Optimizing Urban Multi-Energy Systems using Column Generation

by

M. B. Elgersma

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday November 22, 2022 at 1:30 PM.

Student number:	4585011	
Project duration:	February 1, 2021 – November 22, 2022	
Thesis committee:	Dr. ir. L. J. J. van Iersel,	TU Delft, Responsible Associate Professor
	Dr. Ir. J. T. van Essen,	TU Delft, Daily Supervisor
	Dr. Ir. W. G. M. Groenevelt,	TU Delft
	I. van Beuzekom, MSc.,	ORTEC

ORTEC

TU Delft

Preface

The thesis in front of you describes the final project of my master's degree in Applied Mathematics at the Delft University of Technology. I am proud of the final result, and grateful for the process that led to it. When starting my first year as an applied mathematics student in Delft, I never anticipated that I would enjoy mathematics, specifically optimization, as much as I do. Programming and doing an extensive research project on my own were an entirely new experience to me. Yet, I am starting a PhD project in algorithmics and optimization after this. I could not have done it without the incredible support of the following people.

First and foremost, I would like to thank Theresia and Iris for guiding me throughout this project. At times, it almost seemed as though you had made it into a competition who could give the most helpful points of feedback on a new chapter I had written. Besides ensuring a satisfactory progress of my research, you always checked in on how I was doing personally as well. Theresia; I am very thankful for your attention to detail. Without your help, I might still be looking for the errors in my formulas and code. Iris; thank you for your endless patience when explaining your research, and for all your constructive feedback on writing an accessible and concise explanation of a complex mathematical method.

Furthermore, I would like to thank a few people that supported me throughout the project, by making sure I was never alone and always motivated to keep going. To Eva: thank you for being my thesis buddy and dragging me out of my house every once in a while. To my old roommates, Lieke, Robyn and Daniëlle, and my old-and-new roommate Marieke: thank you for being such a great daily support at home, for patiently listening to my struggles and comforting me with tea-breaks and home-baked goods. To Avelon and Nadine: thank you for helping me take my mind off my thesis when ice skating and swimming together. And last but definitely not least, a huge thank you to my parents and my partner Quinten, for checking in on me every step of the way, cheering me on and keeping me sane.

Maaike Elgersma
Delft, November 2022

Contents

1 Introduction

Toward 2050, a transfer from fossil fuels to renewable energy sources (RES) is needed to work toward climate goals. Urban decision makers need advice on the best way to implement these changes to the energy infrastructure, while always fulfilling the energy demand of each of these carriers. The energy system handles different energy carriers in the same system, and is therefore called a multi-energy system (MES). In this introduction, the motivation for this research is first given in Section 1.1 by explaining the urgency for and challenge of designing such a MES. Thereafter, the aim of this research is given in Section 1.2, namely to reduce the computation time of the multi-energy system design model as designed by Van Beuzekom et al. (2021). The methods that are used in this research to work toward the research objective are also briefly introduced in this section. In Section 1.3, the broader contribution of this research is explained. Lastly, the outline of the rest of this thesis is presented in Section 1.4, displaying the structure in which the research is presented.

1.1. Research motivation

Due to climate change, there is an immediate need for the decarbonization of our energy system. Recent developments in climate change policies, such as the Paris Climate agreement in 2015 ("Paris Agreement", 2015), are implemented to enforce this change. Plans have been made to reduce the production and need for fossil fuels toward 2050. To this end, the large-scale implementation of RES into the energy infrastructure is necessary, for which the prices are increasingly competitive (Ayre, 2015). Therefore, it should be studied which adjustments to the energy infrastructure in cities are required to work toward these climate goals, while fulfilling the energy demand at all times. The objective of this research is to advise urban decision makers on the best plan to achieve this.

Currently, most research on climate action focuses on either a large scale (national level) or a small scale (building level), but not on urban level. Furthermore, the focus is either on a long-term future scenario or the short-term operational challenges, but not on the pathway from today toward these future scenarios. Lastly, most research aims at optimizing the energy infrastructure of a single energy sector, but not at the integration of different energy sectors (Van Beuzekom et al., 2021). This research aims to optimize the multi-energy system of cities from today to 2050.

When optimizing such a multi-energy system, the objective is to minimize the cost of the transition from fossil fuels to RES, whilst the demand of all energy types should be met at all times. Costs include capital costs from opening a new energy supply facility, storage assets, conversion assets or distribution assets in the energy infrastructure, as well as operating costs. However, even without considering the costs, it is a complicated process for cities to transfer from fossil fuels to RES. The challenges that come with this energy transition are depicted in Figure 1.1. There are two main problems in this transition. Firstly, RES provide electricity, but currently only 30% of the world's energy demand is electricity ((IEA), 2022). For example, many households still require gas for cooking or heating. This results in an energy carrier mismatch. Secondly, RES, such as wind and solar energy, are intermittent at daily, seasonal and even annual scales (Van Beuzekom et al., 2017). This results in a temporal mismatch of energy production and demand.

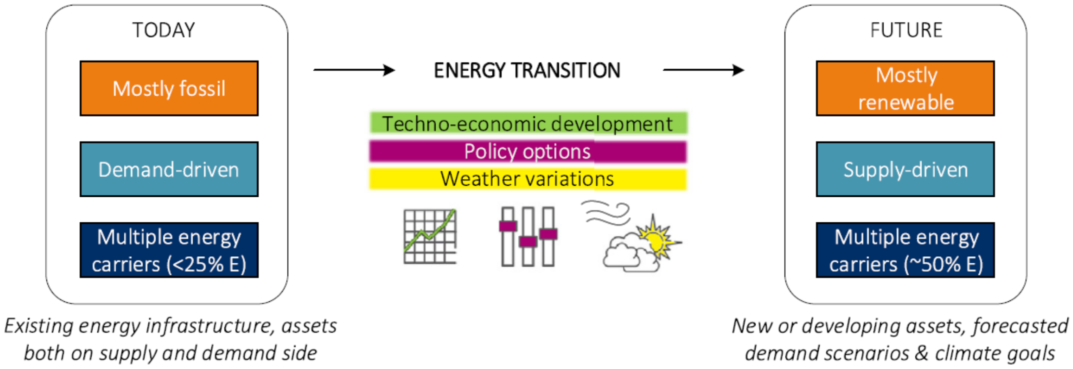

TODAY
Mostly fossil
Demand-driven
Multiple energy carriers (<25% E)
ENERGY TRANSITION
Techno-economic development
Policy options
Weather variations
FUTURE
Mostly renewable
Supply-driven
Multiple energy carriers (~50% E)
Existing energy infrastructure, assets both on supply and demand side
New or developing assets, forecasted demand scenarios & climate goals

Figure 1.1: Challenges of the energy transition - from today to a sustainable future (Van Beuzekom et al., 2021).

1.2. Research aim

To tackle the energy transition challenges, Van Beuzekom et al. (2021) formulated a mixed-integer linear program (MILP). This MILP models a multi-period planning with long-term investments for integrated urban MES. The MES model can help with decision making in urban multi-energy system design and investment planning. The optimization problem in the MILP combines a capacitated facility location problem with a multidimensional, capacitated network design problem, which are both NP-hard. Even with just two energy carriers and one time period, the optimization problem is strongly NP-hard (Bonenkamp, 2020). Therefore, the computation time needed to solve the model with state-of-the-art solvers, or even find a reasonably good solution for the large, realistic cases, is currently a limiting factor.

The aim of this research is to investigate ways in which a reasonably good solution of the MES model, when applied to large, real-life cases, can be found within a certain amount of time. The method that is applied in this research to work toward this aim is column generation. Thus, the main research question is:

Can column generation improve the quality of the solution found by a model that optimizes the design of a real-life urban multi-energy system, as described by Van Beuzekom et al. (2021), within a fixed amount of time?

To answer this question, column generation is applied to the MILP of the MES model (hereafter just called ‘the MES model’) and tested on the case of Eindhoven as studied by Van Beuzekom et al. (2021). The data that is used is largely greenfield, rather than brownfield. Brownfield means that all existing parts of the energy infrastructure are included. Greenfield means that no starting data of the infrastructure is included. In the used data, no starting data is included, except for the existing gas supply and practical investment restrictions. For example, large wind supply assets can only be built in some locations in the city. The results of column generation are compared to the original MES model, when solved by the Gurobi solver.

The column generation algorithm requires an initial solution to start with. To this end, a greedy algorithm is developed in this research, which is able to find a feasible, though likely non-optimal, solution to the MES model. This solution can be used as a starting solution for column generation.

1.3. Contribution of this research

As mentioned earlier, the model by Van Beuzekom et al. (2021) combines a capacitated facility location problem with a capacitated network design problem. Most related literature focuses on only one of these problems, or an uncapacitated combination of the two problems is considered. Only in the work by Rahmaniani and Ghaderi (2015) this combined capacitated problem is considered, but only a single time period is considered. Thus, this research investigates the solvability of a novel combination of problems, which implies significant, additional mathematical complexity. The focus of this research is the quality of the solutions to the model as formulated by Van Beuzekom et al. (2021) that are found within a certain amount of time for large cases.

Such an improvement of computed solutions was already researched by Bonenkamp (2020) in her thesis. She successfully improved the quality of the solutions found within a certain amount of time by adding several valid inequalities and tuning the parameters of the Gurobi solver ("Gurobi - The Fastest Solver", 2022). Section 2.1.2 elaborates on these methods and results. However, the problem is very large in reality, and finding a relatively good solution for realistic cases still takes a very long time. Therefore, this thesis attempts to improve the solutions found for large instances within a certain amount of time, by applying column generation.

1.4. Thesis outline

The thesis describing this research is structured as follows. First, relevant research related to this research is discussed in Chapter 2, to give context to this research, and to indicate more clearly which research gaps exist. This further demonstrates the contribution of this research. Then, the current MES model by Van Beuzekom et al. (2021) is introduced in Chapter 3. More specifically, the notation, input data, constraints, and objective function are explained. Building on this information, Chapter 4 introduces a greedy algorithm that can be used to find a starting solution to the MES model, which is needed for column generation. The general idea of the algorithm is first explained, after which the details of the algorithm are expanded upon, including elaborate examples of calculations in the algorithm. Chapter 5 explains the general theory of column generation, as well as explaining the four ways in which it can be applied to the MES model. In Chapter 6, more information is given on the tests that are performed to evaluate the performance of the current MES model, the greedy algorithm and column generation. The used data sets are explained, as well as detail on the implementation of the MES model, the greedy algorithm and column generation. The results of these tests are then given in Chapter 7. The performance of the model, greedy algorithm and column generation are compared to each other, and the solutions are further investigated to explain the differences. Lastly, in Chapter 8, the research is concluded and discussed.

2 Literature review

In this chapter, some work related to the MES model and similar problems is discussed to give context to this research. Moreover, reviewing the current literature on the topic gives an insight into which research gaps exist. In Section 2.1, an overview of previous work on the subject of multi-energy system design is given. The main focus is on the MES model as described by Van Beuzekom et al. (2021). Thereafter, Section 2.1.2 discusses research on improving solutions to this model found within a certain amount of time, and Section 2.1.3 gives an overview of the research about dealing with uncertainty in the MES model. Lastly, in Section 2.2, relevant literature on the solution proposed in this thesis, namely column generation, is described.

2.1. Multi-energy system optimization

In this section, an overview of previous research on the MES model is given. The research by Van Beuzekom et al. (2021) is unique in researching multi-energy systems in cities, with a temporal scope from today to 2050. Therefore, research leading up to this model and the most current research on the model are first discussed. Thereafter, further research on this model is reviewed.

The paper "A review of multi-energy system planning and optimization tools for sustainable urban development" by Van Beuzekom et al. (2015) gives a review of tools able to model multi-energy systems, applicable to a city scale and incorporating short- and long-term dynamics of an energy system. The conclusion of this research is that some tools combine planning and operational methods. However, none are able to model solutions that are practically feasible from a grid perspective, and none incorporate proper modeling of the power system, either because the used time step is too large or because the electricity grid is not modeled at all.

In 2016, the first paper proposing a model for optimal design and operation of an integrated multi-energy system for smart cities is published by Van Beuzekom et al. (2016). The main objective of this model is to optimize the topology and operation of electricity, heat and gas networks simultaneously. This way, the value of the integrated energy system is optimized to meet a certain energy demand, rather than the individual networks. The geographical scope of the model is everything within the administrative boundaries of a city and the temporal scale is one year (namely 2050), with fifteen-minute time steps. For the demand, residential, commercial, and industrial demand profiles are used as input data. Energy technologies for the supply, conversion and storage of electricity, heat, and gas are considered. In this paper, some promising results of a proof of concept are demonstrated by applying the model to the case of the Strijp district, which is part of the city of Eindhoven. This case is chosen because of its high diversity in current energy infrastructures, consumers, and producers, and because of its innovative and creative character. It is concluded that the model indeed optimizes the multi-energy system. Thus, the model provides insight into the extent of the possibilities offered by an integrated energy system at an operational scale for cities.

The paper by Van Beuzekom et al. (2017) called 'Optimal planning of integrated multi-energy system' expands upon the model by Van Beuzekom et al. (2016) by using two-step optimization. In this two-step optimization, the green- and brownfield scenarios are solved first for an intermediate time step in 2030, and then for 2050. The research is again applied to the case of the Strijp district. Both greenfield and brownfield data are considered for the case. The results are that greenfield designs are more cost efficient, as their results are not constrained by the existing infrastructure. However, in reality, a combination of both the greenfield and brownfield approach will be required.

In 2021, another paper on this model was published by Van Beuzekom et al. (2021). It describes the same framework for long-term, multi-period investment planning of integrated urban energy systems. Different to previous papers, the model is applied to the entire city of Eindhoven. Additionally, eight carbon-emission reduction scenarios are included in the data, to analyze the carrier mismatch problem. Moreover, eight different interannual weather variation scenarios are included in the data to study the effects of the temporal mismatch problem. Our research continues the research on the model as described in this paper.

Bonenkamp (2020) showed in her thesis that the multi-energy system design problem is strongly NP-hard. This motivates the use of heuristics to try to improve the solutions found to the model by Van Beuzekom et al. (2021) within a certain amount of time.

2.1.1. Improving solution quality for 2050

In previous research on the model by Van Beuzekom et al. (2021), the multi-energy system design problem is investigated for the year 2050 only. This is a different temporal scope to our research, since we optimize the MES design over the years from now to 2050, rather than for one time period. Stant (2019) developed valid inequalities and multiple decomposition methods in his thesis, namely a phase model, a multigrid approach and a fix-and-optimize heuristic, to find relatively better solutions to the MES design problem for the year 2050 within a certain amount of time, with a model based on the model by Van Beuzekom et al. (2017). The objective is to reduce the optimality gap when limiting the solving time of the mathematical model to 3600 seconds. Note that for all methods explained in this section (but the valid inequalities), this optimality gap is computed between the lower bound found by the mathematical model with added valid inequalities and the upper bound found by the other methods within 3600 seconds.

The valid inequalities found by Stant (2019) are based on the special aspects of the single time period problem. For example, there is no gas supply in 2050, and the lack of time steps means there is no possibility for energy storage. This results in valid inequalities about the minimum number of RES, heat pumps (HPs) and power-to-gas (P2G) units required, and that at each node, there is at least either an investment in a supply unit or a network connection for each demanded type of energy. The addition of these valid inequalities for the 2050 case improves the optimality gap for larger cases, compared to the original mathematical model. For the largest instance of 110 nodes, the added valid inequalities result in an optimality gap of 8.04%, versus an optimality gap of 10.00% found by the original mathematical model.

The phase model by Stant (2019) decomposes the energy infrastructures, such that they can be solved separately. First, the gas and heat networks are solved separately. Secondly, the demand for the electricity nodes is updated, based on the obtained solution for the gas and heat networks. Lastly, the electricity network is solved with these adjusted demands. The results of this phase model are similar to those of the added valid inequalities, except it performs even better for the largest case. For the largest instance of 110 nodes, the application of the phase model results in an optimality gap of 6.72%.

The multigrid approach by Stant (2019) coarsens the problem by aggregating over the nodes, using a coarsening factor or a general clustering method. This coarsened problem is solved, after which back-mapping is used to compute the solution of the main problem. The results of this multigrid method are again that the optimality gap is smaller than that of the mathematical model with added valid inequalities for the large cases. The best result for the largest instance of 110 nodes is an optimality gap of only 6.50%.

Lastly, a fix-and-optimize heuristic in combination with a firefly algorithm are applied to the 2050 case by Stant (2019). The firefly algorithm is used to determine which facilities to open. This solution is used as input data, after which the model is solved with these fixed facilities. These fixed supply units result in a considerably higher lower bound than was found by the original mathematical model with added valid inequalities. However, for the smallest instance of 7 nodes, the fix-and-optimize heuristic is able to find the optimal solution. For larger instances, it results in a smaller optimality gap than is found by the original mathematical model with added valid inequalities. For the largest instance of 110 nodes, the application of the phase model resulted in an optimality gap of 6.73%.

Although this research succeeds in improving the solutions found to large instances within 3600 seconds for the 2050 case only, the scope of this research is to improve the solutions to the model toward 2050.

2.1.2. Improving solution quality toward 2050

In this section, research dedicated to improving the solutions found within a fixed amount of time to the model as described by Van Beuzekom et al. (2021) toward 2050, rather than for only the year 2050, is discussed. Several methods have been applied to reduce the optimality gap between the lower bound found by the original mathematical model and the upper bound found by these methods. In this section, an overview of the research on these methods and the results is given.

Stant (2019) applied a rolling horizon method to the problem of optimizing the MES from 2014 up to 2050 as described by Van Beuzekom et al. (2017). The only difference between the models of these two papers is that for a network connection between two locations, Stant sets the maximum capacity for each direction separately, whilst Van Beuzekom sets the maximum capacity for the flows in both directions combined. The idea of the rolling horizon method is that the problem is solved for each time step sequentially. This time step can be one year, or multiple time periods combined. After the problem is solved for the first time step, the investments from this period are taken as input for the next time step. This process is continued, in which the investments of earlier time steps are constantly taken as input for the next time step. Stant finds that the rolling horizon method has the best results regarding the computation time and obtained the best solution for the 110 locations problem. However, the computation time is still quite large (10.1 hours for a case of 110 nodes and 34 time periods) and the optimality gap was quite large (23.17%).

Bonenkamp (2020) applied several methods to improve the solutions found to the multi-energy system design model by Van Beuzekom et al. (2017). Bonenkamp rewrote the model by Van Beuzekom et al. to a network flow problem, but the solution space of this model is exactly the same as that of the model by Van Beuzekom et al. (2017).

In her thesis, Bonenkamp (2020) tried parameter tuning of the branch-and-bound algorithm of the Gurobi solver in order to improve the solutions found to a model based on the model by Van Beuzekom et al. (2017). More specifically, aggressive cut generation, aggressive model-specific cuts, strong branching and node selection based on best bound are applied in this research. Aggressive cut generation involves letting the solver spend more time on calculating flow cover cuts throughout the entire algorithm. The application of aggressive model-specific cuts is focused on letting the solver spend more time calculating flow cover cuts throughout the entire algorithm, since these are suitable for the multi-energy system problem. Strong branching is done by running a number of iterations of each candidate branching variable at each node, in order to find the branching variable that gives the best improvement in bound. Similarly, the best node to branch on is selected by finding the node with the lowest objective value of the relaxation. The results indicate that forcing the solver to apply more flow cover cuts has a positive impact on the incumbent solution found by the solver. For the 7 nodes case, an optimality gap of 1.92% was found after 7200 seconds.

Furthermore, Bonenkamp (2020) found several model-specific valid inequalities. The first set of inequalities ensures that all energy coming into a node from supply, from other nodes and from storage is at least higher than the demand. The second set of inequalities originate from the idea that energy taken from storage cannot be larger than the total installed capacity of storage, taking into account loss and efficiency factors. The third set of inequalities ensures that the total inflow in a node from other edges cannot be larger than the total installed capacity of the edges. The fourth set of inequalities concerns the maximum capacity bound for supply. All these inequalities are then combined and strengthened for the gas- and heat nodes. Lastly, several valid inequalities based on graph structure are added. All these valid inequalities are added a priori, as lazy constraints with callback function and as a priori aggressive lazy constraints. The results indicate that adding the valid inequalities as lazy constraints is the preferred strategy, and results in a significant reduction in computation time needed by the solver for the small instance of 5 nodes and 13 time periods. For the 7 nodes case, this resulted in an optimality gap of 1.43% after 7200 seconds.

Lastly, Bonenkamp (2020) applied two decomposition methods: Benders decomposition method and the Lagrangean relaxation method. However, these both do not seem efficient for solving the multi-energy system design problem.

Thus, some research succeeds in improving the solutions found within a certain amount of time to the model toward 2050. However, it is still not possible to find relatively good solutions to very large instances.

2.1.3. Including uncertainty

Besides investigating an improvement of solutions to the model found within a certain amount of time, research has also been done on how the model can deal with uncertainty in the input data. This research is summarized in this section.

Fraiture (2020) wrote a thesis titled “The Robustness of energy systems: a novel method to explore the impact of uncertainties on energy system design optimization models”. In this thesis, she studies how uncertainties such as technology innovations, resource availability and socio-economic dynamics influence the model outcomes. The objective of this research is to propose a generally applicable method with which model-owners can be provided insight into the impact of uncertainties on Energy System Design Optimization Model outcomes. Cosine distance-based agglomerative hierarchical clustering with complete linkage are used to achieve this. She finds that the most determining uncertainties are the PV supply development rate and demand development, especially gas. A lower PV development rate results in higher wind supply investments. A high demand of gas, or a delayed decrease in gas demand, results in high investments in P2G assets, solar supply assets and gas storage capacity and low investments in combined heat and power assets (CHPs).

Thio (2020) also investigates uncertainties in her thesis “Decision making under uncertainty for multi-energy systems: implicitly modelling of uncertainty in the long-term investment planning of city-scaled integrated energy systems”. In this thesis, a multi-stage stochastic programming (MSSP) framework using scenario trees is presented, which can be used to construct robust solutions. This is useful, since the deterministic approach already fails to meet 10% of the total energy demand after the first 2 years when uncertainty is involved, and this percentage increases over time. First, the most important uncertainties are determined, which are demand, investment costs and external factors (political and environmental influences on the supply). Then, scenario trees are constructed, and the framework is again applied to the brownfield data of Eindhoven. The results show that a multi-energy system, constructed by MSSP is on average 19% more robust and 9% more sustainable than the outcomes of the deterministic approach. However, the investment costs of the MSSP approach are 40% higher, which is mainly related to black outs.

Van Kampen (2022) investigates in her thesis how uncertainty of weather can be taken into account in the model by Van Beuzekom et al. (2021). A limited set of scenarios that reflect the distribution of uncertain capacity factors are incorporated in the input data. A newly proposed stochastic optimization approach of using a chance constraint is used to deal with these uncertainties. This constraint ensures that the demand and supply match at least (1-α)% of the time. The success is measured in terms of the occurrence and size of violations (situations in which the demand of at least one energy carrier cannot be met). Van Kampen (2022) finds that this stochastic approach results in significantly fewer cases where demand cannot be met compared to the existing deterministic approach, though the computation time and total cost is higher.

Van der Veen (2021) assesses the impact of short-term weather effects on investment decisions for the integrated multi-energy model by Van Beuzekom et al. (2021) in his thesis. He designs an integration module which soft-links an existing strategic decision-making model of the company ORTEC with EnergyPLAN. EnergyPLAN is a simulation tool which enables hourly analysis of strategic investments decisions. This integration module is established such that it allows for dynamic adaptation to analyze varying inputs, such as energy demand profiles and energy technology properties. Multiple sensitivity analyses are executed to validate the methodology and to assess its robustness. The results show that the intermittency of renewable energy sources causes instabilities on the power grid. Therefore, technologies like energy storage and power-to-gas require large-scale deployment, which is complicated due to their high costs. Moreover, results show that the metrics to which a weather year should be assessed, demand a deeper analysis than its annual average capacity factor. This underlines the importance of the development of the aforementioned technologies.

Although it is important to take uncertainties into account, they are not included in this research, since this only complicates the model and increases the difficulty of finding a good solution to a large instance. In the research that includes uncertainty, only small instances of the problem are considered (7 nodes or less), but the scope of this research is to find good solutions to large instances of 110 nodes.

2.2. Column generation applications

Column generation was first introduced by L. R. Ford and Fulkerson (1958). It has been successfully applied to the capacitated facility location problem by Klose and Görtz (2004), and to the capacitated network design problem by Gong (1996). The MES model is combination of these two NP hard problems. More recently, as the problem of optimizing integrated energy systems has become more important, column generation has been applied to models related to the energy transition, such as in the following two papers.

Flores-Quiroz et al. (2016) apply column generation to a generation expansion model, with unit commitment constraints included. A generation expansion model optimizes the planning problem of integrating new generation units into a system. Constraints ensure that it satisfies technical and financial limits. In this research, it is used to incorporate renewables into future power systems. This is a complex transition, as incorporating renewables significantly increases the need for flexible operational measures and generation technologies. To this end, unit commitment constraints are included. The associated investment decisions must be properly planned in the long term. Several planning horizons are considered in the study, from 1 to 31 years. The addition of unit commitment constraints results in a large scale Mixed-Integer Nonlinear Program (MINLP) that requires significantly more computation time to be solved. Therefore, column generation is applied. The Dantzig-Wolfe decomposition algorithm is applied, by splitting the original model up into subproblems per year. A column for a certain year represents the investments made in renewables in that year. It is demonstrated through multiple case studies that the proposed column generation approach outperforms the CPLEX solver, reducing both the needed computation time and memory, especially for larger planning horizons. For the largest case, the CG algorithm needs 15 days to find a solution with an optimality gap of 0.49%. The CPLEX solver is not able to find a solution, due to limited RAM.

Muts et al. (2021) apply column generation to energy system MINLP models. Experiments are performed on Decentralized Energy Supply System (DESS) models. DESS are complex, integrated systems consisting of several energy conversion units and energy supply and demand forms. Usually, a superstructure is formulated, containing all possible components and their interconnections. Optimization of DESS involves optimizing the design and operating schedule of a plant consisting of superstructures. Column generation is applied by formulating subproblems for each superstructure. Several methods are added, to speed up the computations. NLP local search is used to generate columns more quickly. Furthermore, a Frank-Wolfe algorithm is used, which is and alternative way to generate column. The pricing subproblems are solved in parallel, are not solved to optimality, and do not need to be done at the same time. The numerical results show that column generation is able to obtain significantly better dual bounds and slightly better primal bounds than state-of-the-art MINLP solvers.

