter is assumed to be working at full capacity, but a fifth virtual input material is introduced with the following properties.

- The sum of input amounts of the five inputs is exactly the 100% capacity of the fermenter.
- The fifth virtual input contributes to the investment cost of the particular fermenter, as if it was substituted by manure.
- The fifth virtual input does not contribute to any other material flows and calculations in the model, nor to operating costs and heating requirements. In other words, the variable denoting its amount is used as a slack variable.

Due to the fact that the most expensive biomass type is used as a slack, which is manure, the following can be stated about the calculation used in the MILP model versus the original one obtained by the linear estimation.

- If the fermenter with flexible inputs uses only manure, then the investment cost calculation is exact.
- If the fermenter operates at 100% capacity, there is no slack capacity, therefore the investment cost calculation is also exact.

- If the fermenter operates below 100% capacity and biomass types other than manure are also used, then the investment cost estimates the original value from above.

The constraint for the connection of the slack variable with the total input amount is shown in Equation (A2). Here, M_k^{ferm,CH_4} denotes the total production capacity of a fermenter with size k, regardless of input composition, and $\lambda_t^{CH_4}$ is a conversion factor.

$$M_k^{ferm,CH_4} \cdot v_{k,i,l}^{ferm} = \lambda_{\text{Manure}}^{CH_4} \cdot w_{k,i,l}^{slack} + \sum_t \lambda_t^{CH_4} \cdot w_{k,i,l,t}^{in} \quad \forall\ k, i, l \tag{A2}$$

The investment cost calculation for fermenters is expressed in Equation (A3). Here, $c_{k,i,l}^{ferm,inv}$ is the variable introduced for the investment cost, and $C_{k,t}^{inv}$ is the coefficient obtained from the linear estimation (see Table A2).

$$c_{k,i,l}^{ferm,inv} = C_{k,\text{Manure}}^{inv} \cdot w_{k,i,l}^{slack} + \sum_t C_{k,t}^{inv} \cdot w_{k,i,l,t}^{in} \quad \forall\ k, i, l \tag{A3}$$

There was also a regulation in the original case study: manure composition was required to be at least 30%. This is implicitly satisfied if the fixed input compositions are used, because each design uses at least 30% manure. In the flexible input model, this constraint must further be explicitly stated, see Inequality (A4).

$$w_{k,i,l,\text{Manure}}^{in} \geq 0.3 \sum_t w_{k,i,l,t}^{in} \quad \forall\ k, i, l \tag{A4}$$

A.5 Results for MILP models

Both MILP models were implemented in GNU MathProg, and solved by GLPSOL v4.65 on a ThinkCentre M83 desktop PC with i7-4770 CPU and 16 GB RAM, under Ubuntu 18.04.5 LTS. Model and data files and obtained results are made publicly available [113].

The comparison of the two results shows the impact of the flexible input model on the results, see Table A4.

The solution obtained from the MILP model with flexible inputs is substantially different from the previous ones. A key difference is that a 500 kW fermenter with flexible inputs is more advantageous than two 250 kW fermenters within the fixed designs. Utilization of the available biomass is also better, and consequently, 80 kW more production capacity can be installed. This results in higher revenues, but only slightly higher investment costs due to the economies of scale. The profit, assuming the same 15 years long payoff period, is 31.62% better than for the optimal solution obtained from the fixed input model.

Another observation is that in contrast to fermenters, the 500 kW CHP plant is not used instead of the two 250 kW CHP plants. The reason behind this is the feed-in tariff being higher for the smaller plants.

Table A4: Comparison of the MILP models with fixed inputs and flexible inputs, with the data obtained from the linear parameter estimation.

Scenario	Fixed input model (2.)	Flexible input model (3.)
Decisions	two 250 kW fermenters and one 80 kW CHP plant at L_1, one 160 kW and one 250 kW CHP plant at the center	one 80 kW and one 500 kW fermenter and one 80 kW CHP plant at L_1, two 250 kW CHP plants at the center

Revenues	from electricity: 783,510 EUR/y
	from heating: 93,015 EUR/y
		from electricity: 927,420 EUR/y
		from heating: 105,300 EUR/y

Investments	2,737,360 EUR in total	2,770,220 EUR in total
Profit	233,033 EUR/y	306,711 EUR/y
Material utilization	manure: 100%, intercrops: 75%, grass: 84%, corn silage: 74%	manure, intercrops, grass: 100%, corn silage: 90%
Model size	661 columns, 128 integer variables 16 are binary, solved in 3.9 s	301 columns, 56 integer variables 40 are binary, solved in 0.5 s

The flexible input model is also smaller, with fewer variables, which results in a faster solution, 0.5 s instead of 3.9 s.