3 Model

This chapter describes the current MES model, as designed by Van Beuzekom et al. (2021). The model optimizes the design of urban multi-energy systems. The costs of investments in and usage of energy assets in a city are minimized, while fulfilling the energy demand at all times. This model is the starting point of this research, to which the new greedy algorithm and column generation are applied. Information on the contents of this model, such as the notation, constraints, and objective function, are needed to derive the formulations for column generation. Moreover, a general understanding of the model and its constraints are needed to design a greedy algorithm. In this chapter, the notation of all the sets in the model is first introduced in Section 3.1. This notation is needed to introduce the parameters in the model. The parameters that are given to the model as input are then described in Section 3.2. When all prerequisite information is given, the constraints can be explained, which is done in Section 3.3. Lastly, the objective function of the model is explained in Section 3.4.

3.1. Notation

In this section, the notation of the sets in the MES model is introduced. A full overview of the notation of the sets, as well as all the parameters and variables introduced in the following sections, is given in Appendix A. The energy network in a city consists of nodes V where assets can be placed, edges U and directed edges DU on which energy can be distributed between the nodes. There is one edge between every two nodes in V, and one directed edge from every node in V to another. An edge between nodes v and v' in V is denoted by $\{v, v'\} \in U$, which is the same as $\{v', v\}$. A directed edge from node v to node v' in V is denoted by $(v, v') \in DU$, which is different from (v', v). The difference between edges and directed edges is explained later, as this is relevant for the different variables. On these (directed) edges, energy of a certain energy carrier can be distributed from one node to another. These types of energy carriers are given by set E, for example gas, heat and electricity. At the nodes, energy can be supplied, converted, stored or distributed to other nodes. Such asset types are given by set A, which is the union of four disjoint subsets. These subsets are the set of energy supply asset types SA, the set of energy conversion asset types MA, the set of energy storage asset types WA and the set of energy distribution asset types DA. So $A = SA \cup MA \cup WA \cup DA$. Each of these four disjoint subsets in turn consist of disjoint subsets of asset types for each energy carrier $e \in E$, given by SA_e, MA_e, WA_e, and DA_e. So for example, $SA = \bigcup_{e \in E} SA_e$. Lastly, the time periods are given by set T, where t_0 denotes the first time period and t_{end} denotes the last time period.

3.2. Input data

The objective of the model is to minimize the costs of the energy transition from today to 2050, whilst fulfilling the energy demand. Important input data are therefore the costs and the demand. The different costs and the notation of these parameters are first explained, starting with the capital and operational costs of all asset types.

The fixed cost of investing in a distribution asset $d \in DA$ on edge $u \in U$ in time period $t \in T$ is given by o_{dut}^{fix}. The fixed costs of investing in a supply asset, conversion asset or storage asset at node $v \in V$

in time period $t \in T$ is given by o^{fix}_{avt}, with $a \in SA$, MA or WA, respectively.

The variable cost of using a distribution asset $d \in DA$ on edge $u \in U$ in time period $t \in T$ is given by o^{var}_{dut}. The variable cost of using a supply asset or a conversion asset at node $v \in V$ in time period $t \in T$ is given by o^{var}_{avt}, with $a \in SA$ or $a \in MA$, respectively. The variable costs of using a storage asset $w \in WA$ at node $v \in V$ in time period $t \in T$ for charging and discharging are $o^{+,var}_{wvt}$ and $o^{-,var}_{wvt}$, respectively.

Additionally, costs representing technological development of each type of asset are included. Parameter c_{dut} denotes this cost for distribution asset type $d \in DA$, on edge $u \in U$ in time period $t \in T$. Similarly, c_{avt} denotes this cost for an asset $a \in A \setminus DA$ at node $v \in V$ in time period $t \in T$. These costs are calculated as shown in Equations (3.1) and (3.2).

$$c_{dut} = c_{dut_0} \frac{(1-\phi_d)^{t-t_0}}{(1+\delta)^{t-t_0}}, \quad \forall d \in DA, u \in U, t \in T \tag{3.1}$$

$$c_{avt} = c_{avt_0} \frac{(1-\phi_a)^{t-t_0}}{(1+\delta)^{t-t_0}}, \quad \forall a \in A \setminus DA, v \in V, t \in T \tag{3.2}$$

Parameter ϕ_a denotes the technological development rate of each asset $a \in A$. Parameter δ denotes the social discount rate. As time progresses, the costs are expected to decrease as the technology is developed further. Since both the current costs and these future costs are considered in the model, it is important to include the social discount rate for a fair comparison.

The energy demand of energy carrier $e \in E$ at node $v \in V$ in time period $t \in T$ is denoted by d_{evt}. This data consists of predictions on the energy demand, which is influenced among other things by legislation for decarbonization.

Furthermore, an asset of type $a \in A$ has a capacity given by γ_a.

Lastly, some parameters of efficiency are taken into account. Energy is distributed on edge $u \in U$ with distribution asset $d \in DA$ with loss factor η_{du}. The supply of energy carrier $e \in E$ from supply asset $s \in SA_e$ in time period $t \in T$ depends on external factor σ_{st}. For example, for electricity supply assets, the value of this external factor is determined by the weather forecast for the years up to 2050, as well as policies. Energy carrier $e \in E$ is converted to energy carrier $e' \in E$ by conversion asset $m \in MA_e$ with an efficiency of $\eta_{mee'}$. Energy carrier $e \in E$ is stored in energy storage asset $w \in WA_e$ with an efficiency of η^+_w and withdrawn with an efficiency of η^-_w. Lastly, standing losses of energy storage assets $w \in WA$ are included and denoted by $(1-\eta^{sl}_w)$.

3.3. Constraints

The constraints included in the MES model are given and explained in this section, starting with the main constraint, which ensures that the energy demand is satisfied. This constraint is displayed in Equation (3.3). The variables in this constraint are the following. Variable F_{edut} indicates the amount of energy carrier $e \in E$ that is distributed with distribution asset $d \in DA_e$ on directed edge $u \in DU$ in time period $t \in T$. This variable is defined on the directed edges DU, since the amount of energy that is distributed from node v to node v' should be denoted by a different variable than the amount that is distributed from node v' to node v. Variable S_{esvt} indicates the amount of energy carrier $e \in E$ that is supplied by supply asset $s \in SA_e$ at node $v \in V$ in time period $t \in T$. Variable M_{emvt} indicates the amount of energy carrier $e \in E$ that is converted by energy conversion asset $m \in MA_e$ at node $v \in V$ in time period $t \in T$. Variables W^+_{ewvt} and W^-_{ewvt} indicate the amount of energy carrier $e \in E$ that is stored and withdrawn, respectively, from storage asset $w \in WA_e$ at node $v \in V$ in time period $t \in T$.

$$\begin{aligned}&\sum_{d \in DA_e} \sum_{v' \in V} F_{ed(v',v)t} \cdot \eta_{d\{v,v'\}} + \sum_{s \in SA_e} S_{esvt} \cdot \sigma_{st} + \sum_{\substack{e' \in E \\ e' \neq e}} \sum_{m \in MA_{e'}} M_{e'mvt} \cdot \eta_{me'e} + \sum_{w \in WA_e} W^-_{ewvt} \cdot \eta^-_w \\ &\geq d_{evt} + \sum_{d \in DA_e} \sum_{v' \in V} F_{ed(v,v')t} + \sum_{m \in MA_e} M_{emvt} + \sum_{w \in WA_e} W^+_{ewvt}, \qquad \forall e \in E, v \in V, t \in T\end{aligned} \tag{3.3}$$

In each node $v \in V$, a balance constraint as show in Constraint (3.3) ensures that the amount of energy carrier $e \in E$ that is being produced and being used by the different assets during time period $t \in T$ is balanced. Thus, most importantly, this constraint ensures that the demand is satisfied at all times. Energy is distributed from other nodes (with efficiency η_{du} on edge $u \in U$), it is produced by supply assets (taking into account external factor σ_{st}) and conversion assets (with efficiency $\eta_{me'e}$), and it can be withdrawn from storage (with efficiency η_w^-). This is included on the left-hand side of the constraint. The efficiency $\eta_{me'e}$ of conversion asset type $m \in MA$ is zero if this asset type does not convert energy carrier e' to e. On the other hand, energy can be used to fulfill the demand, be distributed to other nodes, be converted to other energy carrier types, and be stored. This is included on the right-hand side of the constraint. The exact amounts of energy that are distributed, converted, or stored are denoted here. Therefore, parameters for efficiency are not included on the right-hand side. A surplus of energy in a node is not a large issue, and large surpluses are not likely to occur since we are minimizing costs. Therefore, a ≥ sign is used, instead of a = sign.

All these variables have a maximum value, that depends on the number of investments done in assets in previous time periods. This is described by Constraints (3.4) - (3.8). Integer variable B_{dut} indicates the number of investments in energy distribution assets $d \in DA$ on edge $u \in U$ in time period $t \in T$. This variable is defined on the undirected edges U, since an in investment in a distribution assets between two nodes can be used to distribute energy in both directions. Integer variable B_{avt} indicates the number of investments in asset type $a \in A \setminus DA$ at node $v \in V$ in time period $t \in T$. The amount of energy carrier $e \in E$ that is distributed by distribution assets of type $d \in DA_e$ from node v to node v', denoted by parameter $F_{ed(v,v')t}$ cannot exceed the total capacity of such assets that was invested in during previous time periods on the edge $\{v, v'\}$. Similarly, the amount of energy carrier $e \in E$ that is supplied by supply assets of type $s \in SA_e$ or converted by conversion assets of type $m \in MA_e$ at node $v \in V$ in time period $t \in T$ cannot exceed the total capacity of such assets that was invested in during previous time periods. Furthermore, the amount of energy carrier $e \in E$ that is stored in storage assets of type $w \in WA_e$ at a node $v \in V$ in time period $t \in T$ cannot exceed the total capacity of such storage assets that was invested in during previous time periods, minus the total amount of this energy carrier that is already stored in such storage assets at the start of time period $t \in T$. On the other hand, the amount of energy carrier $e \in E$ that is withdrawn from storage assets of type $w \in WA_e$ at node $v \in V$ in time period $t \in T$ cannot exceed the total amount of this energy carrier that is already stored in such storage assets at this node at the start of time period $t \in T$.

$$F_{ed(v,v')t} \leq \gamma_d \sum_{i=t_0}^{t} B_{d\{v,v'\}i}, \quad \forall e \in E, d \in DA_e, v, v' \in V, t \in T \tag{3.4}$$

$$S_{esvt} \leq \gamma_s \sum_{i=t_0}^{t} B_{svi}, \quad \forall e \in E, s \in SA_e, v \in V, t \in T \tag{3.5}$$

$$M_{emvt} \leq \gamma_m \sum_{i=t_0}^{t} B_{mvi}, \quad \forall e \in E, m \in MA_e, v \in V, t \in T \tag{3.6}$$

$$W^+_{ewvt} \cdot \eta^+_w \leq \gamma_w \sum_{i=t_0}^{t} B_{wvi} - W^{tot}_{ewvt}, \quad \forall e \in E, w \in WA_e, v \in V, t \in T \tag{3.7}$$

$$W^-_{ewvt} \leq W^{tot}_{ewvt}, \quad \forall e \in E, w \in WA_e, v \in V, t \in T \tag{3.8}$$

Variable W^{tot}_{ewvt} represents the total amount of energy carrier $e \in E$ that is stored at node $v \in V$ in asset $w \in WA_e$ at the start of time period $t \in T$. It is equal to the total amount of energy that was stored at the end of the previous time period, minus some energy that is lost due to standing losses. The total amount of energy that was stored at the end of the previous time period is equal to the amount of energy that was stored at the start of the previous time period, plus the amount of energy that was stored in the previous time period, minus the amount of energy that was withdrawn in the previous time period. Thus, the value of W^{tot}_{ewvt} is set correctly by Constraints (3.9) and (3.10).

$$W^{tot}_{ewvt_0} = 0, \quad \forall e \in E, w \in WA_e, v \in V \tag{3.9}$$

$$\begin{aligned} W^{tot}_{ewvt} = (W^{tot}_{ewv(t-1)} + W^+_{ewv(t-1)} \cdot \eta^+_w - W^-_{ewv(t-1)}) \cdot (1 - \eta^{sl}_w), \\ \forall e \in E, w \in WA_e, v \in V, T \in T \setminus \{t_0\} \end{aligned} \tag{3.10}$$

All decision variables B_{dut} and B_{avt} for the number of investments in assets are integer, for all $d \in DA, a \in A \setminus DA, u \in U, v \in V, t \in T$, and cannot be larger than N. This value N is taken into account in order to make sure that the solution is feasible in practice. This is described by Constraints (3.11) and (3.12).

$$B_{dut} \leq N,\ B_{dut} \in \mathbb{Z}^{+}, \quad \forall d \in DA, u \in U, t \in T \tag{3.11}$$

$$B_{avt} \leq N,\ B_{avt} \in \mathbb{Z}^{+}, \quad \forall a \in A \setminus DA, v \in V, t \in T \tag{3.12}$$

$$\tag{3.13}$$

Lastly, all other variables are real-valued and positive, which is ensured by Constraints (3.14) - (3.17).

$$F_{edut} \in \mathbb{R}^{+}, \quad \forall e \in E, d \in DA_e, u \in DU, t \in T \tag{3.14}$$

$$S_{esvt} \in \mathbb{R}^{+}, \quad \forall e \in E, s \in SA_e, v \in V, t \in T \tag{3.15}$$

$$M_{emvt} \in \mathbb{R}^{+}, \quad \forall e \in E, m \in MA_e, v \in V, t \in T \tag{3.16}$$

$$W^{+}_{ewvt},\ W^{-}_{ewvt},\ W^{tot}_{ewvt} \in \mathbb{R}^{+}, \quad \forall e \in E, w \in WA_e, v \in V, t \in T \tag{3.17}$$

3.4. Objective function

The Objective Function (3.18) of the model minimizes the costs of the energy transition from now to 2050. For all asset types, fixed costs and costs representing technological development for building such assets are included, as well as variable costs for operating such assets, as explained in Section 3.2.

$$\begin{aligned}
\min \sum_{t \in T} \sum_{e \in E} \Bigg\{ & \sum_{d \in DA_e} \Bigg[\sum_{u \in U} B_{dut}(c_{dut} + o^{fix}_{dut}) + \sum_{u \in DU} o^{var}_{dut} \cdot F_{edut} \Bigg] \\
& + \sum_{v \in V} \Bigg[\sum_{s \in SA_e} (B_{svt}(c_{svt} + o^{fix}_{svt}) + o^{var}_{svt} \cdot S_{esvt}) \\
& \qquad + \sum_{m \in MA_e} (B_{mvt}(c_{mvt} + o^{fix}_{mvt}) + o^{var}_{mvt} \cdot M_{emvt}) \\
& \qquad + \sum_{w \in WA_e} (B_{wvt}(c_{wvt} + o^{fix}_{wvt}) + o^{+,var}_{wvt} \cdot W^{+}_{ewvt} + o^{-,var}_{wvt} \cdot W^{-}_{ewvt}) \Bigg] \Bigg\}
\end{aligned} \tag{3.18}$$

The complete formulation of the resulting Mixed Integer Linear Program (MILP) of the MES model can be found in Appendix B.1.

4 Greedy algorithm

For the implementation of column generation, a solution for the initial columns is needed, as is explained in the next chapter. Such a starting solution contains a value for each variable in the original model. To obtain such a starting solution in a short amount of time, a greedy algorithm is designed. Greedy algorithms are problem specific algorithms, that generate a solution quickly by making locally optimal choices at each step. Therefore, the solution generated by a greedy algorithm is not necessarily the optimal solution, and might not even be a feasible solution. However, this is not a problem since an initial solution for column generation does not need to be feasible. By adding an extra cost of a high number to the objective function value of an infeasible solution, it can still be used as an initial solution.

In this chapter, a specific greedy algorithm is designed for the optimization problem in the model as described in Chapter 3. For the case study used in this research, this algorithm is always able to generate a feasible solution, which can be used by column generation to start with. First, the general idea of this algorithm is given in Section 4.1. Then, a more detailed explanation of the greedy algorithm is given in Section 4.2, including the pseudocode of the algorithm. In Sections 4.3 and 4.4, examples of two complex steps in the algorithm are given, to provide a further understanding of the details of the algorithm. The numbers used in these calculations are taken from the data of the case study, which is later described in Section 6.1, such that the calculations are realistic examples.

4.1. General idea

The general idea of the greedy algorithm designed in this thesis is that it solves the optimization problem per time step, starting with the first time step. In each time step, all possible ways in which energy can be obtained in each node, are determined. The types of assets that are used to obtain energy in a certain way, and the order in which they are used, is hereafter called an 'energy path'. Thus, each energy path describes how energy of a certain carrier can be obtained in a certain node. The price of each energy path per unit of obtained energy is determined, and the capacity of this energy path is determined. Then, the cheapest energy path is chosen and added to the final solution. This changes the costs and capacities of the other energy paths, thus all energy paths are recalculated, and again the cheapest energy path is selected, etc. This process continues until the remaining demand of energy is zero in all nodes and in all time steps.

4.2. Detailed explanation of the greedy algorithm

In this section, all steps of the algorithm are elaborated upon. The steps for the first time period are first elaborated upon. At the end, an explanation of the additional steps for the following time periods is given. The pseudocode of the algorithm is given at the end of the section.

The algorithm starts with determining all possible energy paths for the first time step, specifically by determining the energy paths that consist solely of obtaining energy from supply assets directly in each node. Thus, after this step, all energy paths consist of one type of supply asset. For example, as shown in Figure 4.1, gas can be obtained in node v and wind energy can be obtained in node v'. The costs and capacities of the energy paths depend on the capacities of the supply assets, and are generally calculated in the following way. The cost of a supply asset in an energy path is equal to the costs per PJ energy. It is calculated by dividing the total cost of one asset by the capacity of one asset. In this greedy algorithm, only fixed costs for the investment in an asset are considered, and variable costs of using the assets are not included. This decision was made since variable costs are also not included in the case study of Eindhoven, which is described later in Section 6.1. However, with some small alterations, the variable costs can be taken into account in the greedy algorithm. The capacity of a supply asset in an energy path is equal to the capacity of one asset, multiplied with the number of assets that can be invested in at that node during that time period. A specific example of these calculations can be found in Section 4.3.

After this first step, it is calculated how the energy supply from the first step can be used to obtain energy in other nodes via distribution assets, or to obtain other energy carriers via converter assets. For example, as shown in Figure 4.2, heat can be obtained by converting gas in node v with a CHP, or by distributing obtained electricity from node v' to node v and converting it using a HP. This results in two different energy paths for obtaining heat in node v', both using different assets and a different number of assets, and both with a different cost and capacity. The number of assets that is used in an energy path is hereafter called the ‘length of the energy path’.

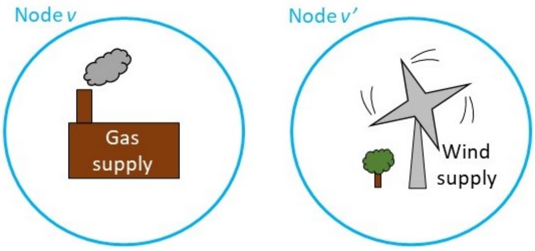


Figure 4.1: An example of two energy paths consisting solely of one supply asset.

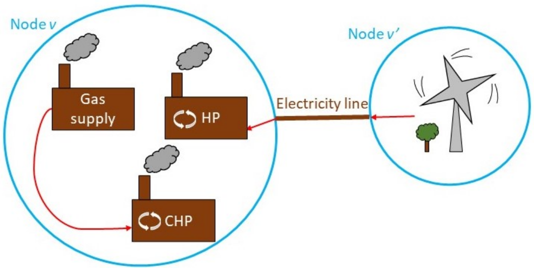


Figure 4.2: An example of two energy paths for obtaining heat in node v.

Calculating the cost and capacity for these energy paths is more complex, since efficiencies/loss factors and previous costs and capacities need to be taken into account. More specifically, the capacity of an energy path is determined recursively. It is calculated using the capacity of the last asset in the path and the capacity of the energy path excluding the last asset. The formulas for calculating the cost and capacity of an energy path are the following:

$$\begin{aligned}
\text{cost of an energy path [M€/PJ]} &= \sum_{\text{all assets in energy path}} \frac{\text{cost of one asset}}{\text{actual capacity* of one asset}}\\
\text{capacity of an energy path [PJ]} &= \min(\text{actual capacity* of last asset in energy path,}\\
&\quad \text{actual capacity* of energy path excluding last asset}\\
&\quad \cdot\ \text{efficiency/loss factor of last asset})\\
\text{*actual capacity of one asset [PJ]} &= \text{capacity of one asset}\ \cdot\ \text{efficiency/loss factor of the asset}
\end{aligned}$$

A step-by-step example of these calculations is given in Section 4.3 as well, which also explains the logic of these formulas.

In this scenario, there are two energy paths to obtain heat in node v. The second calculated energy path, which has a shorter length, is cheaper. However, this does not always have to be the case. An energy path consisting of three cheap assets can be cheaper than an energy path consisting of two expensive assets. It is also important to note that because of previous choices in the greedy algorithm, this energy path might not be available anymore at some point, so then the heat would be supplied through the longer, more expensive energy path. Through this example, it is obvious that the greedy algorithm does not necessarily generate the optimal solution.

Thus, the algorithm starts each iteration by determining possible energy paths consisting solely of one supply asset. The maximum total number of such possible energy paths is equal to $|SA|$. Then, the algorithm adds one distribution or one conversion asset to each of those paths, if possible. And it continues to add one distribution or one conversion asset, until paths of maximum length have been created. Theoretically, energy can be supplied in a certain node, and transported via all the other nodes, and converted by at most $|MA|-1$ different energy converters, thus via $1+(|V|-1)+(|MA|-1)$ different assets. However, calculating all energy paths of length 1 up to $|V|+|MA|-1$ is very time consuming, especially when applied to a case with many nodes. More specifically, the maximum total number of possible energy paths, at any point in the algorithm, is equal to $|SA| \cdot ((|V|-1)+(|MA|-1))!$. For a real case, such as the 7 nodes case that is described later in Section 6.1, this results in a maximum total of 80,640 possible energy paths, for all of which the costs and capacities have to be calculated, in every iteration of the algorithm. Therefore, it might be beneficial to set the maximum length of energy paths, indicated by parameter L, to a value smaller than $|V|+|MA|-1$. This reduces the computation time of the algorithm significantly, but could also result in a different solution, and may even result in the greedy algorithm not being able to find a feasible solution anymore. This should be taken into account when changing the parameter L.

After all possible energy paths of length at most L have been determined and their costs and capacities calculated, the cheapest energy path is selected by the greedy algorithm. This energy path is then implemented to the final solution. This means that the entire energy path is retraced, and the investment and usage of assets in this path is added to the final solution. Moreover, the remaining demand in the node that energy is obtained in, is reduced. To support understanding of this complex process, an example is given in Section 4.4.

Implementing this energy path changes the costs and capacities of the other energy paths. For example, if the cheapest energy path is the path consisting of supplying gas in node v and converting it to heat with a CHP, then this path is added to the final solution. If the maximum number of CHPs that can be placed in node v is 5, and 2 are built in the selected energy path, then there can now only be 3 more built. This changes the capacity of converting gas to heat with CHPs in node v. On the other hand, investments in two CHPs in node v have been made. The second CHP might not be used to convert gas to its full capacity, which means some gas can be converted on this CHP for free, without having to invest in more CHPs. Thus, this changes the costs of converting gas in node v. Moreover, some gas was already converted in the two CHPs to heat, but this also results in electricity, so this electricity is now available in node v for free, which also changes some energy paths. Therefore, all possible energy paths are again determined, and their costs and capacities are recalculated. The cycle then starts again, as the new cheapest energy path is chosen and implemented. This is continued until the remaining demand of energy in all nodes in the first time step is zero.

When all demand is met in the first time step, the algorithm moves on to the next time step, and the process of determining all possible energy paths, calculating their costs and capacities, determining the cheapest and implementing it to the final solution, is repeated again. However, the cost and capacity calculations in the following time steps are slightly different, since investments in assets in previous assets are still available. For example, if a CHP is already built in a previous time step in node v, then it is free to use it to convert gas in the current time step.

Moreover, after the first time step, the option of storage is included in the energy paths. Energy paths consisting of one type of storage asset are determined in the same step as energy paths consisting of one type of supply asset. A node v can obtain energy carrier e in time period t from storage, if there is still capacity left in previous time periods to supply energy of carrier e in node v. The previous time period t^* that has the cheapest supplied energy per unit is determined. The capacity of the created energy path is equal to the supply capacity in time period t^*. To determine the cost of the path, the best time period to invest in the storage assets needs to be determined. It is best make the investments in the first time period already, rather than in time period t^*. Though it may be more expensive to invest in a storage asset in an earlier time period (due to technological development and social discount rates), it is more beneficial to the overall solution that the storage asset is invested in during the first time period, such that it can be used in all time periods thereafter. And, if at some point during the algorithm, no more investments can be made in storage assets at that node in the first time period, then the investments are made in the second time period, etc. The cost of the resulting energy path is equal to the cost of the needed storage assets, when invested in during first time period, divided by the capacity of the path. Thereafter, distribution assets and conversion assets can be added to these paths to create longer possible energy paths, in the same way as to energy paths consisting of one type of supply asset.

Furthermore, the first time period t' in which a storage asset can be invested in at node v is determined. Though this is likely not the cheapest year to invest in a storage asset, it does mean that the asset can thereafter also be used for storing energy in different time periods.

The algorithm is done when the energy demand in all nodes is fulfilled for the last time period. The pseudocode of the resulting full greedy algorithm is given in Algorithm 1.

Algorithm 1 Greedy algorithm for finding a solution to the MES model

```
INPUT: Costs, capacities and efficiencies/loss factors of all assets
INPUT: Demand in every node, in every time period, of every energy carrier
INPUT: Parameter $L$ (maximum length of an energy path)
Initialize: final solution (0 distribution, supply, and conversion and 0 investments in assets)
for all $t \in T$ do
    Initialize: set of paths = $\emptyset$
    Initialize: demand in every node of every energy carrier in time period $t \in T$
    repeat
        Determine possible energy paths consisting of one type of supply asset only
        if current time period is not the first time period then
            Determine possible energy paths consisting of one type of storage asset only
        end if
        Compute costs and capacities of these energy paths
        repeat
            Add possible energy paths by adding one distribution or one conversion asset type to the
            existing paths in the set
            Compute costs and capacities of these added energy paths
        until All energy paths of length at most $L$ have been determined
        Determine the energy path with the cheapest cost per unit energy
        Add this energy path to the final solution
        Reduce demand in this node of this energy carrier by the capacity of the energy path
    until Demand in every node of every energy carrier is 0
end for
OUTPUT: A (possibly feasible) solution to the MES model.
```

4.3. Example calculating costs and capacities of energy paths

In this section, a step-by-step example is given of calculating the costs of energy paths, starting with energy paths consisting solely of supply assets.

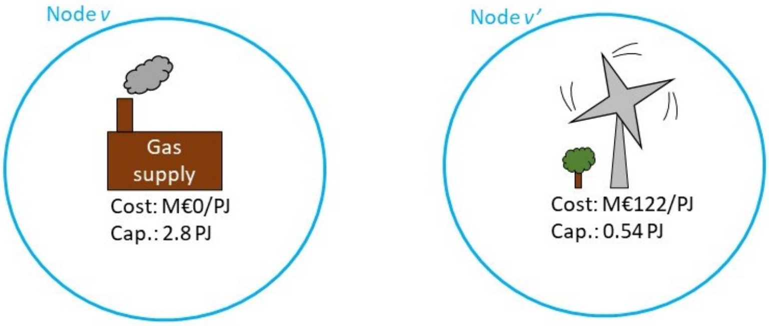


Figure 4.3: An example of two nodes and their supply assets, including the cost per PJ and the capacity of the assets.

Let there be 2.8 PJ gas already available in a node v. This gas is available for free, since the gas supply asset already exists. Thus, this energy path results in gas in node v with a cost of €0 and a capacity of 2.8 PJ. See also Figure 4.3. For a different node v', this might not be a possibility if there is no gas supply. Maybe wind energy supply assets can be built in this node v', with a limit of 10 assets. Then, the capacity of this possibility is 10 times the capacity of this asset. Assume that the capacity of one asset is 0.054 PJ, then the total capacity is 0.54 PJ. The cost per unit energy is equal to the cost of a wind energy supply asset, divided by the capacity of this asset. For example: €6.6 million / 0.054 PJ = M€122/PJ. See also Figure 4.3 for this example. This energy path results in electricity in node

v' with a cost of M€122/PJ and a capacity of 0.54 PJ. The same can be done for solar energy supply assets. By the end of this step, there are then some options available to obtain energy.

Using the example in Figure 4.4, the calculations for the costs and capacities of longer energy paths, containing distribution and conversion assets, can also be explained. Assume that the capacity of an electricity line is 0.179 PJ, such that one line can distribute 0.179 PJ from node v' to node v per time period. However, the electricity distribution has a loss factor of 2.5%, thus eventually only 0.179·0.975=0.175 PJ electricity will end up in node v when 0.179 PJ is distributed. This is called the 'actual capacity' of an asset. Assume that up to 5 times the capacity of one line can be built. The maximum total capacity of the line is then 0.175·5=0.895 PJ. However, the capacity of the wind supply asset at node v' was only 0.54 PJ, which results in 0.527 PJ when taking into account the loss factor of the electricity line. This is less than 0.895 PJ, so the maximum capacity of the energy path is 0.527 PJ. When assuming that the cost of an electricity line is €0.14 million, the cost of distributing energy on the energy line is M€0.14/0.175=M€0.8/PJ. Therefore, the total cost of the energy path is M€122 for building a wind supply asset plus M€0.8, which equals M€122.8/PJ. Thus, the resulting energy path for obtaining electricity in node v has a cost of M€122.8/PJ and a capacity of 0.527 PJ.

Similarly, taking into account an efficiency of 400% for the HP, it can be calculated that the cost of producing heat with a HP is for example M€6.9/PJ, and it can produce 0.54 PJ heat in total. Therefore, the energy path for obtaining heat in node v has a cost of 6.9M+122.8M=M€129.7/PJ, and a capacity of min(0.54, 0.527· 4)=0.54 PJ. In the same way, the energy path for obtaining heat in node v, consisting of taking gas from the supply in node v and converting it to heat using a CHP, can be calculated. Since the gas supply is free, the costs of this energy path are equal to the costs of the CHP, thus equal to M€35.2/PJ. Furthermore, considering the CHP has an efficiency of 54%, it could produce 2.8·0.54=1.512 PJ heat from the 2.8 PJ available gas. But the CHP only has a capacity of 0.383 PJ in total. Therefore, the energy path has a maximum capacity of 0.383 PJ.

As a result, two different energy paths for obtaining heat in node v have been found, both using different assets and a different number of assets, and both with a different cost and capacity.

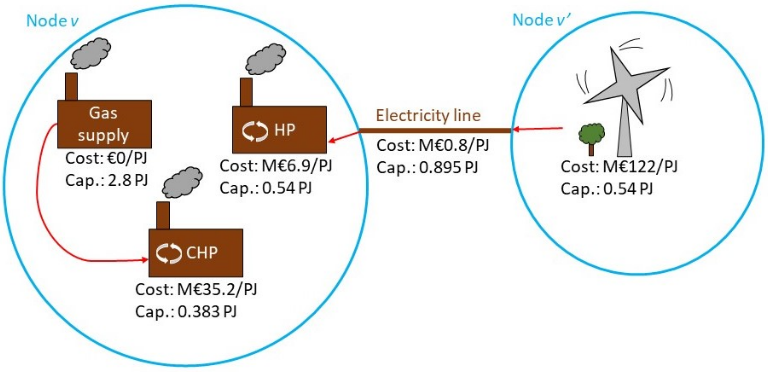


Figure 4.4: An example of two energy paths for obtaining heat in node v, including the cost per unit energy [PJ] and the capacity of the assets involved.

4.4. Example adding selected energy paths to final solution

In this section, the process of adding a selected energy path to the final solution is explained, again using the example from Figure 4.4. If the cheapest energy path is the path consisting of supplying gas in node v and converting it to heat with a CHP, then this path is added to the final solution. This is done in the following way. Assume that the heat demand in v is 0.1 PJ. The last step in this energy path is to convert gas into heat with a CHP. Assume that the CHP has an efficiency of 54%. Then, in order to obtain 0.1 PJ of heat, an amount of 0.1/0.54=0.185 PJ gas must be converted. Therefore, the amount of gas that is converted to heat with a CHP in node v in the first time step is 0.185 PJ, and this is added to the final solution. Furthermore, an investment of two CHPs, each with a capacity of 0.142 PJ, should be made in node v in the first time step, in order to be able to convert 0.185 PJ of gas. This is also added to the final solution. The step before the conversion in the energy path was to take this gas from the gas supply. Therefore, using 0.185 PJ from the gas supply in node v in the first time step is also added to the final solution. Lastly, the heat demand in node v is reduced by 0.1 PJ, which means that there is no heat demand left in this case. This concludes the process of adding a selected energy path to the final solution.

5 Column generation

This chapter discusses column generation, the main method used in this research to work toward the research aim. The aim was to find a better solution to the MES model within a certain amount of time. Column generation is based on splitting a complex problem into smaller subproblems, which are solved separately. This is expected to be well suited for generating good solutions quickly for the MES model, since it is a rather complex model combining two NP-hard problems. The general theory of column generation is first introduced in Section 5.1. It is explained how column generation can be applied to optimization problems, and how the master and pricing subproblem formulations can be derived, which are needed for column generation. At the end of this section, it is described how column generation can be applied to the MES model specifically. Column generation can be applied to the MES model in four different ways, meaning that it can be split up into subproblems in four different ways. In Section 5.2 - 5.5, the four different applications are further expanded upon, by giving the derivations of the master and pricing subproblem formulations for each of them.

5.1. General theory

Column generation (L. R. Ford & Fulkerson, 1958), or more specifically Dantzig-Wolfe decomposition (Dantzig & Wolfe, 1960), is a type of algorithm that can be used to find the optimal solution to large non-integer linear programs, that can be decomposed in a certain way. For example, it is often applied to multi-period problems, such as our MES model, that can be decomposed into subproblems per single time period. It can also be applied to integer linear programs such as the MES model, but then the additional use of a method such as branch-and-price is required to obtain an integer solution, as explained later.

The idea of column generation is that only a subset of the many possible values for variables is considered, such that it takes much less time to find the optimal solution. In each iteration of the algorithm, more possible values for variables are included. This process is continued until the optimal solution is found. The terms ‘column generation’ and ‘Dantzig-Wolfe decomposition’ are often used interchangeably, since applications of column generation mostly use a Dantzig-Wolfe decomposition. Dantzig-Wolfe decomposition is a specific algorithm that uses a column generation approach, which can be applied to linear programs of a certain structure. The Dantzig-Wolfe decomposition algorithm first decomposes the problem. This results in multiple subproblems (of a similar structure), called pricing subproblems, and one overarching problem, called the master problem, that combines the solutions to the subproblems.

The theory of the Dantzig-Wolfe decomposition algorithm is first introduced in Section 5.1.1. Then, the deduction of the master problem and pricing subproblem formulations are explained in Section 5.1.2. Furthermore, it is explained how solutions to integer linear programs can be found when using column generation in Section 5.1.3, namely by solving the integer master problem at the end. This theory is included because the MES model contains integer variables, thus an integer solution is desired. The described method is not guaranteed to give the optimal integer solution, but it may result in a better integer solution than the initial solution, as desired. Lastly, in Section 5.1.4, it is explained how this theory can be applied to the MES model. The MES model can be decomposed in four different ways, namely by formulating subproblems per time period, per energy carrier, per asset type, and per node. Thereafter, a more detailed explanation of the four applications is given in Sections 5.2 - 5.5. The entire derivation of the formulations of the master and pricing subproblems are given for each application.

5.1.1. Dantzig-Wolfe decomposition algorithm

In this section, the general idea of column generation is explained using the Capacitated Facility Location Problem (CFLP) as an example of a mixed-integer problem. The facility location problem was first introduced by Balinski (1965). This problem is used as an example here, since it is related to the MES model. In a basic formulation, the CFLP consists of a set of locations I where a facility producing goods with a certain capacity u_i can be opened, and a set of customer locations J with a certain demand of goods d_j. The goal is to determine at which locations a facility should be opened, and to which customers they should serve their goods, such that the costs of opening facilities and the transportation costs of goods are minimized. The cost of opening a facility $i \in I$ is denoted by parameter f_i, and the cost of transporting one unit of goods from a facility to a customer $j \in J$ is denoted by parameter c_{ij}. This problem is similar to our MES model, since part of the goal of this model is to find out how many of each type of asset should be invested in at each location, such that the opening and operation costs are minimized. However, note that our model is more complicated than a CFLP, but it is a good example problem to explain column generation. An instance of a CFLP is given in Figure 5.1. This instance contains three possible facility locations and four customers. The opening costs, transportation costs, customer demand and facility capacities are also given.

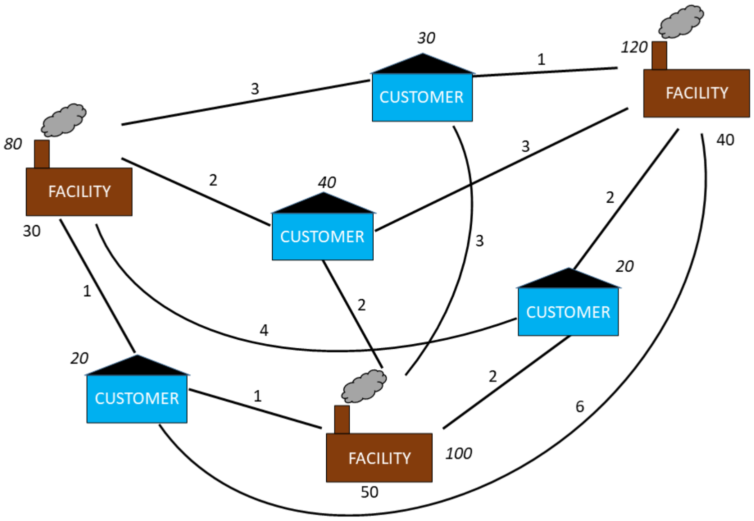


Figure 5.1: An instance of the CFLP with three possible facility locations and four customers. The values beneath the possible facility locations denote the costs of opening that facility. The values at the edges denote the costs per unit of transporting goods on those edges. Lastly, the italic values at the customers denote the demand, and the italic values at the facilities denote their capacity.

An optimal solution to this instance is shown in red in Figure 5.2. Two facilities out of three are opened in this solution. The total costs of this solution to the problem are 240. Note that there might be multiple solutions that have the same objective function value, so this might be just one of multiple optimal solutions.

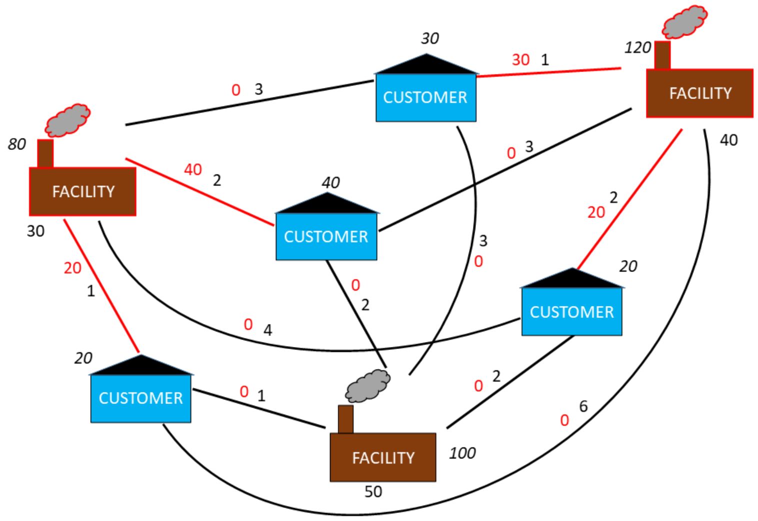


Figure 5.2: An optimal solution to the CFLP instance in Figure 5.1. The locations where a facility is opened are outlined in red. The red numbers on the edges denote the amount of goods that are transported on that edge. For clarity, the edges that are used are colored red. The total cost of this solution is 240.

The problem can be written as a mixed integer program:.

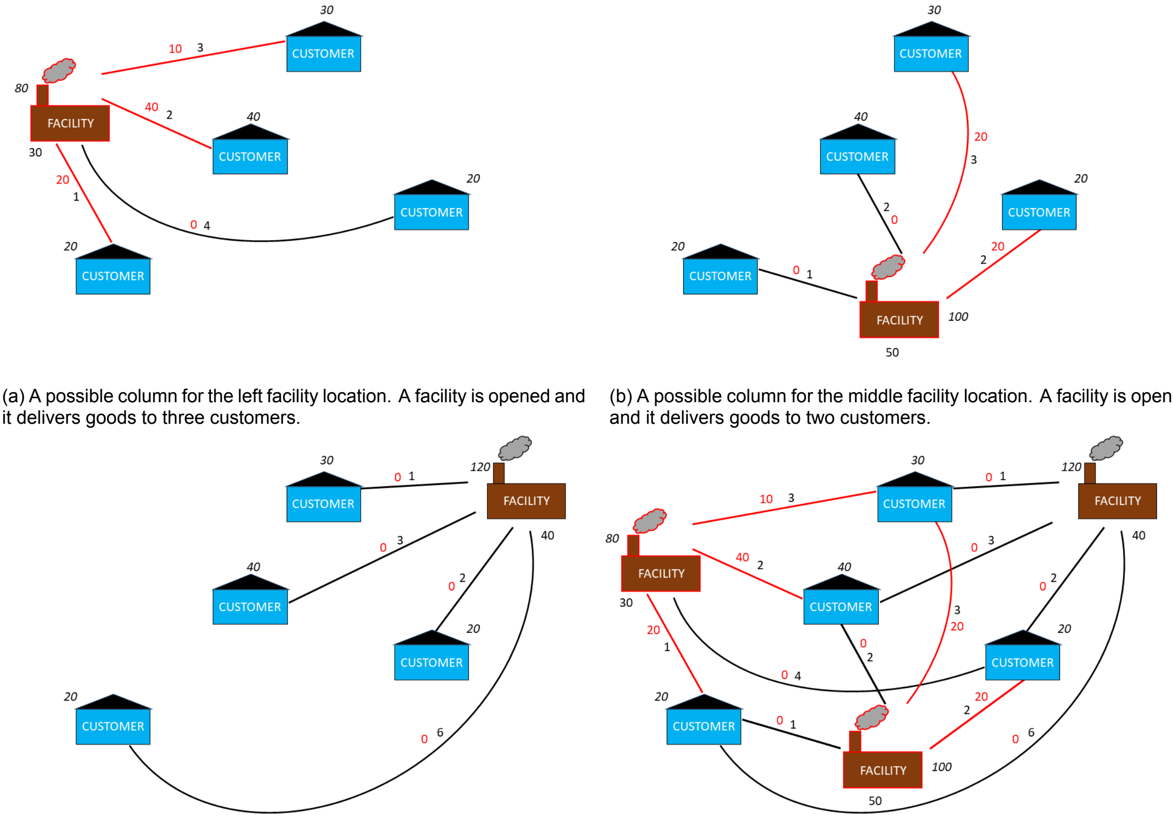
$$
\begin{aligned}
\min \sum_{i\in I}\sum_{j\in J} c_{ij} d_j Y_{ij} + \sum_{i\in I} f_i X_i & \\
\text{s.t.} \sum_{i\in I} Y_{ij} = 1, \quad & \forall j \in J \\
\sum_{j\in J} d_j Y_{ij} \le u_i X_i, \quad & \forall i \in I \\
Y_{ij} \ge 0, \quad & \forall i \in I,\, j \in J \\
X_i \in \{0,1\}, \quad & \forall i \in I
\end{aligned}
\tag{5.1}
$$

Here, binary variable X_i is 1 when facility $i \in I$ is opened and zero otherwise. Variable Y_{ij} can obtain a value between 0 and 1, and it indicates the fraction of the demand of customer $j \in J$ that is satisfied by facility $i \in I$, and thus transported on edge $\{ij\}$. The objective is to minimize the transportation costs of the goods and the opening costs of the facilities. The first constraint ensures that the demand of each location is met exactly. The second constraint ensures that the capacity of each facility is not exceeded.

The challenge of finding an optimal solution to such problems, is that there are many variables, that can have many different values. A possible solution is found by assigning a value to each variable. However, not all combinations of values of these variables result in a feasible solution. Only some possible solutions are actually feasible solutions, and only very few of these feasible solutions are optimal solutions. If one were to consider all possible solutions to a CFLP, in order to find the optimal solution, that means one would consider all possible combinations of values of all variables. The number of possible combinations of values of the variables depends on the number of facilities and customers. Assume the number of facilities is $|I| = n$ and the number of customers is $|J| = m$. Each facility $i \in I$ can be opened or not, represented by variable X_i. This results in a total of 2^n different possible combinations of values for variables X_i with $i \in I$. Moreover, each facility can receive any fraction of its demand from any facility. To illustrate how many different possibilities this results in, assume these fractions are not continuous, but are values between 0 and 1 with a step size of $1/10$. A customer can then receive any fraction from $\{0, 1/10, 2/10, \dots, 1\}$ from each facility, thus 11 different possible fractions. Therefore, there are 11^n different possible values for variable Y_{ij} for customer $j \in J$, thus $(11^n)^m = 11^{nm}$ different possible combinations of values for variables Y_{ij} with $i \in I, j \in J$. This results in a total of $2^n \cdot 11^{nm}$ different possible combinations of values of all variables, which equals more than $2.5 \cdot 10^{13}$ for the small example given in 5.1. Going through the different combinations until a feasible solution is found would already be very time consuming, let alone finding an optimal solution.

In order to apply column generation, the problem has to be decomposed into subproblems. In case of the CFLP, the problem can be decomposed into subproblems for each facility location. By solving a subproblem for a certain facility location, a ‘possible solution’ for that facility location can be found, which is called a ‘column’. Such a column represents a certain value for each variable in that subproblem. So in the case of the CFLP, a column contains the information on whether a facility is opened or not, and how much of the facility’s goods are delivered to which customers. An example of such a column for the left facility location is given in Figure 5.3a. In this column, a facility is opened at the location, and it serves a certain amount of goods to three customers. It is of course also an option that no facility is opened at a location, as shown in Figure 5.3c. In such a column, the facility is not opened and no goods are delivered to customers.

Every feasible solution to the CFLP consists of a combination of such columns, with one column for each facility. The combination of these columns should be such that the demand of every customer is met exactly. The master problem ensures that the combination of columns, with one column for each facility, results in such a feasible solution to the CFLP. An example of a solution to the master problem is given in Figure 5.3d, which consists of a combination of the columns given in Figure 5.3a, 5.3b and 5.3c.



(a) A possible column for the left facility location. A facility is opened and it delivers goods to three customers.

(b) A possible column for the middle facility location. A facility is opened and it delivers goods to two customers.

(c) A possible column for the right facility location. No facility is opened.

(d) A combination of the columns in Figures 5.3a, 5.3b and 5.3c that together form a feasible solution to the master problem. The total cost of this solution is 310.

Figure 5.3: A possible column for all three facilities, and solution formed by the combination of the columns.

The Dantzig-Wolfe decomposition algorithm starts with one column for each subproblem, so for each facility, that together form a feasible solution to the master problem. Thus, the columns given in Figure 5.3a, 5.3b and 5.3c could be used as initial columns for the master problem, since they form a feasible solution together. The relaxed master problem is solved with this set of columns, such that the values of the dual variables corresponding to the constraints in the master problem are found. Then, in every iteration, the pricing subproblem for each facility is solved. The objective function of the pricing subproblem is to minimize the reduced cost of the columns. This is the cost of a column, minus the dual variables multiplied with the relevant parts of their corresponding constraint, so this is what the dual values are used for. Therefore, the optimal solution of a pricing subproblem is the column that has the most potential to improve the objective function value of the master problem. This column is added to the set of columns in the master problem. This process is continued until the objective function value of all pricing subproblems is 0. When the reduced cost is 0, there is no column that could improve the objective function value of the master problem. This means that the optimal solution to the relaxed master problem is found.

The overall algorithm for general application of the Dantzig-Wolfe decomposition algorithm is given in Algorithm 2. Here, the problem is decomposed into subproblems for each $t \in T$. In case of the CFLP, T is the set of facility locations. In the next section, the derivations of the formulations of a master problem and the pricing subproblems is explained, based on the general formulation of a linear program that is decomposable.

Algorithm 2 Dantzig-Wolfe decomposition algorithm for a minimization problem

```
Decompose the problem into subproblems for each $t \in T$ and derive the master problem formulation
and pricing subproblems formulation
INPUT: Parameters of the problem
Initialize: one feasible column $k \in K_t$ for each subproblem $t \in T$ that together form a feasible
solution to the (relaxed) master problem
repeat
    Solve the relaxed master problem and obtain dual values
    for all $t \in T$ do
        Solve pricing subproblem for $t \in T$
        if objective function value of pricing subproblem for $t \in T$ < 0 then
            Add solution $k$ with associated parameters to $K_t$
        end if
    end for
until objective function value of pricing subproblem for each $t \in T$ is 0
OUTPUT: An optimal solution to the relaxed master problem.
```

5.1.2. Formulation master problem and pricing subproblems

Consider an integer linear program of the following structure:

$$\begin{aligned}
\min \quad & \sum_{t \in T} \sum_{j \in J_t} c_{tj} X_{tj} \\
\text{s.t.} \quad & \sum_{t \in T} \sum_{j \in J_t} a_{tij} X_{tj} = b_i, \quad \forall i \in I \\
& \sum_{j \in J_t} f_{tj} X_{tj} \geq g_t, \quad \forall t \in T \\
& X_{tj} \in \mathbb{Z}_+, \quad \forall t \in T,\ j \in J_t
\end{aligned} \tag{5.2}$$

Its objective is to minimize the costs involving a variable X_{tj}. This variable first has to satisfy a constraint for all $t \in T$ together. Furthermore, this variable has to satisfy a constraint for each $t \in T$. When a linear program has the structure of this Example problem (5.2), Dantzig-Wolfe decomposition can be applied, since it can be decomposed into subproblems (pricing subproblems) for every $t \in T$. Let K_t be the set of feasible solutions to such a subproblem. A 'column' $k \in K_t$ then describes a feasible solution to subproblem $t \in T$, meaning it assigns a value to each variable in that subproblem. Every variable X_{tj} in such a column for subproblem $t \in T$ has to satisfy the associated constraint $\sum_{j \in J_t} f_{tj} X_{tj} \geq g_t$, as well as the integrality constraint $X_{tj} \in \mathbb{Z}_+$. Therefore, these constraints are included in each pricing subproblem for $t \in T$. Every solution to the overall problem (master problem) consists of a combination of such columns. Therefore, the columns from every subproblem that are used in the overall problem should satisfy the first constraint together, thus it is included in the master problem. This is called the joint constraint, since it is a constraint that should be satisfied for all $t \in T$ together, because it sums over $t \in T$.

The objective function of the master problem is nearly the same as the objective function of the original formulation, with one addition. Namely, an extra variable λ_t^k is introduced for each column, to indicate whether a column is used or not. Variable λ_t^k is equal to 1 if column $k \in K_t$ is used, and 0 if it is not used in the solution to the master problem. In such a column, a certain value is assigned to each variable in that column. Therefore, for all $t \in T, j \in J_t$, the variable X_{tj} is replaced by $\sum_{k \in K_t} X_{tj}^k \lambda_t^k$ in the master problem formulation. Here, X_{tj}^k is a parameter that represents the integer value of variable X_{tj} in column $k \in K_t$, which is a feasible solution to the subproblem for $t \in T$. Furthermore, to ensure that exactly one column for each $t \in T$ is used in the solution, a convexity constraint is included in the master problem: $\sum_{k \in K_t} \lambda_t^k = 1, \ \forall t \in T$. Lastly, since the relaxed master problem is solved in each iteration in the Dantzig-Wolfe decomposition algorithm, a constraint is added that ensures that the value of λ_t^k is nonnegative. The total formulation of the relaxed master problem is then the following:

$$
\begin{aligned}
\min \quad & \sum_{t \in T} \sum_{j \in J_t} \sum_{k \in K_t} (c_{tj} X_{tj}^k) \lambda_t^k \\
\text{s.t.} \quad & \sum_{t \in T} \sum_{j \in J_t} \sum_{k \in K_t} (a_{tij} X_{tj}^k) \lambda_t^k = b_i, \quad \forall i \in I \quad (\pi_i) \\
& \sum_{k \in K_t} \lambda_t^k = 1, \quad \forall t \in T \quad (\mu_t) \\
& \lambda_t^k \geq 0, \quad \forall t \in T, \ k \in K_t
\end{aligned}
$$

Let π_i be the dual variable associated with the first constraint in the master problem, and let μ_t be the dual variable associated with the second constraint in the master problem. The pricing subproblem can be used to generate columns for the master problem, using these dual variables. Each column is a feasible solution to a subproblem, consisting of a value for each variable in that subproblem. What is contained in such a column can be determined first, by inspecting the master problem, after which the formulation of the pricing subproblems can be deduced from this column.