Overall, it can be concluded that the flexible input model performs better than its fixed input counterpart, in terms of solution quality and also in model complexity. A general observation is that it is more advantageous to assume equipment units with flexible inputs, optimize among them and then design the equipment units according to the solution, than first constructing some fixed input designs and letting the optimization procedure choose from those. Note that this procedure with flexible inputs is only possible if relevant model parameters can be expressed in terms of the input amounts. In this instance, it could be done with an accurate enough estimation of the heating requirement and investment cost parameters of the fermenters, using multiple linear regression. The validity of this proposed method relies on the condition of flexible input compositions obtained from model solution being feasible in reality within a close proximity of the model parameters assumed.

A.6 PNS problem formulation

Finally, the fourth phase was the formulation of the problem as a PNS problem, equivalent to the MILP model using flexible inputs for fermenters. This demonstrates the application of the modeling technique presented in Chapter 4.

Due to problem size, this PNS problem was not implemented using the GUI of P-Graph Studio, but was constructed programmatically, and solved directly by the underlying PNS solver.

The material nodes are the following:

- The raw materials of the problem are the 4 types of biomass situated at 8 different supply locations, and also a single raw material representing fermenter heating that can be purchased. These are 33 raw material nodes in total.

- There is only a single product node, for revenue. The reason for merging heating and electricity into a single node is that in theory, neither heating nor electricity is a mandatory output (although effectively they are), and there are also different selling prices for electricity from different CHP plant sizes.
- All other material nodes are intermediate materials.

The implementation of most of the material flows is straightforward, using a single operating unit node, consuming one input material node and producing one output with some conversion factor. Cases are the following.

- Transportation of biomass types to each of the processing locations.
- Purchase of fermenter heating, which contributes to a heating balance material node at that site.
- Transportation of biogas and heat from the processing locations to the central location via pipes.
- Conversion of heating balance at the central location, and electricity produced from each of the four CHP plant sizes to revenue.

Some infrastructural requirements were modeled using the same scheme as „custom input capacity”, but without an actual bound for capacity. This is illustrated in Figure A2. The infrastructure needed is represented by an operating unit node. It produces a capacity material with arbitrary flow. The capacity material is consumed by the operating unit which requires the presence of the infrastructure. This makes it possible to model dependencies of investments on each other, provided that the cost of the required investment is fixed. This technique was used for the transformer, the silo plates, biogas pipes and heat pipes.

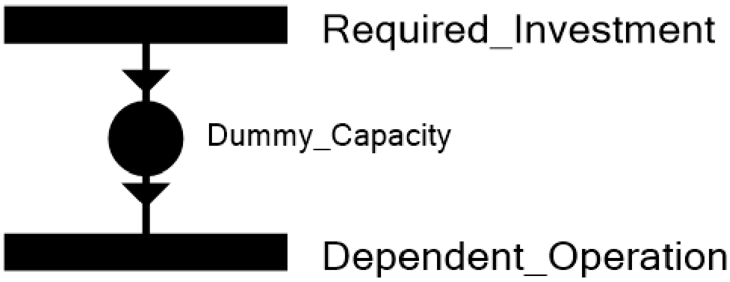


Figure A2: P-Graph model of an investment required for other operations.

The fixed heat loss model for heat pipes required some additional workaround, but this is not detailed in this work.

The most difficult part was the model of the fermenter units using flexible inputs, which is a great example for the application of ratio constraints, and lower and upper bounds for a weighted sum of inputs. This fermenter model is now shown in detail, see Figure A3.

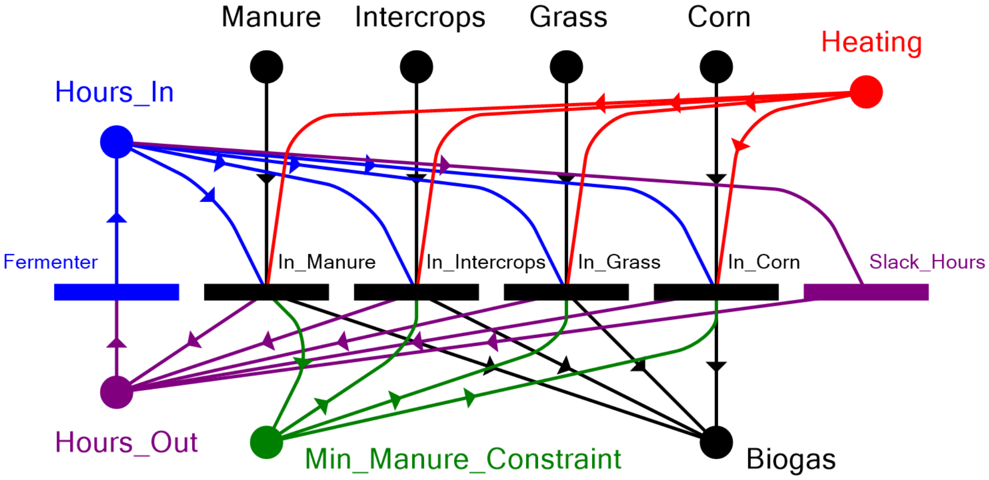


Figure A3: P-Graph model of the fermenter with flexible inputs.