Such a column, for a $t \in T$ in this case, consist of a value for all variables regarding this $t \in T$. Thus, generally speaking, every entry in the column consists of the variables regarding this $t \in T$ from the objective function or the constraints of the master problem, and can be deduced by inspecting the master problem. The first entry of such a column is therefore the following, consisting of the terms containing variables regarding $t \in T$ in the objective:

$$\sum_{j \in J_t} c_{tj} X_{tj}.$$

Since the parameter c_{tj} is multiplied with the variable regarding $t \in T$, it is included in the column. The variable λ_t^k is not included in the column, since this variable indicates whether a column is used or not, but we are defining that very column now, so it should not be included. Similarly, the other entries of the column can be determined by inspecting the constraints in the master problem. The second entry is the vector **a**, that consists of $|I| = n$ entries, one for each $i \in I$. Each entry, for a certain $i \in I$, consists of the terms containing variables regarding $t \in T$ in the first constraint:

$$\sum_{j \in J_t} a_{tij} X_{tj}.$$

The corresponding dual variable of each entry is π_i. The parameter b_i is not included, since it is not multiplied with a variable regarding $t \in T$. The last entry of the column is the following:

$$\mathbf{e}_t.$$

This entry comes from the convexity constraint. The vector $\mathbf{e}_t$ denotes the vector of dimension $|T|$ with all zeros, except for the t^{th} entry, which is a 1. Note that entry is therefore the same in every column for $t \in T$, but it is necessary to include it for the pricing subproblem. The corresponding dual variable is μ_t.

A general column $k \in K_t$ for $t \in T$ is therefore the following:

$$\begin{pmatrix} \sum_{j \in J_t} c_{tj} X_{tj} \\ \mathbf{a} \\ \mathbf{e}_t \end{pmatrix}. \tag{5.3}$$

The column for $t \in T$ that has the most potential to improve the objective function value of the master problem is the column that is the optimal solution to the pricing subproblem for $t \in T$, since the objective function of the pricing subproblem is the reduced cost of such a column. This is the cost of a column, minus the dual variables multiplied with the coefficients of their corresponding constraints. The cost of column $k \in K_t$ is equal to the part of the objective function of the original formulation that contains variables regarding $t \in T$, so that is the first entry of the column. The second entry of the column consists of the terms containing variables regarding $t \in T$, so this entry, multiplied with its corresponding dual variables, is subtracted in the reduced cost of the column. And since this constraint should hold for every $i \in I$, $\sum_{i \in I} \pi_i a_{tij} X_{tj}$ is subtracted. Similarly, for the second constraint of the master problem, $\sum_{\tau \in T} (\mathbf{e}_t)_\tau \mu_\tau = \mu_t$ is subtracted in the reduced cost of a column. Here, $(\mathbf{e}_t)_\tau$ denotes the τ^{th} entry of vector $\mathbf{e}_t$. Thus, the objective function of the pricing subproblem is:

$$\min \sum_{j \in J_t} \left(c_{tj} - \sum_{i \in I} \pi_i a_{tij} \right) X_{tj} - \mu_t$$

As explained earlier, the constraint that is included in the pricing subproblems is the second constraint of the original formulation, since every column should satisfy this constraint. The total formulation of the pricing subproblem for $t \in T$ is then the following:

$$\begin{aligned} C^{t*} = \min \quad & \sum_{j \in J_t} \left(c_{tj} - \sum_{i \in I} \pi_i a_{tij} \right) X_{tj} - \mu_t \\ \text{s.t.} \quad & \sum_{j \in J_t} f_{tj} X_{tj} \geq g_t \\ & X_{tj} \in \mathbb{Z}_+, \quad \forall j \in J_t \end{aligned}$$

5.1.3. Obtaining an integer solution

As explained earlier, column generation solves the relaxed master problem in each iteration, thus a possibly non-integer solution is found by the algorithm. This is a lower bound for the optimal integer solution. If this lower bound is higher than the best bound found by the Gurobi solver in the same amount of time, then the aim of this research is achieved. Another way to achieve the research aim is by finding a better integer solution in the same amount of time, since the MES model is a mixed-integer program, containing integer variables that represent the number of investments in assets. Usually, a branch-and-price algorithm is used in combination with column generation, in order to find an integer solution.

Branch-and-price is a hybrid method of branch-and-bound and column generation, and works in the following way. The column generation algorithm is run fully, until it has converged, meaning the objective function value of all pricing subproblems is zero. Then, if the solution is not integer, which it most likely is not, a column $k \in K_t$ that is used in the final solution which contains a parameter X_{tj}^k that has a noninteger value R is identified. This parameter corresponds to a variable X_{tj} in the pricing subproblem for $t \in T$. This variable is branched on, meaning that the value of this variable constrained by adding the constraint $X_{tj} \geq \lceil R \rceil$ to the pricing subproblem. Existing columns in which the value of this parameter X_{tj}^k does not satisfy $X_{tj}^k \geq \lceil R \rceil$, are removed. Then, the column generation algorithm is run again. The same is done with the value of this variable constrained to $X_{tj} \leq \lfloor R \rfloor$. This process is continued, until an integer solution is found. This is the optimal integer solution.

Thus, running the branch-and-price algorithm involves running the column generation algorithm many times over, and an integer solution found when the branch-and-price algorithm is finished. Since the aim of this research is to find a better solution withing a certain amount of time, it would be better if an integer solution, which is better than the initial solution, could be found earlier. Therefore, another method to obtain an integer solution from column generation is used in this research, namely by solving the integer master problem at the end of the algorithm. This method can be applied whether column generation has converged or not, and it only slightly increases the computation time of the algorithm. It works in the following way. It is certain that if the initial columns form a feasible solution, then solving the integer master problem with these columns results in a feasible solution as well, namely this initial solution. After the set run time, the algorithm has found many other columns that are solutions to the pricing subproblems. This is true, regardless of whether the reduced cost of all pricing subproblems is zero and the algorithm is done, or not. Potentially, some combination of all these generated columns also forms a feasible solution to the integer master problem. If the objective function value of this solution to the integer master problem is lower than the objective function value of the solution of the original model after the same amount of time, then the goal of the research is achieved.

5.1.4. Applying column generation to MESDP

In this section, it is explained how column generation can be applied to the model in four different ways, after which a more in depth explanation of this application is given in the Sections 5.2 - 5.5. Since the MES model is multi-dimensional, the model can be decomposed into subproblems in four ways. First, the model can be decomposed into subproblems per time period. Each subproblem then represents the investment in and operation of all asset types, at all locations in one time period. This is similar to the approach that was performed by Stant (2019), which gave good results. However, Stant optimized the time periods separately, and took the optimal solution of one time period as input for the next time period. The advantage of column generation over this approach is that the overall problem is optimized, and not each time period separately. Similarly, the model can be decomposed into subproblems per energy carrier, per asset type, or per location.

5.2. Column generation per time period

In this section, we describe the first possible way in which column generation could be applied, namely per time period. First, a general explanation of this first application of column generation to the original MILP formulation is given in Section 5.2.1. Then, the formulations of the master problem and the pricing subproblems for each time period are derived in Sections 5.2.2 and 5.2.3, respectively.

5.2.1. General explanation

When decomposing the MES model into subproblems per time period, columns are generated for each time period $t \in T$. Such a column then describes the opening and operation of all assets $a \in A$ during one time period $t \in T$, for all energy carriers $e \in E$ and at all nodes $v \in V$ or edges $u \in U$. Any feasible solution then consists of a combination of columns, with exactly one column for each time period.

The joint constraints are the constraints that are marked in the original MILP formulation in Appendix B.2.1, namely the capacity constraints and the constraint defining W^{tot}_{ewvt}. These are the joined constraints, since these constraints sum over the time periods, or contain variables regarding different time periods. Note that only the constraints for time periods $t \in T \setminus \{t_0\}$ contain variables regarding different time periods. The constraint for time period t_0 only contains variables regarding time period t_0, thus it is not a joint constraint. To select columns in the column generation algorithm, a pricing subproblem for each time period is solved. Such a pricing subproblem contains the original objective function and the energy balance constraint from the original model. After introducing the notation for this column generation formulation, the master problem and pricing subproblem are derived.

Let K_t be the set of columns that represent a feasible solution to the subproblem for time period $t \in T$, for all energy carriers $e \in E$, assets $a \in A$ and nodes $v \in V$ or edges $u \in U$. Let variable λ^k_t be the binary decision variable associated with column $k \in K_t$ for time period $t \in T$. This decision variable is defined to be 1 if the column is used in the solution to the master problem, and 0 if it is not used. Then, parameter B^k_{avt} denotes the number of investments during time period $t \in T$ in column $k \in K_t$ in asset type $a \in A \setminus DA$ at node $v \in V$. Similarly, parameter B^k_{dut} denotes the number of investments during time period $t \in T$ in column $k \in K_t$ in distribution asset $d \in DA$ on edge $u \in U$. Furthermore, parameters F^k_{edut}, S^k_{esvt}, M^k_{emvt}, $W^{k,+}_{ewvt}$ and $W^{k,-}_{ewvt}$ indicate the amount of energy carrier $e \in E$ that is distributed, supplied, converted, stored and withdrawn from storage, respectively, by asset type $a \in A$ on directed edge $u \in DU$ or at node $v \in V$ in time period $t \in T$ in column $k \in K_t$. Lastly, parameter

$W_{ewvt}^{k,tot}$ indicates the total amount of energy carrier $e \in E$ that is stored at node $v \in V$ in asset $w \in WA_e$ at the start of time period $t \in T$ in column $k \in K_t$.

5.2.2. Master problem

The objective function of the master problem for column generation per time period is similar to the original objective function, but including the new binary decision variable λ_t^k for all $t \in T$ and $k \in K_t$. For example, for time period $t \in T$, variables B_{dut} are replaced by $\sum_{k \in K_t} \lambda_t^k B_{dut}^k$ and variables F_{edut} are replaced by $\sum_{k \in K_t} \lambda_t^k F_{edut}^k$. The objective function of the master problem is then as follows:

$$\begin{aligned} \min \quad & \sum_{t \in T} \sum_{k \in K_t} \sum_{e \in E} \Bigl\{ \sum_{d \in DA_e} \Bigl[\sum_{u \in U} \lambda_t^k B_{dut}^k (c_{dut} + o_{dut}^{fix}) + \sum_{u \in DU} o_{dut}^{var} \cdot \lambda_t^k F_{edut}^k \Bigr] \\ & + \sum_{v \in V} \Bigl[\sum_{s \in SA_e} (\lambda_t^k B_{svt}^k (c_{svt} + o_{svt}^{fix}) + o_{svt}^{var} \cdot \lambda_t^k S_{esvt}^k) \\ & + \sum_{m \in MA_e} (\lambda_t^k B_{mvt}^k (c_{mvt} + o_{mvt}^{fix}) + o_{mvt}^{var} \cdot \lambda_t^k M_{emvt}^k) \\ & + \sum_{w \in WA_e} (\lambda_t^k B_{wvt}^k (c_{wvt} + o_{wvt}^{fix}) + o_{wvt}^{+,var} \cdot \lambda_t^k W_{ewvt}^{k,+} + o_{wvt}^{-,var} \cdot \lambda_t^k W_{ewvt}^{k,-}) \Bigr] \Bigr\} \end{aligned}$$

As mentioned in Section 5.2.1, the joint constraints in the original MILP formulation are the capacity constraints and the constraint defining W_{ewvt}^{tot} for time periods $t \in T \setminus \{t_0\}$. The constraints regarding the distribution, supply and conversion capacity are rewritten for the master problem in the following way, again including the new binary decision variable λ_t^k:

$$\sum_{k \in K_t} \lambda_t^k F_{ed(v,v')t}^k \leq \gamma_d \sum_{i=t_0}^{t} \sum_{k \in K_i} \lambda_i^k B_{d\{v,v'\}i}^k, \quad \forall e \in E, d \in DA_e, v, v' \in V, t \in T \setminus \{t_0\}$$

$$\sum_{k \in K_t} \lambda_t^k S_{esvt}^k \leq \gamma_s \sum_{i=t_0}^{t} \sum_{k \in K_i} \lambda_i^k B_{svi}^k, \quad \forall e \in E, s \in SA_e, v \in V, t \in T \setminus \{t_0\}$$

$$\sum_{k \in K_t} \lambda_t^k M_{emvt}^k \leq \gamma_m \sum_{i=t_0}^{t} \sum_{k \in K_i} \lambda_i^k B_{mvi}^k, \quad \forall e \in E, m \in MA_e, v \in V, t \in T \setminus \{t_0\}$$

The dual variables corresponding to these constraints are denoted by π_{edut}^{DA}, π_{esvt}^{SA} and π_{emvt}^{MA}, respectively. The original constraints regarding the charging and discharging capacity of the storage assets and the constraint defining W_{ewvt}^{tot}, which are also joint constraints, are first rewritten to summation constraints, rather than recursive constraints. This makes the derivation of the pricing subproblem objective function easier. In the original model, the constraint for defining W_{ewvt}^{tot} is the following:

$$W_{ewvt}^{tot} = (W_{ewv(t-1)}^{tot} + W_{ewv(t-1)}^{+} \cdot \eta_w^{+} - W_{ewv(t-1)}^{-}) \cdot (1 - \eta_w^{sl}), \quad \forall e \in E, w \in WA_e, v \in V, t \in T \setminus \{t_0\} \tag{3.10}$$

This variable is used in the capacity constraint for W_{ewvt}^{+} and W_{ewvt}^{-} for $e \in E$, $w \in WA_e$, $v \in V$ and $t \in T$ in the original model in the following way:

$$W_{ewvt}^{+} \cdot \eta_w^{+} \leq \gamma_w \sum_{i=t_0}^{t} B_{wvi} - W_{ewvt}^{tot}, \quad \forall e \in E, w \in WA_e, v \in V, t \in T \tag{3.7}$$

$$W_{ewvt}^{-} \leq W_{ewvt}^{tot}, \quad \forall e \in E, w \in WA_e, v \in V, t \in T \tag{3.8}$$

In order to make it easier to find the pricing subproblem objective function later, we can rewrite the definition of W_{ewvt}^{tot} in the following way:

$$
\begin{aligned}
W_{ewvt_0}^{tot} &= 0 \\
W_{ewv(t_0+1)}^{tot} &= \big(W_{ewvt_0}^{tot} + W_{ewv(t_0)}^{+} \cdot \eta_w^+ - W_{ewvt_0}^{-} \big) \cdot (1 - \eta_w^{sl}) \\
&= \big(0 + W_{ewvt_0}^{+} \cdot \eta_w^+ - W_{ewvt_0}^{-} \big) \cdot (1 - \eta_w^{sl}) \\
W_{ewv(t_0+2)}^{tot} &= \big(W_{ewv(t_0+1)}^{tot} + W_{ewv(t_0+1)}^{+} \cdot \eta_w^+ - W_{ewv(t_0+1)}^{-} \big) \cdot (1 - \eta_w^{sl}) \\
&= \Big((W_{ewvt_0}^{+} \cdot \eta_w^+ - W_{ewvt_0}^{-}) \cdot (1 - \eta_w^{sl}) + W_{ewv(t_0+1)}^{+} \cdot \eta_w^+ - W_{ewv(t_0+1)}^{-} \Big) \cdot (1 - \eta_w^{sl}) \\
&= (W_{ewvt_0}^{+} \cdot \eta_w^+ - W_{ewvt_0}^{-}) \cdot (1 - \eta_w^{sl})^2 + (W_{ewv(t_0+1)}^{+} \cdot \eta_w^+ - W_{ewv(t_0+1)}^{-}) \cdot (1 - \eta_w^{sl}) \\
&\vdots \\
W_{ewvt}^{tot} &= (W_{ewv(t-1)}^{tot} + W_{ewv(t-1)}^{+} \cdot \eta_w^+ - W_{ewv(t-1)}^{-}) \cdot (1 - \eta_w^{sl}) \\
&= \sum_{i=t_0}^{t-1} (W_{ewvi}^{+} \cdot \eta_w^+ - W_{ewvi}^{-}) \cdot (1 - \eta_w^{sl})^{t-i}
\end{aligned}
$$

Therefore, Constraints (3.7) and (3.8) can be rewritten to the following such that Constraint (3.10) is redundant:

$$
\begin{aligned}
W_{ewvt}^{+} \cdot \eta_w^+ \le \gamma_w \sum_{i=t_0}^{t} B_{wvi} - \sum_{i=t_0}^{t-1} (W_{ewvi}^{+} \cdot \eta_w^+ - W_{ewvi}^{-}) \cdot (1 - \eta_w^{sl})^{t-i} \\
= B_{wvt} \cdot \gamma_w + \sum_{i=t_0}^{t-1} \Big(B_{wvi} \cdot \gamma_w - (W_{ewvi}^{+} \cdot \eta_w^+ - W_{ewvi}^{-}) \cdot (1 - \eta_w^{sl})^{t-i} \Big), \\
\forall e \in E, w \in WA_e, v \in V, t \in T \setminus \{t_0\}
\end{aligned}
\tag{5.4}
$$

$$
\begin{aligned}
W_{ewvt}^{-} \le \sum_{i=t_0}^{t-1} (W_{ewvi}^{+} \cdot \eta_w^+ - W_{ewvi}^{-}) \cdot (1 - \eta_w^{sl})^{t-i}, \\
\forall e \in E, w \in WA_e, v \in V, t \in T \setminus \{t_0\}
\end{aligned}
\tag{5.5}
$$

And for $t = t_0$ the following constraints hold:

$$W_{ewvt_0}^{+} \cdot \eta_w^+ \le B_{wvt_0} \cdot \gamma_w, \quad \forall e \in E, w \in WA_e, v \in V \tag{5.6}$$

$$W_{ewvt_0}^{-} = 0, \quad \forall e \in E, w \in WA_e, v \in V \tag{5.7}$$

Constraints (5.4) and (5.5) are then rewritten for the master problem in the following way, again including the new binary decision variable λ_t^k:

$$
\begin{aligned}
\sum_{k \in K_t} \lambda_t^k W_{ewvt}^{k,+} \cdot \eta_w^+ \le \gamma_w \sum_{k \in K_t} \lambda_t^k B_{wvt}^k + \sum_{i=t_0}^{t-1} \sum_{k \in K_i} \lambda_i^k \Big(B_{wvi}^k \cdot \gamma_w - (W_{ewvi}^{k,+} \cdot \eta_w^+ - W_{ewvi}^{k,-}) \cdot (1 - \eta_w^{sl})^{t-i} \Big), \\
\forall e \in E, w \in WA_e, v \in V, t \in T \setminus \{t_0\}
\end{aligned}
$$

$$
\begin{aligned}
\sum_{k \in K_t} \lambda_t^k W_{ewvt}^{k,-} \le \sum_{i=t_0}^{t-1} \sum_{k \in K_i} \lambda_i^k (W_{ewvi}^{k,+} \cdot \eta_w^+ - W_{ewvi}^{k,-}) \cdot (1 - \eta_w^{sl})^{t-i}, \\
\forall e \in E, w \in WA_e, v \in V, t \in T \setminus \{t_0\}
\end{aligned}
$$

The dual variables corresponding to these constraints are denoted by π_{ewvt}^{WA+} and π_{ewvt}^{WA-}, respectively. Constraints (5.6) and (5.7) are not included in the master problem, since they only contain variables regarding one time period, namely $t = t_0$, thus they are not joint constraints. These constraints are instead included in the pricing subproblem for $t = t_0$. Furthermore, the following constraint is included in the master problem, which ensures that exactly one column is chosen for each time period:

$$\sum_{k \in K_t} \lambda_t^k = 1, \quad \forall t \in T.$$

The dual variable corresponding to this constraint is denoted by μ_t. Lastly, a constraint is added to ensure that λ_t^k is indeed a binary variable:

$$\lambda_t^k \in \{0,1\}, \quad \forall t \in T, k \in K_t.$$

The complete formulation of the master problem can be found in Appendix B.2.2.

5.2.3. Pricing subproblems

As explained in Section 5.2.1, for this first application of column generation to the MILP, columns are generated per time period. By solving a pricing subproblem for each of these time periods, it is decided which columns are added in each iteration of the algorithm. In this section, the formulation for this pricing subproblem is derived.

By inspecting the master problem formulation in Appendix B.2.2, it can be found that the column for each $t \in T$ is the following:

$$\begin{pmatrix} \sum_{e \in E} \Bigg\{ \sum_{d \in DA_e} \Bigg[\sum_{u \in U} B_{dut}(c_{dut} + o_{dut}^{fix}) + \sum_{u \in DU} o_{dut}^{var} \cdot F_{edut} \Bigg] + \\ \sum_{v \in V} \Bigg[\sum_{s \in SA_e} (B_{svt}(c_{svt} + o_{svt}^{fix}) + o_{svt}^{var} \cdot S_{esvt}) + \\ \sum_{m \in MA_e} (B_{mvt}(c_{mvt} + o_{mvt}^{fix}) + o_{mvt}^{var} \cdot M_{emvt}) + \\ \sum_{w \in WA_e} (B_{wvt}(c_{wvt} + o_{wvt}^{fix}) + o_{wvt}^{+,var} \cdot W_{ewvt}^{+} + o_{wvt}^{-.var} \cdot W_{ewvt}^{-})) \Bigg] \Bigg\} \\ \mathbf{FCV} \\ \mathbf{SCV} \\ \mathbf{MCV} \\ \mathbf{WPCV} \\ \mathbf{WMCV} \\ \mathbf{e}_t \end{pmatrix}. \tag{5.8}$$

This column is not trivially derived by inspecting the master problem. Therefore, it is first explained what each entry is, after which the derivation of each entry is explained extensively. Thereafter, the derivation of the pricing subproblem is explained.

The first four lines of this vector are all part of the first entry of the vector. They represent the variable regarding time period $t \in T$ in the objective function of the master problem. The entries **FCV**, **SCV** and **MCV** are the vectors representing the distribution, supply and conversion capacity constraints, respectively. Vector **FCV** contains the following entry for all $e \in E, d \in DA_e, u = (v, v') \in DU$, for a certain $t \in T \setminus \{t_0\}$:

$$F_{edut} - B_{d\{v,v'\}t} \cdot \gamma_d.$$

The corresponding dual variable for each of these entries is π_{edut}^{DA}. Furthermore, the vector contains the following entry for all $e \in E, d \in DA_e, u = (v, v') \in DU, t' \in T : t' > t$:

$$-B_{d\{v,v'\}t} \cdot \gamma_d.$$

The corresponding dual variable for each of these entries is $\pi_{edut'}^{DA}$. Similarly, the vector **SCV** consists of the following entries, in a column for time period $t \in T$:

$$\begin{aligned} S_{esvt} - B_{svt} \cdot \gamma_s, &\quad \forall e \in E, s \in SA_e, v \in V \\ - B_{svt} \cdot \gamma_s, &\quad \forall e \in E, s \in SA_e, v \in V, t' \in T : t' > t \end{aligned}$$

The first entry is only included if $t > t_0$. The corresponding dual variables are π^{SA}_{esvt} and $\pi^{SA}_{esvt'}$. The vector **MCV** consists of the following entries for time period $t \in T$:

$$\begin{aligned} &M_{emvt} - B_{mvt} \cdot \gamma_m, \quad \forall e \in E, m \in MA_e, v \in V \\ &-B_{mvt} \cdot \gamma_m, \quad \forall e \in E, m \in MA_e, v \in V, t' \in T : t' > t \end{aligned}$$

The first entry is only included if $t > t_0$. The corresponding dual variables are π^{MA}_{emvt} and $\pi^{MA}_{emvt'}$. Furthermore, vector **WPCV** consists of the following entries:

$$\begin{aligned} &W^+_{ewvt} \cdot \eta^+_w - B_{wvt} \cdot \gamma_w, \quad \forall e \in E, w \in WA_e, v \in V \\ &B_{wvt} \cdot \gamma_w - (W^+_{ewvt} \cdot \eta^+_w - W^-_{ewvt}) \cdot (1 - \eta^{sl}_w)^{t'-t}, \quad \forall e \in E, w \in WA_e, v \in V, t' \in T : t' > t \end{aligned}$$

The first entry is only included if $t > t_0$. The corresponding dual variables are π^{WA+}_{ewvt} and $\pi^{WA+}_{ewvt'}$. The vector **WMCV** consists of the following entries:

$$\begin{aligned} &W^-_{ewvt}, \quad \forall e \in E, w \in WA_e, v \in V \\ &(W^+_{ewvt} \cdot \eta^+_w - W^-_{ewvt}) \cdot (1 - \eta^{sl}_w)^{t'-t}, \quad \forall e \in E, w \in WA_e, v \in V, t' \in T : t' > t \end{aligned}$$

The first entry is only included if $t > t_0$. The corresponding dual variables are π^{WA-}_{ewvt} and $\pi^{WA-}_{ewvt'}$. Lastly, vector $\mathbf{e}_t$ denotes the vector of dimension $|T|$ with all zeros, except for the t^{th} entry, which is a 1.

The derivation of each entry in Column (5.8) is now explained, starting with vectors **FCV**, **SCV** and **MCV**. The distribution, supply and conversion capacity constraints in the master problem, corresponding to dual variables π^{DA}_{edut}, π^{SA}_{esvt} and π^{MA}_{emvt}, respectively, contain variables regarding time period $t \in T$. For example, for time period $T \in T \setminus \{t_0\}$, the left hand side of the distribution capacity constraint in the master problem contains the following term, containing a parameter regarding t:

$$\sum_{k \in K_t} \lambda^k_t F^k_{edut}.$$

The right hand side of this constraint can be split up in the following way:

$$\begin{aligned} &\gamma_d \sum_{i=t_0}^{t} \sum_{k \in K_i} \lambda^k_i B^k_{d\{v,v'\}i} \\ &= \gamma_d \sum_{i=t_0}^{t-1} \sum_{k \in K_i} \lambda^k_i B^k_{d\{v,v'\}i} + \boxed{\gamma_d \sum_{k \in K_t} \lambda^k_t B^k_{d\{v,v'\}t}} \end{aligned}$$

The marked part contains a parameter regarding t. Therefore, the vector **FCV** in the column for time period $t \in T \setminus \{t_0\}$ contains the term $F_{edut} - B_{d\{v,v'\}t} \cdot \gamma_d$ for all $e \in E, d \in DA_e, u = (v, v') \in DU$. For all time periods $t' \in T$, $t' > t$, the right hand side of the distribution capacity constraint in the master problem contains the following term:

$$\begin{aligned} &\gamma_d \sum_{i=t_0}^{t'} \sum_{k \in K_i} \lambda^k_i B^k_{d\{v,v'\}i} \\ &= \gamma_d \sum_{\substack{i=t_0, \\ i \neq t}}^{t'} \sum_{k \in K_i} \lambda^k_i B^k_{d\{v,v'\}i} + \boxed{\gamma_d \sum_{k \in K_t} \lambda^k_t B^k_{d\{v,v'\}t}} \end{aligned}$$

Thus, for each time period $t' \in T$, $t' > t$, the right hand side of the distribution capacity constraint contains the marked term concerning time period $t \in T$. Therefore, the vector **FCV** in the column for time period $t \in T$ also contains the term $-B_{d\{v,v'\}t} \cdot \gamma_d$ for all $e \in E, d \in DA_e, u = (v, v') \in DU, t' \in$

$T : t' > t$. This results in the second entry in Column (5.8). The third and fourth entry in this column can be derived in a similar way. For the fifth and sixth entry, this is a bit more involved. The derivation of the fifth entry, vector **WPCV**, is now first explained. The charging capacity constraint in the master problem, corresponding to dual variable π_{ewvt}^{WA+} contains multiple variables regarding time period $t \in T$. For time period $t \in T \setminus \{t_0\}$, the left hand side of this constraint contains the following term, containing a variable regarding t:

$$\sum_{k \in K_t} \lambda_t^k W_{ewvt}^{k,+} \cdot \eta_w^+.$$

And for time period $t \in T$, the right hand side of this constraint contains the following term, containing a variable regarding t:

$$\gamma_w \sum_{k \in K_t} \lambda_t^k B_{wvt}^k.$$

Thus, vector **WPCV** in Column (5.8) for time period $t \in T \setminus \{t_0\}$ contains the term: $W_{ewvt}^+ \cdot \eta_w^+ - B_{wvt} \cdot \gamma_w$. For all time periods $t' \in T \setminus \{t_0\}$, $t' > t$, the right hand side of the charging capacity constraint in the master problem contains the following term:

$$\begin{aligned} &\sum_{i=t_0}^{t'-1} \sum_{k \in K_i} \lambda_i^k \left(B_{wvi}^k \cdot \gamma_w - (W_{ewvi}^{k,+} \cdot \eta_w^+ - W_{ewvi}^{k,-}) \cdot (1 - \eta_w^{sl})^{t'-i} \right) \\ = &\sum_{\substack{i=t_0, \\ i \neq t}}^{t'-1} \sum_{k \in K_i} \lambda_i^k \left(B_{wvi}^k \cdot \gamma_w - (W_{ewvi}^{k,+} \cdot \eta_w^+ - W_{ewvi}^{k,-}) \cdot (1 - \eta_w^{sl})^{t'-i} \right) \\ &+ \sum_{k \in K_t} \lambda_t^k \left(B_{wvt}^k \cdot \gamma_w - (W_{ewvt}^{k,+} \cdot \eta_w^+ - W_{ewvt}^{k,-}) \cdot (1 - \eta_w^{sl})^{t'-t} \right) \end{aligned}$$

Thus, for each time period $t' \in T, t' > t$, the right hand side of the distribution capacity constraint contains the marked term concerning time period $t \in T$. Therefore, vector **WPCV** in Column (5.8) for time period $t \in T$ also contains the term:

$$B_{wvt} \cdot \gamma_w - (W_{ewvt}^+ \cdot \eta_w^+ - W_{ewvt}^-) \cdot (1 - \eta_w^{sl})^{t'-t}, \quad \forall e \in E, w \in WA_e, v \in V, t' \in T : t' > t$$

Similarly, the vector **WMCV** can be derived.

Now, the objective function of the pricing subproblem for $t \in T$ can be derived, which is the reduced cost of such a column $k \in K_t$. This is the cost of the column, subtracted by the dual variables multiplied with the relevant parts of their corresponding constraints. The cost of column $k \in K_t$ is equal to the part of the objective function of the original formulation that contains variables regarding $t \in T$. Thus, the cost of column $k \in K_t$, for $t \in T$ is:

$$\begin{aligned} &\sum_{e \in E} \Bigg\{ \sum_{d \in DA_e} \Bigg[\sum_{u \in U} B_{dut}(c_{dut} + o_{dut}^{fix}) + \sum_{u \in DU} o_{dut}^{var} \cdot F_{edut} \Bigg] \\ &+ \sum_{v \in V} \Bigg[\sum_{s \in SA_e} (B_{svt}(c_{svt} + o_{svt}^{fix}) + o_{svt}^{var} \cdot S_{esvt}) \\ &+ \sum_{m \in MA_e} (B_{mvt}(c_{mvt} + o_{mvt}^{fix}) + o_{mvt}^{var} \cdot M_{emvt}) \\ &+ \sum_{w \in WA_e} (B_{wvt}(c_{wvt} + o_{wvt}^{fix}) + o_{wvt}^{+,var} \cdot W_{ewvt}^+ + o_{wvt}^{-.var} \cdot W_{ewvt}^-)) \Bigg] \Bigg\} \end{aligned}$$

The distribution, supply and conversion capacity constraints in the master problem, corresponding to dual variables π^{DA}_{edut}, π^{SA}_{esvt} and π^{MA}_{emvt}, respectively, each contain one variable regarding time period $t \in T$ on the left hand side. Furthermore, the distribution, supply and conversion capacity constraints in the master problem, corresponding to dual variables $\pi^{DA}_{edut'}$, $\pi^{SA}_{esvt'}$ and $\pi^{MA}_{emvt'}$, respectively, with $t' \geq t$, each contain several variables regarding time period $t \in T$ on the right hand side. Therefore, the following is subtracted in the reduced cost of column $k \in K_t$:

$$\sum_{e\in E}\Bigg\{\sum_{u=(v,v')\in DU}\ \sum_{d\in DA_e}\left(\pi^{DA}_{edut}\cdot F_{edut} - (B_{d\{v,v'\}t}\cdot\gamma_d)\cdot\sum_{i=t}^{t_{end}}\pi^{DA}_{edui}\right)$$
$$+\sum_{v\in V}\Bigg[\sum_{s\in SA_e}\left(\pi^{SA}_{esvt}\cdot S_{esvt} - (B_{svt}\cdot\gamma_s)\cdot\sum_{i=t}^{t_{end}}\pi^{SA}_{esvi}\right)$$
$$+\sum_{m\in MA_e}\left(\pi^{MA}_{emvt}\cdot M_{emvt} - (B_{mvt}\cdot\gamma_m)\cdot\sum_{i=t}^{t_{end}}\pi^{MA}_{emvi}\right)\Bigg]\Bigg\}$$

However, for time period t_0, a different term is subtracted, since the capacity constraints for time period t_0 are not included in the master problem. For time period t_0, the following term is added in the reduced cost of column $k \in K_{t_0}$:

$$\sum_{e\in E}\Bigg\{\sum_{u=(v,v')\in DU}\ \sum_{d\in DA_e}(B_{d\{v,v'\}t_0}\cdot\gamma_d)\cdot\sum_{i=t_0+1}^{t_{end}}\pi^{DA}_{edui}$$
$$+\sum_{v\in V}\Bigg[\sum_{s\in SA_e}(B_{svt_0}\cdot\gamma_s)\cdot\sum_{i=t_0+1}^{t_{end}}\pi^{SA}_{esvi}$$
$$+\sum_{m\in MA_e}(B_{mvt_0}\cdot\gamma_m)\cdot\sum_{i=t_0+1}^{t_{end}}\pi^{MA}_{emvi}\Bigg]\Bigg\}$$

The charging capacity constraint in the master problem, corresponding to dual variable π^{WA+}_{ewvt} also contains one variable regarding time period $t \in T$ on the left hand side, and one on the right hand side. Moreover, the charging capacity constraint in the master problem, corresponding to dual variable $\pi^{WA+}_{ewvt'}$, with $t' > t$, contains several variables regarding time period $t \in T$ on the right hand side. Therefore, the following is subtracted in the reduced cost of column $k \in K_t$ for time period $t \in T \setminus \{t_0\}$:

$$\sum_{e\in E}\sum_{v\in V}\sum_{w\in WA_e}\Bigg(\pi^{WA+}_{ewvt}\big(W^{+}_{ewvt}\cdot\eta^{+}_{w} - B_{wvt}\cdot\gamma_w\big)$$
$$-\sum_{i=t+1}^{t_{end}}\pi^{WA+}_{ewvi}\Big(B_{wvt}\cdot\gamma_w - (W^{+}_{ewvt}\cdot\eta^{+}_{w} - W^{-}_{ewvt})\cdot(1-\eta^{sl}_{w})^{i-t}\Big)\Bigg)$$

Similarly, it can be derived that the following term regarding the discharging capacity constraint is subtracted in the reduced cost of column $k \in K_t$ for time period $t \in T \setminus \{t_0\}$:

$$\sum_{e\in E}\sum_{v\in V}\sum_{w\in WA_e}\left(\pi^{WA-}_{ewvt}\cdot W^{-}_{ewvt} - \sum_{i=t+1}^{t_{end}}\pi^{WA-}_{ewvi}\cdot(W^{+}_{ewvt}\cdot\eta^{+}_{w} - W^{-}_{ewvt})\cdot(1-\eta^{sl}_{w})^{i-t}\right)$$

Note that in the objective of the pricing subproblem for $t = t_{end}$, the sums from time period $i = t+1$ to t_{end} disappear. Furthermore, since the charging and discharging capacity constraints for time period t_0 are not included in the master problem, a different term is added in the objective function of the pricing subproblem for time period t_0, namely:

$$\begin{aligned}
&\sum_{e\in E}\sum_{v\in V}\sum_{w\in WA_e}\bigg[\sum_{i=t_0+1}^{t_{end}} \pi^{WA+}_{ewvi}\Big(B_{wvt_0}\cdot\gamma_w-(W^{+}_{ewvt_0}\cdot\eta^{+}_{w}-W^{-}_{ewvt_0})\cdot(1-\eta^{sl}_{w})^{i-t_0}\Big)\\
&+\sum_{i=t_0+1}^{t_{end}}\pi^{WA-}_{ewvi}\cdot(W^{+}_{ewvt_0}\cdot\eta^{+}_{w}-W^{-}_{ewvt_0})\cdot(1-\eta^{sl}_{w})^{i-t_0}\bigg]
\end{aligned}$$

Lastly, for the constraint corresponding to dual variable μ_t in the master problem, $\sum_{i\in T}(\mathbf{e}_t)_i\mu_i=\mu_t$ is also subtracted in the reduced cost of column $k\in K_t$. Here, $(\mathbf{e}_t)_i$ denotes the i^{th} entry of vector $\mathbf{e}_t$. In total, the objective function of the pricing subproblem for $t\in T\setminus\{t_0\}$, which is the reduced cost of column $k\in K_t$, is:

$$\begin{aligned}
\min\quad &\sum_{e\in E}\bigg\{\sum_{d\in DA_e}\bigg[\sum_{u\in U}B_{dut}(c_{dut}+o^{fix}_{dut})+\sum_{u=(v,v')\in DU}\bigg(o^{var}_{dut}\cdot F_{edut}-\pi^{DA}_{edut}\cdot F_{edut}+(B_{d\{v,v'\}t}\cdot\gamma_d)\cdot\sum_{i=t}^{t_{end}}\pi^{DA}_{edui}\bigg)\\
&+\sum_{v\in V}\bigg[\sum_{s\in SA_e}\bigg(B_{svt}(c_{svt}+o^{fix}_{svt})+o^{var}_{svt}\cdot S_{esvt}-\pi^{SA}_{esvt}\cdot S_{esvt}+(B_{svt}\cdot\gamma_s)\cdot\sum_{i=t}^{t_{end}}\pi^{SA}_{esvi}\bigg)\\
&+\sum_{m\in MA_e}\bigg(B_{mvt}(c_{mvt}+o^{fix}_{mvt})+o^{var}_{mvt}\cdot M_{emvt}-\pi^{MA}_{emvt}\cdot M_{emvt}+(B_{mvt}\cdot\gamma_m)\cdot\sum_{i=t}^{t_{end}}\pi^{MA}_{emvi}\bigg)\\
&+\sum_{w\in WA_e}\bigg(B_{wvt}(c_{wvt}+o^{fix}_{wvt})+o^{+,var}_{wvt}\cdot W^{+}_{ewvt}+o^{-,var}_{wvt}\cdot W^{-}_{ewvt}-\pi^{WA+}_{ewvt}\cdot W^{+}_{ewvt}\cdot\eta^{+}_{w}\\
&+(B_{wvt}\cdot\gamma_w)\cdot\sum_{i=t}^{t_{end}}\pi^{WA+}_{ewvi}-\sum_{i=t+1}^{t_{end}}\pi^{WA+}_{ewvi}\cdot(W^{+}_{ewvt}\cdot\eta^{+}_{w}-W^{-}_{ewvt})\cdot(1-\eta^{sl}_{w})^{i-t}\\
&-\pi^{WA-}_{ewvt}\cdot W^{-}_{ewvt}+\sum_{i=t+1}^{t_{end}}\pi^{WA-}_{ewvi}\cdot(W^{+}_{ewvt}\cdot\eta^{+}_{w}-W^{-}_{ewvt})\cdot(1-\eta^{sl}_{w})^{i-t}\bigg)\bigg]\bigg\}\\
&-\mu_t
\end{aligned}$$

The objective function for time period t_0 is given on the next page.

For time period t_0, the objective function of the pricing subproblem is the following:

$$
\begin{aligned}
\min \quad & \sum_{e\in E}\Bigg\{ \sum_{d\in DA_e}\Bigg[\sum_{u\in U} B_{dut_0}(c_{dut_0}+o^{fix}_{dut_0}) + \sum_{u=(v,v')\in DU}\Bigg(o^{var}_{dut_0}\cdot F_{edut_0} + (B_{d\{v,v'\}t_0}\cdot\gamma_d)\cdot\sum_{i=t_0+1}^{t_{end}}\pi^{DA}_{edui}\Bigg) \\
& + \sum_{v\in V}\Bigg[\sum_{s\in SA_e}\Bigg(B_{svt_0}(c_{svt_0}+o^{fix}_{svt_0}) + o^{var}_{svt_0}\cdot S_{esvt_0} + (B_{svt_0}\cdot\gamma_s)\cdot\sum_{i=t_0+1}^{t_{end}}\pi^{SA}_{esvi}\Bigg) \\
& + \sum_{m\in MA_e}\Bigg(B_{mvt_0}(c_{mvt_0}+o^{fix}_{mvt_0}) + o^{var}_{mvt_0}\cdot M_{emvt_0} + (B_{mvt_0}\cdot\gamma_m)\cdot\sum_{i=t_0+1}^{t_{end}}\pi^{MA}_{emvi}\Bigg) \\
& + \sum_{w\in WA_e}\Bigg(B_{wvt_0}(c_{wvt_0}+o^{fix}_{wvt_0}) + o^{+,var}_{wvt_0}\cdot W^{+}_{ewvt_0} + o^{-,var}_{wvt_0}\cdot W^{-}_{ewvt_0} \\
& + \sum_{i=t_0+1}^{t_{end}}\pi^{WA+}_{ewvi}\Big(B_{wvt_0}\cdot\gamma_w - (W^{+}_{ewvt_0}\cdot\eta^{+}_{w} - W^{-}_{ewvt_0})\cdot(1-\eta^{sl}_{w})^{i-t_0}\Big) \\
& + \sum_{i=t_0+1}^{t_{end}}\pi^{WA-}_{ewvi}\cdot(W^{+}_{ewvt_0}\cdot\eta^{+}_{w} - W^{-}_{ewvt_0})\cdot(1-\eta^{sl}_{w})^{i-t_0}\Bigg)\Bigg]\Bigg]\Bigg\} \\
& - \mu_{t_0}
\end{aligned}
$$

However, a problem arises with the objective function for time periods $t \in T\backslash\{t_0\}$. The dual variables of type π are always negative. In the fifth line of the objective function, $-\pi^{WA+}_{wvt}$ is multiplied with variable $-W^{-}_{ewvt}$, and in the sixth line, π^{WA-}_{wvt} is multiplied with variable W^{+}_{ewvt}. Since there are no constraints in the pricing subproblem that create bounds on these variables, and it is a minimization problem, the solution space is unbounded. Therefore, bounding constraints need to be included on these variables in the pricing subproblem. This can be done by adding valid inequalities, concerning these variables. These constraints should only contain variables regarding one time period $t \in T$. Otherwise, they would be joint constraints, which cannot be included the pricing subproblem. A valid inequality for the variables regarding the charging of energy is that no more energy can be charged than the maximum amount of investments that could have been done in previous time periods, times the capacity. The same holds for the amount of energy that can be discharged from storage. This results in the following constraints:

$$
\begin{aligned}
W^{+}_{ewvt} &\leq (t-t_0)\cdot N\cdot\gamma_w, && \forall e\in E, w\in WA_e, v\in V \\
W^{-}_{ewvt} &\leq (t-t_0-1)\cdot N\cdot\gamma_w, && \forall e\in E, w\in WA_e, v\in V
\end{aligned}
$$

These constraints are included in the pricing subproblem for time period $t \in T \setminus \{t_0\}$. The other constraints included in the pricing subproblem for $t \in T$ are the constraints in the original MILP problem that are not joint constraints. These are the unmarked constraints in the MILP formulation in Appendix B.2.1. For time periods $t \in T$, these are the energy balance constraint and the constraints regarding the maximum number of investments that can be done. Moreover, constraints are included that ensure that B_{dut} and B_{avt} are nonnegative integer variables and F_{edut}, S_{esvt}, M_{emvt}, W^{+}_{ewvt} and W^{-}_{ewvt} are nonnegative continuous variables. All constraints are given on the next page.

Thus, the constraints in the pricing subproblem for $t \in T$ are:

$$\sum_{d \in DA_e} \sum_{v' \in V} F_{ed(v',v)t} \cdot \eta_{d\{v,v'\}} + \sum_{s \in SA_e} S_{esvt} \cdot \sigma_{st} + \sum_{\substack{e' \in E \\ e' \neq e}} \sum_{m \in MA_{e'}} M_{e'mvt} \cdot \eta_{me'e} + \sum_{w \in WA_e} W^-_{ewvt} \cdot \eta^-_w$$

$$\geq d_{evt} + \sum_{d \in DA_e} \sum_{v' \in V} F_{ed(v,v')t} + \sum_{m \in MA_e} M_{emvt} + \sum_{w \in WA_e} W^+_{ewvt}, \qquad \forall e \in E, v \in V$$

$B_{dut} \leq N,\ B_{dut} \in \mathbb{Z}^+, \quad \forall d \in DA, u \in U$

$B_{avt} \leq N,\ B_{avt} \in \mathbb{Z}^+, \quad \forall a \in A \setminus DA, v \in V$

$F_{edut} \in \mathbb{R}^+, \quad \forall e \in E, d \in DA_e, u \in DU$

$S_{esvt} \in \mathbb{R}^+, \quad \forall e \in E, s \in SA_e, v \in V$

$M_{emvt} \in \mathbb{R}^+, \quad \forall e \in E, m \in MA_e, v \in V$

$W^+_{ewvt},\ W^-_{ewvt} \in \mathbb{R}^+, \quad \forall e \in E, w \in WA_e, v \in V$

The complete MILP formulation of the pricing subproblem for $t \in T \setminus \{t_0\}$ can be found in Appendix B.2.3. The pricing subproblem for $t = t_0$ is the same, but the following constraints are added:

$$F_{ed(v,v')t_0} \leq B_{d(\{v,v'\}t_0} \cdot \gamma_d, \quad \forall e \in E, d \in DA_e, v, v' \in V \tag{5.9}$$

$$S_{esvt_0} \leq B_{svt_0} \cdot \gamma_s, \quad \forall e \in E, s \in SA_e, v \in V \tag{5.10}$$

$$M_{emvt_0} \leq B_{mvt_0} \cdot \gamma_m, \quad \forall e \in E, m \in MA_e, v \in V \tag{5.11}$$

$$W^+_{ewvt_0} \cdot \eta^+_w \leq B_{wvt_0} \cdot \gamma_w, \quad \forall e \in E, w \in WA_e, v \in V \tag{5.12}$$

$$W^-_{ewvt_0} = 0, \quad \forall e \in E, w \in WA_e, v \in V \tag{5.13}$$

5.3. Column generation per energy carrier

In this section, we describe the second possible way in which column generation could be applied, namely per energy carrier. First, a general explanation of this second application of column generation to the original MILP formulation is given in Section 5.3.1. Then, the formulations of the master problem and the pricing subproblems are derived in Sections 5.3.2 and 5.3.3, respectively.

5.3.1. General explanation

The second possible way in which column generation can be applied to the original MILP formulation is by noting that the problem can be decomposed into subproblems for each energy carrier $e \in E$. Therefore, column generation can be applied to this formulation, where columns are generated for each energy carrier $e \in E$. Such a column then describes the opening and operation of assets of all types $a \in A_e$ handling one energy carrier $e \in E$, for all time periods $t \in T$ and at all nodes $v \in V$ or all edges $u \in U$. Here, A_e denotes the set of all asset types handling energy carrier $e \in E$, so $A_e = DA_e \cup SA_e \cup MA_e \cup WA_e$. Any feasible solution then consists of a combination of columns, with exactly one column for each energy carrier.

The joint constraint is the constraint in the original MILP formulation in Appendix B.3.1 that contains a marked part, namely the energy balance constraint. This is the joined constraint, since this constraint sums over different energy carriers. Thus, this constraint is included in the master problem. To select columns in the column generation algorithm, a pricing subproblem for each energy carrier is solved. Such a pricing subproblem contains the original objective function and all constraints from the original MILP formulation, except the energy balance constraint. After introducing the notation for this column generation formulation, the master problem and pricing subproblems are derived.

Let K_e be the set of columns that represent a feasible solution for an energy carrier $e \in E$, for all time periods $t \in T$, assets $a \in A_e$ and nodes $v \in V$ or edges $u \in U$. Let variable λ^k_e be the binary decision variable associated with column $k \in K_e$ for energy carrier $e \in E$. This decision variable is defined to be 1 if the column is used in the solution to the master problem, and 0 if it is not used. Then, parameter B^k_{avt} denotes the number of investments in asset type $a \in A_e \setminus DA_e$ handling energy carrier $e \in E$ in column $k \in K_e$ at node $v \in V$ in time period $t \in T$. Similarly, parameter B^k_{dut} denotes the number of investments in distribution asset $d \in DA_e$ handling energy carrier $e \in E$ in column $k \in K_t$ on edge $u \in U$ in time period $t \in T$. Furthermore, parameters F^k_{edut}, S^k_{esvt}, M^k_{emvt}, $W^{k,+}_{ewvt}$ and $W^{k,-}_{ewvt}$ indicate the amount of energy carrier $e \in E$ that is distributed, supplied, converted, stored and withdrawn from storage, respectively, by asset type $d, s, m, w \in A_e$ on directed edge $u \in DU$ or at node $v \in V$ in column $k \in K_e$. Lastly, parameter $W_{ewvt}^{k,tot}$ indicates the total amount of energy carrier $e \in E$ that is stored at node $v \in V$ in asset $w \in WA_e$ at the start of time period $t \in T$ in column $k \in K_e$.

5.3.2. Master problem

The objective function of the master problem for column generation per energy carrier is similar to the original objective function, but including the new binary decision variable λ_e^k with $e \in E$, $k \in K_e$. For example, for asset $d \in DA_e$, variables B_{dut} are replaced by $\sum_{k \in K_e} \lambda_e^k B_{dut}^k$, and variables F_{edut} are replaced by $\sum_{k \in K_e} \lambda_e^k F_{edut}^k$. The objective function of the master problem is then as follows:

$$
\begin{aligned}
\min \quad & \sum_{t \in T} \sum_{e \in E} \sum_{k \in K_e} \Bigg\{ \sum_{d \in DA_e} \Bigg[\sum_{u \in U} \lambda_e^k B_{dut}^k (c_{dut} + o_{dut}^{fix}) + \sum_{u \in DU} o_{dut}^{var} \cdot \lambda_e^k F_{edut}^k \Bigg] \\
& + \sum_{v \in V} \Bigg[\sum_{s \in SA_e} (\lambda_e^k B_{svt}^k (c_{svt} + o_{svt}^{fix}) + o_{svt}^{var} \cdot \lambda_e^k S_{esvt}^k) \\
& + \sum_{m \in MA_e} (\lambda_e^k B_{mvt}^k (c_{mvt} + o_{mvt}^{fix}) + o_{mvt}^{var} \cdot \lambda_e^k M_{emvt}^k) \\
& + \sum_{w \in WA_e} (\lambda_e^k B_{wvt}^k (c_{wvt} + o_{wvt}^{fix}) + o_{wvt}^{+,var} \cdot \lambda_e^k W_{ewvt}^{k,+} + o_{wvt}^{+,var} \cdot \lambda_e^k W_{ewvt}^{k,-}) \Bigg] \Bigg\}
\end{aligned}
$$

As mentioned in Section 5.3.1, the joint constraint in the original MILP formulation is the energy balance constraint. This constraint is rewritten for the master problem in the following way, again including the new binary decision variable λ_e^k:

$$
\begin{aligned}
& \sum_{d \in DA_e} \sum_{v' \in V} \sum_{k \in K_e} \lambda_e^k F_{ed(v',v)t}^k \cdot \eta_{d\{v,v'\}} + \sum_{s \in SA_e} \sum_{k \in K_e} \lambda_e^k S_{esvt}^k \cdot \sigma_{st} \\
& + \sum_{\substack{e' \in E \\ e' \neq e}} \sum_{m \in MA_{e'}} \sum_{k \in K_{e'}} \lambda_{e'}^k M_{e'mvt}^k \cdot \eta_{me'e} + \sum_{w \in WA_e} \sum_{k \in K_e} \lambda_e^k W_{ewvt}^{k,-} \cdot \eta_w^- && (\pi_{evt}) \\
& \geq d_{evt} + \sum_{d \in DA_e} \sum_{v' \in V} \sum_{k \in K_e} \lambda_e^k F_{ed(v,v')t}^k + \sum_{m \in MA_e} \sum_{k \in K_e} \lambda_e^k M_{emvt}^k + \sum_{w \in WA_e} \sum_{k \in K_e} \lambda_e^k W_{ewvt}^{k,+}, \\
& \qquad \forall e \in E, v \in V, t \in T
\end{aligned}
$$

The dual variable corresponding to this constraint is denoted by π_{evt}. Furthermore, the following constraint is included in the master problem, which ensures that exactly one column is chosen for each energy carrier:

$$
\sum_{k \in K_e} \lambda_e^k = 1, \quad \forall e \in E
$$

The dual variable corresponding to this constraint is denoted by μ_e. Lastly, a constraint is added to ensure that λ_e^k is indeed a binary variable:

$$
\lambda_e^k \in \{0, 1\}, \quad \forall e \in E, k \in K_e
$$

The complete formulation of the master problem can be found in Appendix B.3.2.

5.3.3. Pricing subproblems

As explained in Section 5.3.1, for this second application of column generation to the MILP, columns are generated per energy carrier. By solving a pricing subproblem for each of these energy carriers, it is decided which columns are added in each iteration of the algorithm. In this section, the formulation for this pricing subproblem is derived.