The black part models the independent inputs of the fermenter, which are the four biomass types. Each input has its own operating unit node, which also play a key role in other components of the design. Flow rates for inputs are all 1, and flow rates for the output *Biogas* can be set according to the energy content of each biomass type.

The red part models fermenter heating. Since heating requirement is expressed as a linear function of biomass inputs (without a constant term), flow rates can be set according to each biomass type.

The blue part models the capacity of the fermenter. A single *Fermenter* operating unit produces the *Hours_In* material, in a maximum amount of 7800 hours, which represents total working hours during the year. Flow rates can be set according to the rule that a fully operating fermenter can exactly feed a CHP plant with the same size, regardless of input type. Note that, counter-intuitively, working hours are not really split among input types, but rather a mix of inputs is fed to the fermenter in the composition described by the consumption rate of *Hours_In*. This is only a modeling trick for implementing an upper bound for a weighted sum. Fixed costs of the fermenter are associated with the *Fermenter* node.

The purple part was necessary for the investment costs. The exact same problem described for the MILP model arises here as well, namely the fact that the investment cost cannot be scaled down if the fermenter is not used at full capacity. The same assumption was made here as well: unused fermenter capacity is assumed to be spent on manure. The *Slack_Hours* operating unit is introduced for consuming the remaining fermenter capacity, and has a proportional cost equal to that of the operating unit for manure. All of $Hours\_In$ are reproduced into $Hours\_Out$, which is a new input for the $Fermenter$ operating unit. Also, the $Fermenter$ node has a minimum production also set to 7800 hours. These ensure that investment costs are always calculated for 7800 working hours.

The green part implements the constraint that at least 30% of input fresh matter must be manure. The introduced material node is connected to the operating unit of each biomass type. The operating unit of manure produces the material with a flow rate of 7, all other operating units consume it with a flow rate of 3.

The final P-Graph consists of 147 material nodes, 319 operating unit nodes, and 1144 arcs. These required 24 distinct instances of the fermenter models, since there were 3 processing locations, 4 fermenter sizes, and two allowed per the same kind.

The solver with ABB algorithm required 413.45 s to finish, and the exact same solution was reported as for the MILP model using flexible inputs. This was done on a Dell Latitude E5470 laptop, having an Intel i7.6600U 2.60 GHz CPU and 16 GB RAM. The PNS solver version v2.0.3 was used, running on Windows 10.

The significantly larger solution time can be attributed to the fact that in the MILP model, CHP plants of the same kind are represented as integer variables instead of multiple binary variables to decrease redundancy, while in the P-Graph implementation, each CHP plant has its own operating unit node. This redundancy could be mitigated by introducing an artificial ordering of the interchangeable operating unit nodes, but this was omitted from this work. The ABB algorithm itself could treat redundancy better. The PNS implementation could also allow the decision variable of an operating unit to possibly be an integer rather than a binary variable.

Note that the PNS solver allowed to generate a ZIMPL implementation of a MILP model „technically equivalent" to the PNS problem itself, used in the solution procedure. This technically equivalent MILP model was further investigated.

First, this is not equivalent to the aforementioned MILP model for flexible inputs. This is due to several reasons, one is the aforementioned implementation of multiple CHP plants of the same kind. Another reason is that material flows in a P-Graph are represented by an operating unit node which introduces a binary decision variable, while in MILP models the binary decision variable can possibly be omitted.

The technically equivalent MILP model was transformed into CPLEX LP format, and then solved by the CBC and GLPSOL MILP solvers. The CBC solver finished in 19.99 s reporting the same solution, which is still significantly more than the initially formulated MILP model. The GLPSOL solver could not finish solving the problem in 1,000 s.

Nevertheless, the fact that the same solution was reproduced as a PNS problem indicates that the P-Graph framework and its existing tools provide an alternative to mathematical programming models, and can be extended by modeling techniques.