By inspecting the master problem formulation in Appendix B.3.2, it can be found that the column for each $e \in E$ is the following:

$$\begin{pmatrix} \sum_{t \in T} \left\{ \sum_{d \in DA_e} \left[\sum_{u \in U} B_{dut}(c_{dut} + o_{dut}^{fix}) + \sum_{u \in DU} o_{dut}^{var} \cdot F_{edut} \right] + \right. \\ \sum_{v \in V} \left[\sum_{s \in SA_e} (B_{svt}(c_{svt} + o_{svt}^{fix}) + o_{svt}^{var} \cdot S_{esvt}) + \right. \\ \sum_{m \in MA_e} (B_{mvt}(c_{mvt} + o_{mvt}^{fix}) + o_{mvt}^{var} \cdot M_{emvt}) + \\ \left. \left. \sum_{w \in WA_e} (B_{wvt}(c_{wvt} + o_{wvt}^{fix}) + o_{wvt}^{+,var} \cdot W_{ewvt}^{+} + o_{wvt}^{-.var} \cdot W_{ewvt}^{-})) \right] \right\} \\ \mathbf{EBV} \\ \mathbf{MV} \\ \mathbf{e}_e \end{pmatrix}. \tag{5.14}$$

Note that the first four lines of this vector are all part of the first entry of the vector, and represents the variables regarding energy carrier $e \in E$ in the objective of the master problem. The second entry **EBV** is the vector representing the energy balance constraint. It contains the following entry for all $v \in V, t \in T$:

$$\sum_{d \in DA_e} \sum_{v' \in V} F_{ed(v',v)t} \cdot \eta_{d\{v,v'\}} + \sum_{s \in SA_e} S_{esvt} \cdot \sigma_{st} + \sum_{w \in WA_e} W_{ewvt}^{-} \cdot \eta_{w}^{-} -$$
$$\sum_{d \in DA_e} \sum_{v' \in V} F_{ed(v,v')t} - \sum_{m \in MA_e} M_{emvt} - \sum_{w \in WA_e} W_{ewvt}^{+}$$

The corresponding dual variable for each entry in this vector is π_{evt}. The third entry **MV** is the vector representing the conversion part in the energy balance constraint. It contains the following entry for all $e' \in E : e' \neq e, v \in V, t \in T$:

$$\sum_{m \in MA_e} M_{emvt} \cdot \eta_{mee'}.$$

The corresponding dual variable for each entry in this vector is $\pi_{e'vt}$. The last entry, vector $\mathbf{e}_e$, denotes the vector of dimension $|E|$ with all zeros, except for the eth entry, which is a 1. The corresponding dual variable for every entry in this vector is μ_e. The derivation of the vectors **EBV** and **MV** is not trivial. Therefore, before deducing the formulation of the pricing subproblem, this is first explained.

The energy balance constraint in the master problem, corresponding to dual variable π_{evt}, contains variables regarding energy carrier $e \in E$. More specifically, for energy carrier $e \in E$, the left hand side of the energy balance constraint in the master problem contains the following term, containing variables regarding $e \in E$:

$$\sum_{d \in DA_e} \sum_{v' \in V} F_{ed(v',v)t} \cdot \eta_{d\{v,v'\}} + \sum_{s \in SA_e} S_{esvt} \cdot \sigma_{st} + \sum_{w \in WA_e} W_{ewvt}^{-} \cdot \eta_{w}^{-}.$$

And the right hand side contains the following term containing variables regarding $e \in E$:

$$\sum_{d \in DA_e} \sum_{v' \in V} F_{ed(v,v')t} + \sum_{m \in MA_e} M_{emvt} + \sum_{w \in WA_e} W_{ewvt}^{+}.$$

Therefore, for energy carrier $e \in E$, these terms are included in the vector **EBV**, where the terms on the left hand side are added, and those on the right hand side are subtracted. Furthermore, the energy balance constraint for a certain energy carrier $e' \neq e, e' \in E$, also contains variables regarding energy carrier $e \in E$. For all $e' \in E, e' \neq e$, $v \in V$ and $t \in T$, the energy balance constraint in the master problem is the following:

$$\begin{aligned}
&\sum_{d \in DA_{e'}} \sum_{v' \in V} \sum_{k \in K_{e'}} \lambda_{e'}^{k} F_{e'd(v',v)t}^{k} \cdot \eta_{d\{v,v'\}} + \sum_{s \in SA_{e'}} \sum_{k \in K_{e'}} \lambda_{e'}^{k} S_{e'svt}^{k} \cdot \sigma_{st} \\
&+ \sum_{\substack{e'' \in E \\ e'' \neq e'}} \sum_{m \in MA_{e''}} \sum_{k \in K_{e''}} \lambda_{e''}^{k} M_{e''mvt}^{k} \cdot \eta_{me''e'} + \sum_{w \in WA_{e'}} \sum_{k \in K_{e'}} \lambda_{e'}^{k} W_{e'wvt}^{k,-} \cdot \eta_{w}^{-} \\
&\geq d_{e'vt} + \sum_{d \in DA_{e'}} \sum_{v' \in V} \sum_{k \in K_{e'}} \lambda_{e'}^{k} F_{e'd(v,v')t}^{k} + \sum_{m \in MA_{e'}} \sum_{k \in K_{e'}} \lambda_{e'}^{k} M_{e'mvt}^{k} + \sum_{w \in WA_{e'}} \sum_{k \in K_{e'}} \lambda_{e'}^{k} W_{e'wvt}^{k,+} \qquad (\pi_{e'vt})
\end{aligned}$$

The left hand side of this constraint contains the following term:

$$\begin{aligned}
&\sum_{\substack{e'' \in E \\ e'' \neq e'}} \sum_{m \in MA_{e''}} \sum_{k \in K_m} \lambda_{m}^{k} M_{e''mvt}^{k} \cdot \eta_{me''e'} \\
&= \sum_{\substack{e'' \in E \\ e'' = e}} \sum_{m \in MA_{e''}} \sum_{k \in K_m} \lambda_{m}^{k} M_{e''mvt}^{k} \cdot \eta_{me''e'} + \sum_{\substack{e'' \in E \\ e'' \neq e' \wedge e}} \sum_{m \in MA_{e''}} \sum_{k \in K_m} \lambda_{m}^{k} M_{e''mvt}^{k} \cdot \eta_{me''e'} \\
&= \sum_{m \in MA_e} \sum_{k \in K_m} \lambda_{m}^{k} M_{emvt}^{k} \cdot \eta_{mee'} + \sum_{\substack{e'' \in E \\ e'' \neq e' \wedge e}} \sum_{m \in MA_{e''}} \sum_{k \in K_m} \lambda_{m}^{k} M_{e''mvt}^{k} \cdot \eta_{me''e'}
\end{aligned}$$

Thus, for energy carrier $e' \in E$, $e' \neq e$, the left hand side of the energy balance constraint contains the marked term concerning energy carrier $e \in E$. Therefore, the column for energy carrier $e \in E$ contains the term: $\sum_{m \in MA} M_{emvt} \cdot \eta_{mee'}$ for every $e' \in E : e' \neq e, v \in V, t \in T$. This results in the vector **MV** in Column (5.14).

Now, the objective function of the pricing subproblem for $e \in E$ can be derived, which is the reduced cost of such a column $k \in K_e$. The cost of column $k \in K_e$ is equal to the part of the objective function of the original formulation that contains variables regarding $e \in E$. Thus, the cost of column $k \in K_e$, for $e \in E$ is:

$$\begin{aligned}
&\sum_{t \in T} \Bigl\{ \sum_{d \in DA_e} \Bigl(\sum_{u \in U} B_{dut}(c_{dut} + o_{dut}^{fix}) + \sum_{u \in DU} o_{dut}^{var} \cdot F_{edut} \Bigr) \\
&\quad + \sum_{v \in V} \Bigl[\sum_{s \in SA_e} \Bigl(B_{svt}(c_{svt} + o_{svt}^{fix}) + o_{svt}^{var} \cdot S_{esvt} \Bigr) \\
&\quad + \sum_{m \in MA_e} \Bigl(B_{mvt}(c_{mvt} + o_{mvt}^{fix}) + o_{mvt}^{var} \cdot M_{emvt} \Bigr) \\
&\quad + \sum_{w \in WA_e} \Bigl(B_{wvt}(c_{wvt} + o_{wvt}^{fix}) + o_{wvt}^{+,var} \cdot W_{ewvt}^{+} + o_{wvt}^{-,var} \cdot W_{ewvt}^{-} \Bigr) \Bigr] \Bigr\}
\end{aligned}$$

The second and third entry in Column (5.14) show the variables regarding $e \in E$ in the energy balance constraint in the master problem. The entries in these vectors are multiplied with their corresponding dual variables, and subtracted in the objective function. Therefore, in the objective function of the pricing subproblem for $e \in E$, the following term is subtracted:

$$\begin{aligned}
&\sum_{v\in V}\sum_{t\in T}\Bigg[\pi_{evt}\Bigg(\sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v',v)t}\cdot \eta_{d\{v,v'\}} + \sum_{s\in SA_e} S_{esvt}\cdot \sigma_{st} + \sum_{w\in WA_e} W^{-}_{ewvt}\cdot \eta^{-}_{w} \\
&\quad - \sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v,v')t} - \sum_{m\in MA_e} M_{emvt} - \sum_{w\in WA_e} W^{+}_{ewvt}\Bigg) \\
&\quad + \sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_e} \pi_{e'vt}\cdot M_{emvt}\cdot \eta_{mee'}\Bigg]
\end{aligned}$$

Lastly, for the constraint corresponding to dual variable μ_e in the master problem, $\sum_{i\in V}(\mathbf{e}_e)_i\mu_i = \mu_e$ is also subtracted in the reduced cost of column $k \in K_e$. Here, $(\mathbf{e}_e)_i$ denotes the i^{th} entry of vector $\mathbf{e}_e$. In total, the objective function of the pricing subproblem for $e \in E$ is:

$$\begin{aligned}
\min \quad &\sum_{t\in T}\Bigg\{\sum_{d\in DA_e}\Bigg(\sum_{u\in U} B_{dut}(c_{dut} + o^{fix}_{dut}) + \sum_{u\in DU} o^{var}_{dut}\cdot F_{edut}\Bigg) \\
&\quad + \sum_{v\in V}\Bigg[\sum_{s\in SA_e}\Bigg(B_{svt}(c_{svt} + o^{fix}_{svt}) + o^{var}_{svt}\cdot S_{esvt}\Bigg) \\
&\quad + \sum_{m\in MA_e}\Bigg(B_{mvt}(c_{mvt} + o^{fix}_{mvt}) + o^{var}_{mvt}\cdot M_{emvt}\Bigg) \\
&\quad + \sum_{w\in WA_e}\Bigg(B_{wvt}(c_{wvt} + o^{fix}_{wvt}) + o^{+,var}_{wvt}\cdot W^{+}_{ewvt} + o^{-,var}_{wvt}\cdot W^{-}_{ewvt}\Bigg) \\
&\quad - \pi_{evt}\Bigg(\sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v',v)t}\cdot \eta_{d\{v,v'\}} + \sum_{s\in SA_e} S_{esvt}\cdot \sigma_{st} + \sum_{w\in WA_e} W^{-}_{ewvt}\cdot \eta^{-}_{w} \\
&\quad - \sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v,v')t} - \sum_{m\in MA_e} M_{emvt} - \sum_{w\in WA_e} W^{+}_{ewvt}\Bigg) \\
&\quad - \sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_e} \pi_{e'vt}\cdot M_{emvt}\cdot \eta_{mee'}\Bigg]\Bigg\} \\
&\quad - \mu_e
\end{aligned}$$

As explained earlier, the constraints included in the pricing subproblem are all the constraints from the original MES model, except the energy balance constraint. The total formulation of the pricing subproblems can be found in Appendix B.3.3.

5.4. Column generation per asset type

In this section, we describe the third possible way in which column generation could be applied, namely per asset type. First, a general explanation of this third application of column generation to the original MILP formulation is given in Section 5.4.1. Then, the formulations of the master problem and the pricing subproblems are derived in Sections 5.4.2 and 5.4.3, respectively.

5.4.1. General explanation

The third possible way in which column generation can be applied to the original MILP formulation is by noting that the problem can be decomposed into subproblems for each asset type $a \in A$. More specifically, as can be seen in the MILP formulation in Appendix B.4.1, the problem can be split up into subproblems for each category of assets, namely distribution, supply, conversion and storage assets, corresponding to the four different colors in this MILP formulation. Therefore, column generation can be applied to this formulation, where columns are generated for each asset type $a \in A$. Note that the energy carrier is then also set for each column, since all asset types handle one type of energy carrier. Such a column then describes the opening and operation of assets of one asset type $a \in A_e$ for energy type $e \in E$, in all time periods $t \in T$ and at all nodes $v \in V$ or all edges $u \in U$. Here, A_e denotes the set of asset types handling energy carrier $e \in E$, so $A_e = DA_e \cup SA_e \cup MA_e \cup WA_e$. Any feasible solution then consists of a combination of columns, with exactly one column for each asset type.

The joint constraint is the energy balance constraint, since this constraint sums over the asset types. Thus, the energy balance constraint is included in the master problem. To select columns in the column generation algorithm, a pricing subproblem for each asset type is solved. Since all asset types belong to one of the four categories of asset types, a different pricing subproblem is constructed for each category of asset types. These four different pricing subproblems all contain part of the original objective function, one (or more) constraint(s) regarding the capacity and a constraint regarding the maximum number of investments. After introducing the notation for this column generation formulation, the master problem and pricing subproblems are derived.

Let K_a be the set of columns that represent a feasible solution for asset type a, for all energy carriers $e \in E$, nodes $v \in V$, edges $u \in U$ and time periods $t \in T$. Let variable λ_a^k be the binary decision variable associated with column $k \in K_a$ for asset type $a \in A$. This decision variable is defined to be 1 if the column is used in the solution to the master problem, and 0 if it is not used. Then, parameter B_{avt}^k denotes the number of investments in asset type $a \in A \setminus DA$ in column $k \in K_a$ at node $v \in V$ in time period $t \in T$. Similarly, parameter B_{dut}^k denotes the number of investments in distribution asset $d \in DA$ in column $k \in K_d$ on edge $u \in U$ in time period $t \in T$. Furthermore, parameters F_{edut}^k, S_{esvt}^k, M_{emvt}^k, $W_{ewvt}^{k,+}$ and $W_{ewvt}^{k,-}$ indicate the amount of energy carrier $e \in E$ that is distributed, supplied, converted, stored and withdrawn from storage, respectively, by asset type $d, s, m, w \in A$ on directed edge $u \in DU$ or at node $v \in V$ in time period $t \in T$ in column $k \in K_a$. Lastly, parameter $W_{ewvt}^{k,tot}$ indicates the total amount of energy carrier $e \in E$ that is stored at node $v \in V$ in asset $w \in WA_e$ at the start of time period $t \in T$ in column $k \in K_w$.

5.4.2. Master problem

The objective function of the master problem for column generation per asset type is similar to the original objective function, but including the new binary decision variable λ_a^k with $a \in A$, $k \in K_a$. For example, for asset $d \in DA_e$, variables B_{dut} are replaced by $\sum_{k \in K_d} \lambda_d^k B_{dut}^k$, and variables F_{edut} are replaced by $\sum_{k \in K_d} \lambda_d^k F_{edut}^k$. The objective function of the master problem is then as follows:

$$
\begin{aligned}
\min \quad & \sum_{t \in T} \sum_{e \in E} \Bigg\{ \sum_{d \in DA_e} \sum_{k \in K_d} \Bigg[\sum_{u \in U} \lambda_d^k B_{dut}^k (c_{dut} + o_{dut}^{fix}) + \sum_{u \in DU} o_{dut}^{var} \cdot \lambda_d^k F_{edut}^k \Bigg] \\
& + \sum_{v \in V} \Bigg[\sum_{s \in SA_e} \sum_{k \in K_s} (\lambda_s^k B_{svt}^k (c_{svt} + o_{svt}^{fix}) + o_{svt}^{var} \cdot \lambda_s^k S_{esvt}^k) \\
& + \sum_{m \in MA_e} \sum_{k \in K_m} (\lambda_m^k B_{mvt}^k (c_{mvt} + o_{mvt}^{fix}) + o_{mvt}^{var} \cdot \lambda_m^k M_{emvt}^k) \\
& + \sum_{w \in WA_e} \sum_{k \in K_w} (\lambda_w^k B_{wvt}^k (c_{wvt} + o_{wvt}^{fix}) + o_{wvt}^{+,var} \cdot \lambda_w^k W_{ewvt}^{k,+} + o_{wvt}^{-,var} \cdot \lambda_w^k W_{ewvt}^{k,-}) \Bigg] \Bigg\}
\end{aligned}
$$

As mentioned in Section 5.4.1, the joint constraint in the original MILP formulation is the energy balance constraint. This constraint is rewritten for the master problem in the following way, again including the new binary decision variable λ_a^k:

$$\begin{aligned}
&\sum_{d\in DA_e}\sum_{v'\in V}\sum_{k\in K_d}\lambda_d^k F_{ed(v',v)t}^k\cdot\eta_{d\{v,v'\}}+\sum_{s\in SA_e}\sum_{k\in K_s}\lambda_s^k S_{esvt}^k\cdot\sigma_{st}\\
&+\sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_{e'}}\sum_{k\in K_m}\lambda_m^k M_{e'mvt}^k\cdot\eta_{me'e}+\sum_{w\in WA_e}\sum_{k\in K_w}\lambda_w^k W_{ewvt}^{k,-}\cdot\eta_w^- \qquad (\pi_{evt})\\
&\geq d_{evt}+\sum_{d\in DA_e}\sum_{v'\in V}\sum_{k\in K_d}\lambda_d^k F_{ed(v,v')t}^k+\sum_{m\in MA_e}\sum_{k\in K_m}\lambda_m^k M_{emvt}^k+\sum_{w\in WA_e}\sum_{k\in K_w}\lambda_w^k W_{ewvt}^{k,+},\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall e\in E, v\in V, t\in T
\end{aligned}$$

The dual variable corresponding to this constraint is denoted by π_{evt}. Furthermore, the following constraint is included in the master problem, which ensures that exactly one column is chosen for each asset type:

$$\sum_{k\in K_a}\lambda_a^k=1,\quad \forall a\in A.$$

The dual variable corresponding to this constraint is denoted by μ_a. Lastly, a constraint is added to ensure that λ_a^k is indeed a binary variable:

$$\lambda_a^k\in\{0,1\},\quad \forall a\in A, k\in K_a.$$

The complete formulation of the master problem can be found in Appendix B.4.2.

5.4.3. Pricing subproblems

As explained in Section 5.4.1, for this third application of column generation to the MILP, columns are generated per asset type. By solving a pricing subproblem for each of these asset types, it is decided which columns are added in each iteration of the algorithm. However, as can be seen in the colored original MILP formulation in Appendix B.4.1, different constraints hold for different categories of asset types. Therefore, a separate pricing subproblem for each category of asset types is formulated. In this section, the formulation for each of these four pricing subproblems is derived.

Pricing subproblem for distribution assets

The pricing subproblem for each energy carrier $e\in E$ and distribution asset $d\in DA_e$ is first derived. By inspecting which parts of the master problem formulation in Appendix B.4.2 contain variables regarding d, it can be found that the column for each $e\in E$, $d\in DA_e$ is the following:

$$\begin{pmatrix}\sum_{t\in T}\left(\sum_{u\in U}B_{dut}(c_{dut}+o_{dut}^{fix})+\sum_{u\in DU}o_{dut}^{var}\cdot F_{edut}\right)\\ \mathbf{FV}\\ \mathbf{e}_d\end{pmatrix}.$$

Here, vector **FV** consists of the following entries for every $v\in V, t\in T$:

$$\sum_{v'\in V}F_{ed(v',v)t}\cdot\eta_{d\{v,v'\}}-\sum_{v'\in V}F_{ed(v,v')t}.$$

The corresponding dual variable is π_{evt}. Furthermore, $\mathbf{e}_d$ denotes the vector of dimension $|A|$ with all zeros, except for the d^{th} entry, which is a 1. The corresponding dual variable for every entry in this vector is μ_a.

The objective function of the pricing subproblem for $e\in E$, $d\in DA_e$ is the reduced cost of such a column $k\in K_d$. This is the cost of the column, subtracted by the dual variables multiplied with the relevant parts of their corresponding constraints. The cost of column $k\in K_d$ is equal to the part of the objective function of the original formulation that contains variables regarding $d \in DA$. Thus, the cost of column $k \in K_d$, for $e \in E$ and $d \in DA_e$ is:

$$\sum_{t \in T}\left(\sum_{u \in U} B_{dut}(c_{dut} + o_{dut}^{fix}) + \sum_{u \in DU} o_{dut}^{var} \cdot F_{edut}\right).$$

The energy balance constraint in the master problem, corresponding to dual variable π_{evt}, contains several variables regarding $d \in DA$. Therefore, the following is subtracted in the reduced cost of column $k \in K_d$:

$$\sum_{t \in T}\sum_{v \in V}\pi_{evt}\left(\sum_{v' \in V} F_{ed(v',v)t} \cdot \eta_{d\{v,v'\}} - \sum_{v' \in V} F_{ed(v,v')t}\right).$$

Lastly, for the constraint corresponding to dual variable μ_d in the master problem, $\sum_{a \in A}(\mathbf{e}_d)_a \mu_a = \mu_d$ is also subtracted in the reduced cost of column $k \in K_d$. Here, $(\mathbf{e}_d)_a$ denotes the ath entry of vector $\mathbf{e}_d$. In total, the objective function of the pricing subproblem for $e \in E$, $d \in DA_e$, which is the reduced cost of column $k \in K_d$, is:

$$\begin{aligned}\min \quad \sum_{t \in T}\Bigg[&\sum_{u \in U} B_{dut}(c_{dut} + o_{dut}^{fix}) + \sum_{u \in DU} o_{dut}^{var} \cdot F_{edut}\\ &- \sum_{v \in V}\pi_{evt}\left(\sum_{v' \in V} F_{ed(v',v)t} \cdot \eta_{d\{v,v'\}} - \sum_{v' \in V} F_{ed(v,v')t}\right)\Bigg] - \mu_d\end{aligned}$$

The constraints in the pricing subproblem for $e \in E$, $d \in DA_e$ are equal to the constraints in the original MILP problem regarding $e \in E$, $d \in DA_e$. These constraints are all marked green in the MILP formulation in Appendix B.4.1. They are the constraints regarding the distribution capacity and the maximum number of investments that can be done. Lastly, constraints are included that ensure that B_{dut} is a nonnegative integer variable and F_{edut} is a nonnegative continuous variable. Thus, the constraints in the pricing subproblem for $e \in E$, $d \in DA_e$ are:

$$\begin{aligned}F_{ed(v,v')t} &\leq \gamma_d \sum_{i=t_0}^{t} B_{d\{v,v'\}i}, \quad \forall v, v' \in V, t \in T\\ B_{dut} &\leq N, \quad \forall u \in U, t \in T\\ B_{dut} &\in \mathbb{Z}^+, \quad \forall u \in U, t \in T\\ F_{edut} &\in \mathbb{R}^+, \quad \forall u \in DU, t \in T\end{aligned}$$

The complete MILP formulation of the pricing subproblem for $e \in E$, $d \in DA_e$ can be found in Appendix B.4.3.

Pricing subproblem for supply assets

The pricing subproblem for each energy carrier $e \in E$ and supply asset $s \in SA_e$ is now derived. By inspecting which parts of the master problem formulation in Appendix B.4.2 contain variables regarding s, it can be found that the column for each $e \in E$, $s \in SA_e$ is the following:

$$\begin{pmatrix}\sum_{t \in T}\sum_{v \in V}\left(B_{svt}(c_{svt} + o_{svt}^{fix}) + o_{svt}^{var} \cdot S_{esvt}\right)\\ \mathbf{SV}\\ \mathbf{e}_s\end{pmatrix}. \tag{5.15}$$

Here, vector **SV** consists of the following entries for every $v \in V, t \in T$:

$$S_{esvt} \cdot \sigma_{st}.$$

The corresponding dual variable is π_{evt}. Furthermore, $\mathbf{e}_s$ denotes the vector of dimension $|A|$ with all zeros, except for the s^{th} entry, which is a 1. The corresponding dual variable for every entry in this vector is μ_a.

The objective function of the pricing subproblem for $e \in E$, $s \in SA_e$ is the reduced cost of such a column $k \in K_s$. We can derive this reduced cost in a similar way to deriving the reduced cost of a column for a distribution asset, by inspecting Column (5.15). Thus, we can deduce that the reduced cost such a column $k \in K_s$, for $e \in E$ and $s \in SA_e$ is the following:

$$\min \quad \sum_{t\in T}\sum_{v\in V}\left[B_{svt}(c_{svt} + o^{fix}_{svt}) + o^{var}_{svt}\cdot S_{esvt} - \pi_{evt}\cdot S_{esvt}\cdot \sigma_{st}\right] - \mu_s$$

The constraints in the pricing subproblem for $e \in E$, $s \in SA_e$ are equal to the constraints in the original MILP problem regarding $e \in E$, $s \in SA_e$. These constraints are all marked yellow in the MILP formulation in Appendix B.4.1. They are the constraints regarding the supply capacity and the maximum number of investments that can be done. Lastly, constraints are included that ensure that B_{svt} is a nonnegative integer variable and S_{esvt} is a nonnegative continuous variable. Thus, the constraints in the pricing subproblem for $e \in E$, $s \in SA_e$ are:

$$\begin{aligned} S_{esvt} &\leq \gamma_s \sum_{i=t_0}^{t} B_{svi}, \quad \forall v \in V, t \in T \\ B_{svt} &\leq N, \quad \forall v \in V, t \in T \\ B_{svt} &\in \mathbb{Z}^+, \quad \forall v \in V, t \in T \\ S_{esvt} &\in \mathbb{R}^+, \quad \forall v \in V, t \in T \end{aligned}$$

The complete MILP formulation of the pricing subproblem for $e \in E$, $s \in SA_e$ can be found in Appendix B.4.4.

Pricing subproblem for conversion assets
The pricing subproblem for each energy carrier $e \in E$ and conversion asset $m \in MA_e$ is now derived. By inspecting which parts of the master problem formulation in Appendix B.4.2 contain variables regarding m, it can be found that the column for each $e \in E$, $m \in MA_e$ is the following:

$$\begin{pmatrix} \sum_{t\in T}\sum_{v\in V}\left(B_{mvt}(c_{mvt} + o^{fix}_{mvt}) + o^{var}_{mvt}\cdot M_{emvt}\right) \\ \mathbf{MV1} \\ \mathbf{MV2} \\ \mathbf{e}_m \end{pmatrix}. \tag{5.16}$$

Here, vector **MV1** consists of the following entries for every $v \in V, t \in T$:

$$-M_{emvt}.$$

The corresponding dual variable is π_{evt}. And vector **MV2** consists of the following entries for every $e' \in E \setminus \{e\}, v \in V, t \in T$:

$$M_{emvt}\cdot \eta_{mee'}.$$

The corresponding dual variable is $\pi_{e'vt}$. Furthermore, $\mathbf{e}_m$ denotes the vector of dimension $|A|$ with all zeros, except for the m^{th} entry, which is a 1. The corresponding dual variable for every entry in this vector is μ_a. The derivation of the vector **MV2** in this column is not trivial. Therefore, before deducing the formulation of the pricing subproblem, this is explained first. This is similar to the derivation of the pricing subproblem for each energy carrier $e \in E$ as explained in Section 5.3.3, since the column in this section is also per energy carrier. For all energy carriers $e' \in E$, $e' \neq e$, the left hand side of the energy balance constraint in the master problem contains the following term:

$$\begin{aligned}
&\sum_{\substack{e''\in E\\ e''\neq e'}} \sum_{m\in MA_{e''}} \sum_{k\in K_m} \lambda_m^k M_{e''mvt}^k \cdot \eta_{me''e'}\\
&= \sum_{\substack{e''\in E\\ e''=e}} \sum_{m\in MA_{e''}} \sum_{k\in K_m} \lambda_m^k M_{e''mvt}^k \cdot \eta_{me''e'} + \sum_{\substack{e''\in E\\ e''\neq e'\wedge e}} \sum_{m\in MA_{e''}} \sum_{k\in K_m} \lambda_m^k M_{e''mvt}^k \cdot \eta_{me''e'}\\
&= \sum_{m\in MA_e} \sum_{k\in K_m} \lambda_m^k M_{emvt}^k \cdot \eta_{mee'} + \sum_{\substack{e''\in E\\ e''\neq e'\wedge e}} \sum_{m\in MA_{e''}} \sum_{k\in K_m} \lambda_m^k M_{e''mvt}^k \cdot \eta_{me''e'}
\end{aligned}$$

Thus, for each energy carrier $e' \in E$, $e' \neq e$, the left hand side of the energy balance constraint contains the marked term concerning energy carrier $e \in E$. Therefore, for energy carrier $e \in E$, the vector **MV2** in the column for $m \in MA_e$ contains the term: $\sum_{m\in MA} M_{emvt} \cdot \eta_{mee'}, \forall e' \in E : e' \neq e, v \in V, t \in T$. This results in Column (5.16).

Now, the objective function of the pricing subproblem for $e \in E$, $m \in MA_e$ can be derived, which is the reduced cost of such a column $k \in K_m$. This is the cost of the column, subtracted by the dual variables multiplied with the relevant parts of their corresponding constraints. The cost of column $k \in K_m$ is equal to the part of the objective function of the original formulation that contains variables regarding $m \in MA$. Thus, the cost of column $k \in K_m$, for $e \in E$ and $m \in MA_e$ is:

$$\sum_{t\in T} \sum_{v\in V} \Big(B_{mvt}(c_{mvt} + o_{mvt}^{fix}) + o_{mvt}^{var} \cdot M_{emvt} \Big).$$

In the reduced cost of column $k \in K_m$, for $e \in E$ and $m \in MA_e$, these costs are subtracted by the terms in vectors **MV1** and **MV2** in the column, multiplied with their corresponding dual variables. Therefore, the following is subtracted in the reduced cost of column $k \in K_m$:

$$\sum_{t\in T} \sum_{v\in V} \Bigg(-\pi_{evt} \cdot M_{emvt} + \sum_{\substack{e'\in E\\ e'\neq e}} \pi_{e'vt} \cdot M_{emvt} \cdot \eta_{mee'} \Bigg).$$

Lastly, for the constraint corresponding to dual variable μ_m in the master problem, $\sum_{a\in A} (\mathbf{e}_m)_a \mu_a = \mu_m$ is also subtracted in the reduced cost of column $k \in K_m$. Here, $(\mathbf{e}_m)_a$ denotes the a^{th} entry of vector $\mathbf{e}_m$. In total, the objective function of the pricing subproblem for $e \in E$, $m \in MA_e$, which is the reduced cost of column $k \in K_m$, is:

$$\min \quad \sum_{t\in T} \sum_{v\in V} \Bigg[B_{mvt}(c_{mvt} + o_{mvt}^{fix}) + o_{mvt}^{var} \cdot M_{emvt} + \pi_{evt} \cdot M_{emvt} - \sum_{\substack{e'\in E\\ e'\neq e}} \pi_{e'vt} \cdot M_{emvt} \cdot \eta_{mee'} \Bigg] - \mu_m$$

The constraints in the pricing subproblem for $e \in E$, $m \in MA_e$ are equal to the constraints in the original MILP problem regarding $e \in E$, $m \in MA_e$. These constraints are all marked red in the MILP formulation in Appendix B.4.1. They are the constraints regarding the conversion capacity and the maximum number of investments that can be done. Lastly, constraints are included that ensure that B_{mvt} is a nonnegative integer variable and M_{emvt} is a nonnegative continuous variable. Thus, the constraints in the pricing subproblem for $e \in E$, $m \in MA_e$ are:

$$\begin{aligned}
M_{emvt} &\leq \gamma_m \sum_{i=t_0}^{t} B_{mvi}, \quad && \forall v \in V, t \in T\\
B_{mvt} &\leq N, && \forall v \in V, t \in T\\
B_{mvt} &\in \mathbb{Z}^+, && \forall v \in V, t \in T\\
M_{emvt} &\in \mathbb{R}^+, && \forall v \in V, t \in T
\end{aligned}$$

The complete MILP formulation of the pricing subproblem for $e \in E$, $m \in MA_e$ can be found in Appendix B.4.5.

Pricing subproblem for storage assets
The pricing subproblem for each energy carrier $e \in E$ and storage asset $w \in WA_e$ is now derived. By inspecting which parts of the master problem formulation in Appendix B.4.2 contain variables regarding w, it can be found that the column for each $e \in E$, $w \in WA_e$ is the following:

$$\begin{pmatrix} \sum_{t\in T} \sum_{v\in V} \left(B_{wvt}(c_{wvt} + o_{wvt}^{fix}) + o_{wvt}^{+,var} \cdot W_{ewvt}^{+} + o_{wvt}^{-,var} \cdot W_{ewvt}^{-} \right) \\ \mathbf{WV} \\ \mathbf{e}_w \end{pmatrix}. \tag{5.17}$$

Here, vector **WV** consists of the following entries for every $v \in V, t \in T$:

$$W_{ewvt}^{-} \cdot \eta_w^{-} - W_{ewvt}^{+}.$$

The corresponding dual variable is π_{evt}. Furthermore, $\mathbf{e}_w$ denotes the vector of dimension $|A|$ with all zeros, except for the w^{th} entry, which is a 1. The corresponding dual variable for every entry in this vector is μ_a.

The objective function of the pricing subproblem for $e \in E$, $w \in WA_e$ is the reduced cost of such a column $k \in K_w$. We can derive this reduced cost in a similar way to deriving the reduced cost of a column for the other asset types, by inspecting Column (5.17). Thus, we can deduce that the reduced cost of such a column $k \in K_w$, for $e \in E$ and $w \in WA_e$ is the following:

$$\min \quad \sum_{t\in T} \sum_{v\in V} \left[B_{wvt}(c_{wvt} + o_{wvt}^{fix}) + o_{wvt}^{+,var} \cdot W_{ewvt}^{+} + o_{wvt}^{-,var} \cdot W_{ewvt}^{-} - \pi_{evt}\left(W_{ewvt}^{-} \cdot \eta_w^{-} - W_{ewvt}^{+} \right) \right] - \mu_w$$

The constraints in the pricing subproblem for $e \in E$, $w \in WA_e$ are equal to the constraints in the original MILP problem regarding $e \in E$, $w \in WA_e$. These constraints are all marked blue in the MILP formulation in Appendix B.4.1. They are the constraints regarding the charging and discharging capacity and the maximum number of investments that can be done. Furthermore, the constraints defining W_{ewvt}^{tot} should be included in the pricing subproblem. Lastly, constraints are included that ensure that B_{wvt} is a nonnegative integer variable and W_{ewvt}^{+}, W_{ewvt}^{-} and W_{ewvt}^{tot} are nonnegative continuous variables. Thus, the constraints in the pricing subproblem for $e \in E$, $w \in WA_e$ are:

$$\begin{aligned} W_{ewvt}^{+} \cdot \eta_w^{+} &\leq \gamma_w \sum_{i=t_0}^{t} B_{wvi} - W_{ewvt}^{tot}, \quad \forall v \in V, t \in T \\ W_{ewvt}^{-} &\leq W_{ewvt}^{tot}, \quad \forall v \in V, t \in T \\ W_{ewvt_0}^{tot} &= 0, \quad \forall v \in V \\ W_{ewvt}^{tot} &= (W_{ewv(t-1)}^{tot} + W_{ewv(t-1)}^{+} \cdot \eta_w^{+} - W_{ewv(t-1)}^{-}) \cdot (1 - \eta_w^{sl}), \quad \forall v \in V, t \in T \setminus \{t_0\} \\ B_{wvt} &\leq N, \quad \forall v \in V, t \in T \\ B_{wvt} &\in \mathbb{Z}^{+}, \quad \forall v \in V, t \in T \\ W_{ewvt}^{+}, W_{ewvt}^{-}, W_{ewvt}^{tot} &\in \mathbb{R}^{+}, \quad \forall v \in V, t \in T \end{aligned}$$

The complete MILP formulation of the pricing subproblem for $e \in E$, $w \in WA_e$ can be found in Appendix B.4.6.

5.5. Column generation per node

In this section, we describe the fourth possible way in which column generation could be applied, namely per node. First, a general explanation of this fourth application of column generation to the original MILP formulation is given in Section 5.5.1. This application of column generation requires the variable B_{dut} to be defined in a different way. There are three options to do so, which are explained in Section 5.5.2. Then, the formulations of the master problem and the pricing subproblems are derived in Sections 5.5.3 and 5.5.4, respectively.

5.5.1. General explanation

The fourth possible way in which column generation can be applied to the original MILP formulation is by noting that the problem can be decomposed into subproblems for each node $v \in V$. Therefore, column generation can be applied to this formulation, where columns are generated for each node $v \in V$. Such a column then describes the opening and operation of all assets $a \in A$ for all energy carriers $e \in E$, in all time periods $t \in T$, at a certain node $v \in V$ or at edges $\{v, v'\} \in U$ connected to that node. Any feasible solution then consists of a combination of columns, with exactly one column for each node.

The joint constraint is the constraint in the original MILP formulation in Appendix B.5.1 that contains a marked part, namely the energy balance constraint. This is the joined constraint, since this constraint sums over edges, that go between different nodes. Thus, this constraint is included in the master problem. To select columns in the column generation algorithm, a pricing subproblem for each node is solved. Such a pricing subproblem contains the original objective function and all constraints from the original MILP formulation, except the energy balance constraint. After introducing the notation for this column generation formulation, the master problem and pricing subproblems are derived.

But first, some notation is introduced. Let K_v be the set of columns that represent a feasible solution for node $v \in V$, for all time periods $t \in T$, all energy carriers $e \in E$, and for all assets $a \in A_e$. Let variable λ_v^k be the binary decision variable associated with column $k \in K_v$ for node $v \in V$. This decision variable is defined to be 1 if the column is used in the solution to the master problem, and 0 if it is not used. Then, the following parameters are used in these columns for supply, conversion and storage. Parameter B_{avt}^k denotes the number of investments at node v in column $k \in K_v$, in time period $t \in T$, in asset type $a \in A_e \setminus DA_e$ handling energy carrier $e \in E$. Furthermore, parameters S_{esvt}^k, M_{emvt}^k, $W_{ewvt}^{k,+}$ and $W_{ewvt}^{k,-}$ indicate the amount of energy carrier $e \in E$ that is supplied, converted, stored and withdrawn from storage, respectively, by asset type $s, m, w \in A_e$ at node $v \in V$ in column $k \in K_v$. Lastly, parameter $W_{ewvt}^{k,tot}$ indicates the total amount of energy carrier $e \in E$ that is stored at node $v \in V$ in asset $w \in WA_e$ at the start of time period $t \in T$ in column $k \in K_v$.

The parameters regarding the distribution involve two nodes, namely the two nodes that energy is distributed between. However, a parameter can only belong to a column for one of the nodes. Therefore, it is decided that each column for a certain node $v \in V$ only contains information on the amount of energy that is distributed out of that node, on all directed edges $(v, v') \in DU$. Thus, parameter F_{edut}^k indicates the amount of energy carrier $e \in E$ that is distributed on directed edge $u = (v, v') \in DU$ by asset type $d \in DA_e$ in column $k \in K_v$. However, the same separation per node cannot be made for the parameter regarding investments in distribution assets, since these assets are invested in on an edge, rather than a directed edge. There are three ways in which the column generation per node can still be implemented, which are explained in the next section.

5.5.2. Three options for implementation

The difficulty of implementing column generation per node mainly lies in the variable B_{dut}, which represents the investments in distribution assets $d \in DA$ at edge $u \in U$ in time period $t \in T$. In this section, it is explained why this creates a difficulty. Then, three ways in which column generation per node can still be implemented are explained and discussed, as they all have different drawbacks.

The variable B_{dut} is defined for each undirected edge, meaning that for two nodes $v, v' \in V$, $B_{d\{v,v'\}t}$ is the same as $B_{d\{v',v\}t}$. The problem then arises that this one variable is related to two nodes, namely v and v'. When column generation is implemented per node, there is one pricing subproblem for each node, which contains all variables concerning that node. However, variable $B_{d\{v,v'\}t}$ is related to both node v and v', but it cannot be in both pricing subproblems.

Therefore, B_{dut} has to be defined for directed edges $u \in DU$, instead of for undirected edges. Then, similarly to variable F_{edut}, all variables regarding directed edges going out of node v are in the pricing subproblem of v, but all variables regarding direct edges going into node v are not. Thus, parameter B_{dut}^k indicates the number of investments in distribution assets $d \in DA$ that are done on directed edge $u = (v, v') \in DU$ in time period $t \in T$ in column $k \in K_v$.

Therefore, the pricing subproblem of node v contains the following constraints:

$$B_{d(v,v')t} \leq N, \quad \forall d \in DA, v' \in V, t \in T$$
$$B_{d(v,v')t} \in \mathbb{Z}^+, \quad \forall d \in DA, v' \in V, t \in T$$
$$F_{ed(v,v')t} \in \mathbb{R}^+, \quad \forall d \in DA, v' \in V, t \in T$$

However, implementing the distribution capacity constraints is not straight-forward, since these originally contain constraints that concern the undirected edges, rather than the directed edges. The distribution capacity constraints in the original model for distribution between nodes v and v' in V are the following:

$$F_{ed(v,v')t} \leq \gamma_d \sum_{i=t_0}^{t} B_{d\{v,v'\}i}, \quad \forall e \in E, d \in DA_e, t \in T$$
$$F_{ed(v',v)t} \leq \gamma_d \sum_{i=t_0}^{t} B_{d\{v',v\}i}, \quad \forall e \in E, d \in DA_e, t \in T$$

Here, $B_{d\{v,v'\}t}$ and $B_{d\{v',v\}t}$ are the same variable. The constraints ensure that the amount $F_{ed(v,v')t}$ of energy carrier $e \in E$ that is distributed on a directed edge $(v, v') \in DU$ in time period $t \in T$, is constrained by the number of investments $B_{d\{v,v'\}i}$ on edge $\{v, v'\}$ in previous time periods. However, for column generation per node, the investments are done on directed edges, rather than undirected edges, thus these constraints need to be changed. There are three different options for implementing these constraints, which are explained now, as well as the disadvantage of each of them.

The first way in which the distribution capacity constraint can be implemented in column generation per node, is by including the following constraint in the pricing subproblem of node $v \in V$:

$$F_{ed(v,v')t} \leq \gamma_d \sum_{i=t_0}^{t} B_{d(v,v')i}, \quad \forall e \in E, d \in DA_e, v' \in V, t \in T.$$

The flow on every directed edge is then constrained by the investments in distribution assets on that directed edge. However, a negative consequence is that the investments on a directed edge from v to v' cannot be used to distribute energy from node v' to v, and the other way around. Thus, energy is to be distributed from node v to v' in one time period, but from node v' to v in a later time period, then new investments need to be done in distribution assets from node v' to v. This leads to a higher number of total investments, which leads to higher costs. Moreover, the constraints are different. Therefore, the problem and the solution space are different. This may result in an entirely different optimal solution.

For the second option for implementation, the following constraint is added to the master problem, to overcome this problem:

$$\sum_{k \in K_v} \lambda_v^k B_{d(v,v')t} = \sum_{k \in K_{v'}} \lambda_{v'}^k B_{d(v',v)t}, \quad \forall d \in DA, v, v' \in V, t \in T.$$

This constraint ensures that the investments done on the both of the directed edges between two nodes are the same. Note that the total costs of investments in distribution assets is then the following:

$$\sum_{d \in DA} \sum_{v \in V} \sum_{k \in K_v} \sum_{v' \in V} \sum_{t \in T} \frac{1}{2} \lambda_v^k B_{d(v,v')t}^k (c_{dut} + o_{dut}^{fix}).$$

The negative consequence of this option for implementation is that extra constraints are added to the master problem, which is likely to have a negative effect on the performance of column generation. Moreover, they are equality constraints, which are more difficult to satisfy than inequality constraints, which is also likely to have a negative effect on the performance.

A third option for implementing column generation per node is different from the previous two, since no constraints regarding the distribution capacity are included in the pricing subproblem. For this third option, the following constraints are added to the master problem, regarding the distribution of energy between nodes $v, v' \in V$:

$$\sum_{k\in K_v} \lambda_v^k F^k_{ed(v,v')t} \leq \gamma_d \sum_{i=t_0}^{t} \Bigg(\sum_{k\in K_v} \lambda_v^k B^k_{d(v,v')i} + \sum_{k\in K_{v'}} \lambda_{v'}^k B^k_{d(v',v)i} \Bigg), \quad \forall e \in E, d \in DA_e, t \in T$$

$$\sum_{k\in K_{v'}} \lambda_{v'}^k F^k_{ed(v',v)t} \leq \gamma_d \sum_{i=t_0}^{t} \Bigg(\sum_{k\in K_v} \lambda_v^k B^k_{d(v,v')i} + \sum_{k\in K_{v'}} \lambda_{v'}^k B^k_{d(v',v)i} \Bigg), \quad \forall e \in E, d \in DA_e, t \in T$$

$$\sum_{k\in K_v} \lambda_v^k B^k_{d(v,v')t} + \sum_{k\in K_{v'}} \lambda_{v'}^k B^k_{d(v',v)t} \leq N, \quad \forall d \in DA, t \in T$$

The first two constraints ensure that the flow on a directed edge (v, v') is constrained by the total number of investments done between the nodes v and v', namely by adding the investments done in the undirected edges in both ways together. Then, an extra constraint has to be added, to ensure that the total number of distribution asset investments between these two nodes does not exceed the limit N. The drawback to this option for implementation is also that many constraints are added to the master problem, which is likely to have a negative effect on the performance of column generation.

In the next sections, all three of these implementation options are taken into account. In the results section, the difference between the performances of all three of these implementation options is evaluated.

5.5.3. Master problem

The objective function of the master problem for column generation per node is similar to the original objective function, but including the new binary decision variable λ_v^k with $v \in V$, $k \in K_v$. For example, for node $v \in V$, variables $B_{d(v,v')t}$ are replaced by $\sum_{k\in K_v} \lambda_v^k B^k_{d(v,v')t}$, and variables $F_{ed(v,v')t}$ are replaced by $\sum_{k\in K_v} \lambda_v^k F^k_{ed(v,v')t}$. The objective function of the master problem is then as follows:

$$\begin{aligned}
\min \quad & \sum_{t\in T}\sum_{e\in E}\Bigg\{\sum_{v\in V}\sum_{k\in K_v}\Bigg[\sum_{v'\in V}\sum_{d\in DA_e} (\lambda_v^k B^k_{d(v,v')t}(c_{d\{v,v'\}t} + o^{fix}_{d\{v,v'\}t}) + o^{var}_{d(v,v')t} \cdot \lambda_v^k F^k_{ed(v,v')t}) \\
& + \sum_{s\in SA_e} (\lambda_v^k B^k_{svt}(c_{svt} + o^{fix}_{svt}) + o^{var}_{svt} \cdot \lambda_v^k S^k_{esvt}) \\
& + \sum_{m\in MA_e} (\lambda_v^k B^k_{mvt}(c_{mvt} + o^{fix}_{mvt}) + o^{var}_{mvt} \cdot \lambda_v^k M^k_{emvt}) \\
& + \sum_{w\in WA_e} (\lambda_v^k B^k_{wvt}(c_{wvt} + o^{fix}_{wvt}) + o^{+,var}_{wvt} \cdot \lambda_v^k W^{k,+}_{ewvt} + o^{-,var}_{wvt} \cdot \lambda_v^k W^{k,-}_{ewvt})\Bigg]\Bigg\}
\end{aligned}$$

Note that for the second implementation option, the costs regarding the investments in distribution assets are divided in half, thus:

$$\sum_{t\in T}\sum_{e\in E}\sum_{v\in V}\sum_{k\in K_v}\sum_{v'\in V}\sum_{d\in DA_e} \frac{1}{2}\lambda_v^k B^k_{d(v,v')t}(c_{d\{v,v'\}t} + o^{fix}_{d\{v,v'\}t}).$$

As explained in Section 5.5.1, the joint constraint in the original MILP formulation is the energy balance constraint. This constraint is rewritten for the master problem in the following way, again including the new binary decision variable λ_v^k:

$$\begin{aligned}
&\sum_{d\in DA_e}\sum_{v'\in V}\sum_{k\in K_{v'}} \lambda_{v'}^k F_{ed(v',v)t}^k \cdot \eta_{d\{v,v'\}} + \sum_{s\in SA_e}\sum_{k\in K_v} \lambda_v^k S_{esvt}^k \cdot \sigma_{st} \\
&\quad + \sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_{e'}}\sum_{k\in K_v} \lambda_v^k M_{e'mvt}^k \cdot \eta_{me'e} + \sum_{w\in WA_e}\sum_{k\in K_v} \lambda_v^k W_{ewvt}^{k,-} \cdot \eta_w^- \qquad\qquad (\pi_{evt}) \\
&\quad \geq d_{evt} + \sum_{d\in DA_e}\sum_{v'\in V}\sum_{k\in K_v} \lambda_v^k F_{ed(v,v')t}^k + \sum_{m\in MA_e}\sum_{k\in K_v} \lambda_v^k M_{emvt}^k + \sum_{w\in WA_e}\sum_{k\in K_v} \lambda_v^k W_{ewvt}^{k,+}, \\
&\qquad\qquad\qquad\qquad\qquad\qquad \forall e \in E, v \in V, t \in T
\end{aligned}$$

The dual variable corresponding to this constraint is denoted by π_{evt}. Furthermore, the following constraint is included in the master problem, which ensures that exactly one column is chosen for each node:

$$\sum_{k\in K_v} \lambda_v^k = 1, \quad \forall v \in V.$$

The dual variable corresponding to this constraint is denoted by μ_v. Lastly, a constraint is added to ensure that λ_v^k is indeed a binary variable:

$$\lambda_v^k \in \{0,1\}, \quad \forall v \in V, k \in K_v.$$

For the first implementation option, as described in Section 5.5.2, these are all the constraints in the master problem. For the second implementation option, the following constraints are added:

$$\sum_{k\in K_v} \lambda_v^k B_{d(v,v')t} = \sum_{k\in K_{v'}} \lambda_{v'}^k B_{d(v',v)t}, \quad \forall d \in DA, (v,v') \in DU, t \in T.$$

The dual variable corresponding to this constraint is denoted by ρ_{dut}. And for the third implementation option, the following constraints are added:

$$\begin{aligned}
&\sum_{k\in K_v} \lambda_v^k F_{ed(v,v')t}^k \leq \gamma_d \sum_{i=t_0}^{t} \left(\sum_{k\in K_v} \lambda_v^k B_{d(v,v')i}^k + \sum_{k\in K_{v'}} \lambda_{v'}^k B_{d(v',v)i}^k \right), \quad \forall e \in E, d \in DA_e, (v,v') \in DU, t \in T \\
&\sum_{k\in K_v} \lambda_v^k B_{d(v,v')t}^k + \sum_{k\in K_{v'}} \lambda_{v'}^k B_{d(v',v)t}^k \leq N, \quad \forall d \in DA, \{v,v'\} \in U, t \in T
\end{aligned}$$

The dual variable corresponding to the first constraint is denoted by $\tau_{ed(v,v')t}$, and the dual variable corresponding to the second constraint is denoted by $\xi_{d\{v,v'\}t}$. The complete formulations of the master problem can be found in Appendix B.5.2.

5.5.4. Pricing subproblems

As explained in Section 5.5.1, for this fourth application of column generation to the MILP, columns are generated per node. By solving a pricing subproblem for each of these nodes, it is decided which columns are added in each iteration of the algorithm. In this section, the formulation for this pricing subproblem is derived. Since this formulation is different for each of the implementation options, as described in Section 5.5.2, the notation of a column $k \in K_v$ for a node $v \in V$ is skipped, and the formulation of the different pricing subproblems is immediately derived.

The objective function of the pricing subproblem for $V \in V$ is the reduced cost of a column $k \in K_v$. The cost of column $k \in K_v$ is equal to the part of the objective function of the original formulation that contains variables regarding $v \in V$. Thus, the cost of column $k \in K_v$, for $v \in V$ for the first and third implementation option is:

$$\sum_{t\in T}\sum_{e\in E}\Bigg\{\sum_{d\in DA_e}\sum_{v'\in V}\left(B_{d(v,v')t}(c_{d\{v,v'\}t}+o^{fix}_{d\{v,v'\}t})+o^{var}_{d(v,v')t}\cdot F_{ed(v,v')t}\right)$$
$$+\sum_{s\in SA_e}\left(B_{svt}(c_{svt}+o^{fix}_{svt})+o^{var}_{svt}\cdot S_{esvt}\right)$$
$$+\sum_{m\in MA_e}\left(B_{mvt}(c_{mvt}+o^{fix}_{mvt})+o^{var}_{mvt}\cdot M_{emvt}\right)$$
$$+\sum_{w\in WA_e}\left(B_{wvt}(c_{wvt}+o^{fix}_{wvt})+o^{+,var}_{wvt}\cdot W^{+}_{ewvt}+o^{-,var}_{wvt}\cdot W^{-}_{ewvt}\right)\Bigg\}$$

For the second implementation option, the factor $\frac{1}{2}$ is included for the distribution asset investment costs. In the reduced cost of a column, the costs are subtracted by the terms in constraints in the master problem that contain a variable related to node $v \in V$, multiplied with their corresponding dual variable. Therefore, the terms in the constraints in the master problem that contain a variable related to node $v \in V$ are first identified.