Appendix B

Nomenclature for MWM model

This appendix chapter provides the nomenclature for the mobile workforce management approaches presented in Chapter 5. Symbols used in the description of the standalone MILP model the and algorithmic framework are listed here. Some sets, parameters and variables are only relevant in the algorithmic framework. These are explicitly mentioned.

Sets

Sets are given with the typical index symbols for their elements.

$k \in K$	Set of tasks.
$k \in K^{done}$	Set of tasks already scheduled (algorithm only). $K^{done} \subseteq K$
$k \in K^{rem}$	Set of tasks not scheduled yet (algorithm only). $K^{rem} = K \setminus K^{done}$
$m \in M$	Set of teams.
$(m, i) \in J^{slots}$	Set of job slots. $i \in [1, N_m]$
$(m, i) \in T^{slots}$	Set of travelling slots. $i \in [0, N_m]$
$(m, i) \in X^{slots}$	Set of site slots. $i \in [0, N_m + 1]$
$(k_1, k_2) \in P^{prec}$	Set of all precedence relationships. $P^{prec} = P^{free} \cup P^{prot} \cup P^{same}$
$(k_1, k_2) \in P^{free}$	Set of free precedence relationships. $k_1, k_2 \in K$
$(k_1, k_2) \in P^{same}$	Set of same-team precedence relationships. $k_1, k_2 \in K$
$(k_1, k_2) \in P^{prot}$	Set of protected precedence relationships. $k_1, k_2 \in K$
$(k_1, k_2) \in P^{mtx}$	Set of mutual exclusion relationships. $k_1, k_2 \in K$
$(k_1, k_2) \in P^{par}$	Set of parallel execution relationships. $k_1, k_2 \in K$
$r \in R$	Set of all resources.
$r \in R^{cons}$	Set of consumable resources. $S^{cons} \subseteq S$
$r \in R^{tool}$	Set of tool resources. $S^{tool} = S \setminus S^{cons}$
$s \in S$	Set of all sites.
$s \in S^{depot}$	Set of team depot sites. $S^{depot} \subseteq S$
$s \in S^{tasksites}$	Set of task execution sites. $S^{tasksites} = S \setminus S^{depot}$

Parameters

If a parameter is numeric, it is assumed to be nonnegative. The domain of indices is by default always the largest possible set for which the symbol is used. For example, the index k means the parameter is defined for all $k \in K$ unless otherwise stated.

C_m^{travel} Cost factor for the total travelling time of team m.

C_m^{unpack} Unpacking cost of team m upon arrival on a site.

C_m^{pack} Packing cost of team m before departure from a site.

$C_k^{earliness}$ Penalty cost factor if task k is executed earlier than $T_k^{expected,start}$.

$C_k^{lateness}$ Penalty cost factor if task k is executed later than $T_k^{expected,end}$.

$C_{k,m}^{exec}$ Cost of task k if executed by team m.

C_r^{res} Cost of utilization of one unit of resource r.

C_{k_1,k_2}^{opcl} Cost of protected precedence relationship $(k_1, k_2) \in P^{prot}$ if the closing and opening option is chosen.

C_m^{work} Cost factor for the total working time of team m.

D_{s_1,s_2} Travelling distance from site s_1 to site s_2.

$D_m^{travel,max}$ Maximum total distance that team m may travel.

H_k^{slot} Job slot to which task k was assigned (algorithm only). $k \in K^{done}$, $H_k^{slot} \in J^{slots}$

N_m Number of predefined job slots for team m.

Q_r^{cap} Total available amount of resource r.

$Q_{r,k,m}^{req}$ Requirement of resource r for execution of task k by team m.

$Q_{r,m}^{max}$ Maximal amount of resource r that team m is allowed to carry.

S_m^{start} Starting position of team m. $S_m^{start} \in S^{depot}$

S_m^{end} Final position of team m. For simplicity, $S_m^{start} = S_m^{end}$ is assumed.

S_k^{task} Site of task k.

$T_k^{earliest}$ Start of absolute time window of task k.

T_k^{latest} End of absolute time window of task k.

$T_k^{expected,start}$ Start of expected time window of task k.

$T_k^{expected,end}$ End of expected time window of task k.

$T^{day,start}$ Starting time of the workday.

$T^{day,end}$ Ending time of the workday.

$U_{k,m}^{exec}$ Net time spent by team m for executing task k if assigned.

U_m^{pack} Packing time of team m before departure from a site.

U_m^{unpack} Unpacking time of team m after arrival on a site.

U_k^{close} Site closing time after task k is executed, if k is first in a protected precedence relationship.

U_k^{open} Site opening time before task k is executed, if k is second in a protected precedence relationship.

$U_m^{travel,max}$ Total time team m may spend travelling, (un)packing or being idle.

$U_m^{work,max}$ Total time team m may spend in duty during the day.

$U^{workday}$ Length of workday. $U^{workday} = T^{day,end} - T^{day,start}$

V_m Speed of team m for travelling between different sites. $V_m > 0$

Binary decision variables

Each binary decision variable represents a choice about a logical statement. The value 1 means the statement is fulfilled, and the value 0 means it is not.

$a_{k,m,i}$ Task k is executed by team m in its job slot $(m, i) \in J^{slots}$.

$a^{task}_{k,m}$ Task k is executed by team m.

$b^{present}_{m,i,s}$ Team m is present at site s during site slot $(m, i) \in X^{slots}$.

b^{sch}_{m,i,s_1,s_2} Team m moves from site s_1 to site s_2 in travelling slot $(m, i) \in T^{slots}$. $s_1 \neq s_2$

$b^{travel,move}_{m,i}$ Team m changes site in travelling slot $(m, i) \in T^{slots}$.

p^{close}_k Site S^{task}_k of a task k first in a protected precedence relationship is closed after executing task k.

p^{open}_k Site S^{task}_k of a task k second in a protected precedence relationship is opened before executing task k.

$p^{prot}_{k_1,k_2}$ Task site closing and opening is chosen as part of a protected precedence relationship $(k_1, k_2) \in P^{prot}$ between executing k_1 and k_2.

$p^{mtx}_{k_1,k_2}$ Task k_1 is chosen to be executed before k_2 to fulfill a mutual exclusion relationship $(k_1, k_2) \in P^{mtx}$.

x^{task}_k Task $k \in K^{rem}$ is the selected, newly assigned task (algorithm only).

x^{task}_m Team m is selected to execute the new task (algorithm only).

$x^{slot}_{m,i}$ The new task is assigned to job slot $(m, i) \in J^{slots}$ of team m (algorithm only).

$y_{m,i}$ The new task is assigned before travelling slot $(m, i) \in T^{slots}$ (algorithm only).

Note that $a_{k,m,i}$, $p^{prot}_{k_1,k_2}$ and $p^{mutex}_{k_1,k_2}$ from the original MILP formulation, and x^{task}_k, x^{team}_k and $x^{slot}_{m,i}$ used by the algorithmic framework are required to be binary. The constraints ensure that the values of all other variables shown here are unambiguously determined by the values of the ones mentioned. Therefore, all other binary decision variables can be continuous $[0, 1]$ variables in the model implementation.

Continuous variables

Continuous variables are bounded by $[0, \infty[$, except for variables representing time points, which are bounded by $\left[T^{day,start}, T^{day,end}\right]$, unless explicitly stated otherwise.

$c^{pen,early}_k$ Penalty cost for starting the execution of task k earlier than $T^{expected,start}_k$.

$c^{pen,late}_k$ Penalty cost for finishing the execution of task k later than $T^{expected,start}_k$.

c^{travel}	Total cost of travelling.
$c^{packing}$	Total cost of packing and unpacking at sites.
c^{tw}	Total cost of penalties for violating expected time windows.
c^{exec}	Total cost of task execution.
c^{res}	Total cost of resource utilization.
c^{opcl}	Total cost of opening and closing activities for fulfilling protected precedence relationships.
c^{work}	Total cost of team working times.
c^{total}	Total of all costs.
$d_{m,i}$	Distance travelled by team m in its travelling slot $(m, i) \in T^{slots}$
$q^{req}_{r,m,i}$	Amount of resource r used in job slot $(m, i) \in J^{slots}$.
$q^{carry}_{r,m}$	Amount of resource r carried by team m from its starting depot.
t^{start}_{k}	Starting time point of the execution of task k.
t^{end}_{k}	Ending time point of the execution of task k.
$t^{presence,start}_{k}$	Starting time point of the job slot in which task k is executed.
$t^{presence,end}_{k}$	Ending time point of the job slot in which task k is executed.
$t^{travel,start}_{m,i}$	Starting time point of travelling slot $(m, i) \in T^{slots}$.
$t^{travel,end}_{m,i}$	Ending time point of travelling slot $(m, i) \in T^{slots}$.
u^{close}_{k}	Time spent closing at site S^{task}_{k} of task k after its execution.
u^{open}_{k}	Time spent opening at site S^{task}_{k} of task k before its execution.
$u^{idle}_{m,i}$	Idle time spent by team m in travelling slot $(m, i) \in T^{slots}$.
u^{slack}_{k}	Amount of delay in the net execution time for task k.
$u^{wait,before}_{k}$	Waiting time spent by the team before executing task k.
$u^{wait,after}_{k}$	Waiting time spent by the team after executing task k.