The energy balance constraint in the master problem, corresponding to dual variable π_{evt}, contains variables regarding node $v \in V$. More specifically, for node $v \in V$, the left hand side of the energy balance constraint in the master problem contains the following term, containing variables regarding $v \in V$, for every $e \in E$ and $t \in T$:

$$\sum_{s\in SA_e} S_{esvt}\cdot\sigma_{st}+\sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_e} M_{emvt}\cdot\eta_{mee'}+\sum_{w\in WA_e} W^{-}_{ewvt}\cdot\eta^{-}_{w}.$$

And the right hand side contains the following term containing variables regarding $v \in V$:

$$\sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v,v')t}+\sum_{m\in MA_e} M_{emvt}+\sum_{w\in WA_e} W^{+}_{ewvt}.$$

The corresponding dual variable is π_{evt}. Furthermore, the energy balance constraint for a certain node $v' \neq v,\ v' \in V$, also contains variables regarding node $v \in V$. For all $v' \in V, v' \neq v$, $e \in E$ and $t \in T$, the energy balance constraint in the master problem is the following:

$$\begin{aligned}&\sum_{d\in DA_e}\sum_{v''\in V}\sum_{k\in K_{v''}}\lambda^{k}_{v''}F^{k}_{ed(v'',v')t}\cdot\eta_{d\{v',v''\}}+\sum_{s\in SA_e}\sum_{k\in K_{v'}}\lambda^{k}_{v'}S^{k}_{esv't}\cdot\sigma_{st}\\ &+\sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_{e'}}\sum_{k\in K_{v'}}\lambda^{k}_{v'}M^{k}_{e'mv't}\cdot\eta_{me'e}+\sum_{w\in WA_e}\sum_{k\in K_{v'}}\lambda^{k}_{v'}W^{k,-}_{ewv't}\cdot\eta^{-}_{w} &&(\pi_{ev't})\\ &\geq d_{ev't}+\sum_{d\in DA_e}\sum_{v''\in V}\sum_{k\in K_{v'}}\lambda^{k}_{v'}F^{k}_{ed(v',v'')t}+\sum_{m\in MA_e}\sum_{k\in K_{v'}}\lambda^{k}_{v'}M^{k}_{emv't}+\sum_{w\in WA_e}\sum_{k\in K_{v'}}\lambda^{k}_{v'}W^{k,+}_{ewv't},\\ &\qquad\qquad\forall e\in E, v'\in V, t\in T\end{aligned}$$

Then the left hand side of this constraint contains the following term, concerning the directed edge from node v to v':

$$\begin{aligned}
&\sum_{d\in DA_e}\sum_{v''\in V}\sum_{k\in K_{v''}} \lambda^k_{v''}F^k_{ed(v'',v')t}\cdot\eta_{d\{v',v''\}}\\
&=\sum_{d\in DA_e}\sum_{\substack{v''\in V:\\ v''=v}}\sum_{k\in K_{v''}} \lambda^k_{v''}F^k_{ed(v'',v')t}\cdot\eta_{d\{v',v''\}}+\sum_{d\in DA_e}\sum_{\substack{v''\in V:\\ v''\neq v}}\sum_{k\in K_{v''}} \lambda^k_{v''}F^k_{ed(v'',v')t}\cdot\eta_{d\{v',v''\}}\\
&=\sum_{d\in DA_e}\sum_{k\in K_v}\lambda^k_v F^k_{ed(v,v')t}\cdot\eta_{d\{v,v'\}}+\sum_{d\in DA_e}\sum_{\substack{v''\in V:\\ v''\neq v}}\sum_{k\in K_{v''}} \lambda^k_{v''}F^k_{ed(v'',v')t}\cdot\eta_{d\{v',v''\}}
\end{aligned}$$

Thus, for node $v' \in V$, the left hand side of the energy balance constraint contains the marked term concerning node $v \in V$. The corresponding dual variable is $\pi_{ev't}$. Thus, in the objective function of the pricing subproblem for $v \in V$, the following term is subtracted:

$$\begin{aligned}
&\sum_{t\in T}\sum_{e\in E}\Bigg\{\pi_{evt}\Bigg(\sum_{s\in SA_e} S_{esvt}\cdot\sigma_{st}+\sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_{e'}} M_{e'mvt}\cdot\eta_{me'e}+\sum_{w\in WA_e} W^-_{ewvt}\cdot\eta^-_w\\
&\quad-\sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v,v')t}-\sum_{m\in MA_e} M_{emvt}-\sum_{w\in WA_e} W^+_{ewvt}\Bigg)\\
&\quad+\sum_{d\in DA}\sum_{v'\in V}\pi_{ev't}\cdot F_{ed(v,v')t}\cdot\eta_{d\{v,v'\}}\Bigg\}
\end{aligned}$$

Lastly, for the constraint corresponding to dual variable μ_v in the master problem, μ_v is also subtracted in the reduced cost of column $k \in K_v$. In total, the objective function of the pricing subproblem for $v \in V$ for the first implementation option is:

$$\begin{aligned}
\min\quad &\sum_{t\in T}\sum_{e\in E}\Bigg\{\sum_{v'\in V}\sum_{d\in DA_e}\Big(B_{dut}(c_{dut}+o^{fix}_{dut})+o^{var}_{dut}\cdot F_{edut}\Big)\\
&\quad+\sum_{s\in SA_e}\Big(B_{svt}(c_{svt}+o^{fix}_{svt})+o^{var}_{svt}\cdot S_{esvt}\Big)\\
&\quad+\sum_{m\in MA_e}\Big(B_{mvt}(c_{mvt}+o^{fix}_{mvt})+o^{var}_{mvt}\cdot M_{emvt}\Big)\\
&\quad+\sum_{w\in WA_e}\Big(B_{wvt}(c_{wvt}+o^{fix}_{wvt})+o^{+,var}_{wvt}\cdot W^+_{ewvt}+o^{-,var}_{wvt}\cdot W^-_{ewvt}\Big)\\
&\quad-\pi_{evt}\Bigg(\sum_{s\in SA_e} S_{esvt}\cdot\sigma_{st}+\sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_{e'}} M_{e'mvt}\cdot\eta_{me'e}+\sum_{w\in WA_e} W^-_{ewvt}\cdot\eta^-_w\\
&\quad-\sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v,v')t}-\sum_{m\in MA_e} M_{emvt}-\sum_{w\in WA_e} W^+_{ewvt}\Bigg)\\
&\quad-\sum_{d\in DA}\sum_{v'\in V}\pi_{ev't}\cdot F_{ed(v,v')t}\cdot\eta_{d\{v,v'\}}\Bigg\}\\
&\quad-\mu_v
\end{aligned}$$

As discussed in Section 5.5.2, more constraints are included in the master problem for the second and third implementation option. These constraints also need to be incorporated in the objective of the pricing subproblem

For the second implementation option, the added constraint corresponding to dual variable ρ_{dut} contains the following terms regarding $v \in V$. For all directed edges $u = (v, v') \in U$ exiting node v, the left hand side of the constraint contains the term $B_{d(v,v')t}$, for every distribution asset $d \in DA$ and time period $t \in T$. And all directed edges $u = (v', v) \in DU$ entering node v, the right hand side of the constraint contains the term $B_{d(v,v')t}$, for every distribution asset $d \in DA$ and time period $t \in T$. Thus, the following term is added to the objective function of the pricing subproblem of node v:

$$\sum_{d \in DA} \sum_{v' \in V} \sum_{t \in T} \Big(- \rho_{d(v,v')t} \cdot B_{d(v,v')t} + \rho_{d(v',v)t} \cdot B_{d(v,v')t} \Big).$$

Note again that the costs of a column for the second implementation option is also slightly different, as a factor of $\frac{1}{2}$ in included in the distribution asset investment costs. For the third implementation option, it can be derived that the following term is added to the objective function of the pricing subproblem of node v:

$$\sum_{d \in DA_e} \sum_{v' \in V} \sum_{t \in T} \Bigg\{ \sum_{e \in E} \Bigg[\tau_{ed(v,v')t} \Bigg(- F_{ed(v,v')t} + \gamma_d \sum_{i=t_0}^{t} B_{d(v,v')i} \Bigg) + \tau_{ed(v',v)t} \cdot \gamma_d \sum_{i=t_0}^{t} B_{d(v,v')i} \Bigg]$$
$$- \xi_{d\{v,v'\}} \cdot B_{d(v,v')t} \Bigg\}$$

As explained earlier in Section 5.5.2, the following constraints are included in the pricing subproblem of node v:

$$B_{d(v,v')t} \leq N, \quad \forall d \in DA, v' \in V, t \in T$$
$$B_{d(v,v')t} \in \mathbb{Z}^+, \quad \forall d \in DA, v' \in V, t \in T$$
$$F_{ed(v,v')t} \in \mathbb{R}^+, \quad \forall d \in DA, v' \in V, t \in T$$

Furthermore, for the first and second implementation option, the following constraints are included in the pricing subproblem of node v:

$$F_{ed(v,v')t} \leq \gamma_d \sum_{i=t_0}^{t} B_{d(v,v')i}, \quad \forall e \in E, d \in DA_e, v' \in V, t \in T$$

There are no extra constraints regarding distribution included in the pricing subproblem for the third implementation option.

As explained in Section 5.5.1, the other constraints included in the pricing subproblem (for all implementation options) are all the constraints from the original MES model, except the energy balance constraint and constraints regarding distribution. The total formulations of the pricing subproblem can be found in Appendix B.5.3.

6 Data and implementation

In this section, a short description is given of the used data and implementation for the tests that are performed in the next chapter. First, the data set used for this research is described in Section 6.1. Then, some details on the implementation of the model are given in Section 6.2, as well as some details on the supercomputer on which the tests are run.

6.1. Case study Eindhoven

The data set used in this research is a case study of the city of Eindhoven in the Netherlands, as taken from Van Beuzekom et al. (2021). The energy carriers that are included in this multi-energy system case study are electricity, gas, and heat. A description of the node set is first given in Section 6.1.1. Thereafter, the included assets are given in Section 6.1.2. An explanation of the restrictions on investments in these assets is given in Section 6.1.3. Lastly, it is explained in Section 6.1.4 how the gas supply and demand data are obtained.

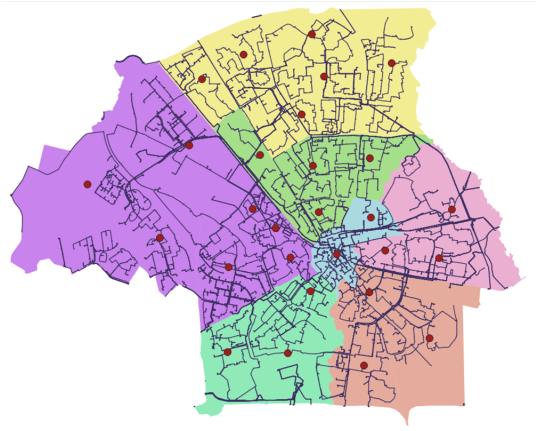


Figure 6.1: Current electricity network in the city of Eindhoven and the locations of the 28 nodes, spread across the seven city parts of Eindhoven (Bonenkamp, 2020).

6.1.1. Node locations

The data set is based on the city of Eindhoven, which consists of 7 city parts, and of 22 neighbourhoods. A map of Eindhoven is pictured in Figure 6.1, where the city parts are depicted by the different background colors. The network operator in Eindhoven, Enexis, has open data available on the placement of assets (“Enexis open data”, 2022). In the entire city, there are 110 locations where a medium pressure gas station is located, and over 400 points where a medium voltage electricity station is located. The set of nodes in the data set used in this research, is equal to the locations where a gas station is located, and the 400 electricity points are clustered to these same locations. The geographical coordinates of these locations are included in the data. With these coordinates, the distance between the nodes can be calculated, and thus the costs of a distribution asset between any two nodes can be calculated, since this cost depends on the length of the asset. However, since it takes very long for a solver such as Gurobi to find a relatively good feasible solution to the model for this large case, smaller cases were derived from this case by Van Beuzekom et al. (2021). The coordinates of the 110 nodes are clustered to 28 nodes, such that there is one node in each neighbourhood, and two in the neighbourhoods with the largest demand. The locations of these 28 nodes are also indicated in Figure 6.1. This data set is clustered even further to 7 nodes, such that there is one node in each city part. Thus, there are three different cases, consisting of 7, 28 and 110 nodes.

6.1.2. Assets

The assets that are included in the data set are the following. For distribution assets and storage assets, one type of asset for each energy carrier is included in the case study. Thus, one type of electricity distribution asset, one type of gas distribution asset and one type of heat distribution asset are included, and the same holds for storage assets. The costs c_{dut_0} or c_{avt_0} of these assets in the first time period are known. For later time periods, these costs change depending on the social discount rate and the technological development rate, as explained in Section 3. The social discount rate is assumed to be 4% in this research. The initial costs, technological development rates, capacities and loss factors of these distribution and storage assets are given in Table C.1 and C.4, respectively, in Appendix C. Other costs o_{dut}^{fix}, o_{dut}^{fix}, o_{avt}^{var} and o_{avt}^{var} are set to zero in this case study for all $d \in DA$, $u \in U$, $t \in T$, $a \in A \setminus DA$ and $v \in V$. All charging losses η_w^+ and discharging losses η_w^- are set to 1 in this research. Two types of electricity supply assets are included in the case study, namely a wind and a solar energy supply asset. The initial costs, technological development rates and capacities of these assets are given in Table C.2. The external factor for these supply assets (σ_t^s) is set to 1 for all time periods in this research. There are no supply assets for gas, since it is assumed that gas assets are already present in the city, in line with the current situation in Eindhoven. Therefore, the costs for investments in gas assets in the data are set to zero. Details on the available gas supply are discussed later in this section. No heat supply assets are included in the case study. Heat can only be generated by converting gas or electricity. The conversion assets included in the case study are a heat pump (HP), a combined heat and power asset (CHP) and a power-to-gas (P2G) asset. The initial costs, technological development rates, capacities and efficiencies of these assets are given in Table C.3. A summary of all the asset sets is the following, given that E = {electricity, gas, heat} = $\{e_1, e_2, e_3\}$, where the conversion assets belong to the subset of the incoming energy carrier:

$DA = DA_{e_1} \cup DA_{e_2} \cup DA_{e_3}$ = {electricity line} ∪ {gas pipeline} ∪ {heat pipline}
$SA = SA_{e_1} \cup SA_{e_2} \cup SA_{e_3}$ = {wind, solar} ∪ {gas supply} ∪ ∅
$MA = MA_{e_1} \cup MA_{e_2} \cup MA_{e_3}$ = {HP, P2G} ∪ {CHP} ∪ ∅
$WA = WA_{e_1} \cup WA_{e_2} \cup WA_{e_3}$ = {electricity storage} ∪ {gas storage} ∪ {heat storage}

6.1.3. Investment restrictions

The described data set is mostly greenfield data, rather than brownfield data. With brownfield data, current placement of energy assets, such as electricity lines and supply assets, would have been included in the data set. Greenfield means that no starting data of the infrastructure is included. In the data set, it is as though Eindhoven is a new city, where no energy assets are placed yet, thus it is Greenfield data. However, information on the current locations of gas stations is used to determine the coordinates of the nodes in the data set. Moreover, practical investment restrictions are included, which are explained in this section.

In order to obtain a solution from the model that can actually be implemented, certain constraints were added to limit the number of investments in assets that can be done in each node, namely Constraints (3.11) and (3.12) in the original model. In this case study, this maximum value of N investments is different for different asset types. Moreover, some extra constraints regarding investment limitations are added, to increase feasibility of the solution in Eindhoven in practice. The most important constraints are those regarding investments in wind supply assets. Wind supply assets are much less efficient when they are placed inside a city, due to turbulence, and they may cause nuisance. Therefore, wind supply assets are always built outside of cities. Electricity that is produced by wind supply assets outside the city can be imported in 7 locations in Eindhoven, according to data from “Enexis open data” (2022). Therefore, in the 110 nodes case, constraints are added such that wind supply assets can only be invested in at these 7 nodes. For the 7 and 28 nodes cases, these points are again clustered, and constraints are added such that wind supply assets can only be invested in at 4 and 5 nodes, respectively. Note that this may lead to significantly different results, as 7 out of 110 nodes is much less than 5 out of 28 nodes or 4 out of 7 nodes. The 110 nodes case is closest to reality.

Furthermore, for all asset types, constraints are added for the maximum number of investments in assets that can be done in each time period. For wind supply assets, this maximum number is 15, for solar supply assets, it is 30, and for all other asset types, it is 5. However, for the constraints on investments in wind and solar supply assets in the first time period, this maximum number is smaller. This prevents that the solution outcome of the model is to build many supply assets in the first time period. Because, if this number is very high, it is not a feasible solution in practice. The maximum number of investments in wind supply assets in the first time period is 10 in all nodes where wind assets can be invested in, and for solar supply assets this number is 20 in the first time period, in all nodes.

In summary, the maximum number of investments that can be made in each asset type, is different for each asset type and some time periods and locations. Therefore, Constraints (3.11) and (3.12) in the original model are replaced by the following constraints:

$$B_{dut} \leq 5, \quad \forall d \in DA, u \in U, t \in T,$$
$$B_{svt} \leq N_{svt}, \quad \forall a \in SA, v \in V, t \in T,$$
$$B_{avt} \leq 5, \quad \forall a \in MA \cup WA, v \in V, t \in T.$$

Here, N_{svt} denotes the maximum number of investments in supply asset $s \in SA$, at node $v \in V$, during time period $t \in T$. For solar supply assets, this value is 20 in the first time period, and 30 for the other time periods, for all nodes. For wind supply assets, this value value is 10 in the first time period at selected nodes, 15 in all other time periods at selected nodes, and 0 for the other nodes.

6.1.4. Time periods, gas supply and demand

The set of time periods in the data set consists of the even years from 2018 up to 2050, thus 2018, 2020, 2022, etc. The demand data of 2018 in the data set represents the total demand of 2018, for each energy carrier. The same holds for the gas supply data. The data on the odd years was left out, to constrain the size of the data set. For this same reason, only the data of the total demand and gas supply in the even years are included, rather than per smaller time periods.

The data on the gas supply is constructed in the following way. The total amount of available gas for Eindhoven in 2018 is known. It is also known, by the open data from “Enexis open data” (2022), in which locations in the city this gas can be imported, namely in 7 locations. These 7 locations correspond to 7 nodes in the 110 nodes case, and are clustered to 6 nodes for the 28 nodes case, and to 5 nodes for the 7 nodes case. For the first time period, the available gas is evenly distributed over these nodes. Eventually, the goal of the case is to enforce a 100% reduction of CO2-emissions in 2050, which translates to using no gas in 2050. Therefore, for the time periods after 2018, the total gas supply is linearly reduced to zero in 2050. The resulting gas supply in each time period is plotted in Figure 6.2. The supply capacity constraint, which is Constraint (3.5) in the original model, is therefore altered for gas supply in this case study. Instead of being limited by the number of investments in assets, the maximum gas supply is constrained by the amount of available gas in that node in that year, represented by parameter γ_{vt}^{gas}. This parameter has a value of zero in nodes where there is no gas supply, and it has a value of zero for time period 2050. Thus, the constraint is the following, for $e =$ gas and $s =$ gas supply asset:

$$S_{esvt} \leq \gamma_{vt}^{gas}, \quad \forall v \in V, t \in T.$$

Note that this constraint only contains variables concerning one time period $t \in T$. Therefore, it is not a joint constraint for column generation per time period. Thus it is not included in the master problem, but in the pricing subproblems instead.

The construction of the demand data required more data processing. Two reports regarding the climate plans for Eindhoven, namely Eindhoven Klimaatplan 2016-2020 (Municipality of Eindhoven, 2016) and Klimaatplan 2021-2050 (Municipality of Eindhoven, 2020), were used by Van Beuzekom to construct this data, as well as national statistics. Further details on the construction of this demand data can be found in the PhD thesis by Van Beuzekom (2022). The resulting demand of each carrier in each time period is plotted in Figure 6.3. Note that although electricity and heat demand do not appear to change much, in reality many energy services that once required gas, are shifting to using heat or electricity (so-called 'electrification'). The reason that the demand does not increase because of that, is due to the expected increase in energy efficiency in these services.

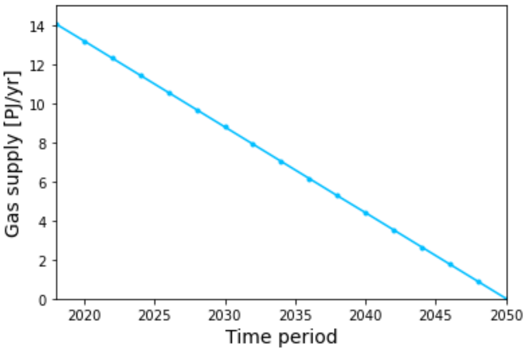


Figure 6.2: Gas supply in all time periods, linearly reduced to zero in 2050.

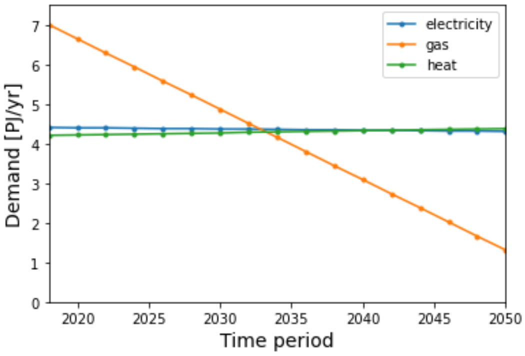


Figure 6.3: Electricity, gas and heat demand in all time periods.

6.2. Implementation and test details

The model as taken from Van Beuzekom et al. (2021) is programmed in Python, using the package Pyomo. Pyomo is a Python-based, open-source optimization modeling language with a diverse set of optimization capabilities (Pyomo, 2022). It provides the component **Block**, which is a generalized container for defining hierarchical models. These blocks are perfectly suitable for implementing the master problem of column generation, in the following way. If there is a pricing subproblem for each $t \in T$, then columns are produced for each $t \in T$, containing a parameter value for each variable in that subproblem. It can be rather involved to implement all these different parameter values, for different columns $k \in K_t$, since (the size of) the set of columns K_t is different for every $t \in T$. An elegant solution is to create a **Block** for each $t \in T$, containing the information of all generated columns for that $t \in T$, as well as the associated variables λ_t^k for all $k \in K_t$, and the convexity constraint $\sum_{k \in K_t} \lambda_k^t = 1$.

Some details on how the tests in the next chapter are run are now explained. The MES model in this research is complex, since it is based on two NP-hard optimization problems, and the data sets used are large, especially the 110 nodes case. To be able to run the model properly and get a consistent outcome for all the tests, a supercomputer was used in this research, namely DelftBlue ((DHPC), 2022). All tests in this research are performed on this supercomputer.

The DelftBlue supercomputer is operated by the Delft High Performance Computing Centre (DHPC), which is the central computing center of the Delft University of Technology. It is currently in phase 1, and consists of 10 high memory nodes, among other node types. These high memory nodes have two Inel Xeon E5-6248R processors, with 24 cores and a processor base frequency of 3.0 GHz. Six of these nodes have a memory of 768 GB, and four a memory of 1536 GB. These high memory nodes are the nodes that the tests for this research are run on, since the data sets are rather large, and the computations are complex and require much memory. The maximum time limit that is allowed for a student for a test is 24 hours, and the maximum limit of available memory is 250GB. In practice, 100GB of memory was satisfactory for all tests in this research.

In order to evaluate the performance of the greedy algorithm and column generation, the performance of the current model first has to be established. Therefore, several tests are first run to check the performance of the current model, using the Delftblue supercomputer and the solver Gurobi ("Gurobi - The Fastest Solver", 2022) to solve the model for the 7, 28 and 110 node cases. After evaluating the performance of the original model, the greedy algorithm is applied to the model and tested using the supercomputer. Lastly, the performance of column generation is evaluated by running tests on the supercomputer.

7 Results

In this chapter, the results of this research are given. First, the results of the original model are given in Section 7.1, which are used as a benchmark to compare the results of the greedy algorithm and column generation. Then, the results of the greedy algorithm are given in Section 7.2, which can thereafter be used as a starting solution for the column generation algorithm. Lastly, the results of the four different applications of column generation to the MES model are given in Section 7.3. The performances of the greedy algorithm and column generation are evaluated and compared to the performance of the original MES model. Moreover, the number of investments that are made in the solutions in different assets and different time periods are further investigated, to explain the differences in performance.

7.1. Results original model

In this section, the results of the original MES model are discussed. The model is solved by the Gurobi solver on the supercomputer with a time limit of 24 hours. First, a short overview of the results of the MES model is given in Section 7.1.1. These results can be used to compare the results of column generation to. Thereafter, an in depth analysis of the solutions found by the solver for the 7 nodes case and the 28 and 110 nodes cases are given in Sections 7.1.2 and 7.1.3, respectively. Here, the solutions are further investigated in terms of the investments in each asset type in the different time periods. These results give a better insight into the solutions found by the solver, and they are used to compare it with the solution found by the greedy algorithm in the section thereafter.

7.1.1. Results Gurobi solver

The performance of the original model by Van Beuzekom et al. (2021) is evaluated in this section. The model is applied to the 7, 28 and 110 nodes cases, and the Gurobi solver is used to solve the model. To get an impression of the performance, the cases are run for 23 hours. A run time limit of 23 hours was chosen, despite the run time limit of the supercomputer being 24 hours, because Gurobi needs some time to stop after the given run time limit. Therefore, it cannot always show the best found solution if the run time limit is set to a value closer to 24 hours. In Table 7.1, the results of these runs can be found. The table shows the objective function value of the best found integer solution after the set run time, and the optimality gap of that solution.

Table 7.1: Objective function value and optimality gap of the best found integer solution to the 7, 28 and 110 nodes cases by the Gurobi solver after a set run time of 23 hours.

Number of nodes	Set run time [h]	Objective function value [M€]	Optimality gap [%]
7	23	699.66	0.14
28	23	631.47	1.78
110	23	922.35	30.83

It can be seen in Table 7.1 that it indeed is more difficult for the solver to decrease the optimality gap for the larger cases. Especially for the 110 nodes case, the optimality gap is very large. It was expected that the model would not perform well for the larger cases, since the optimization problem of the MES model is very complicated, as it is a combination of two NP-hard optimization problems.

Considering the objective of the model, namely to find a good solution that can be implemented as an investment plan in Eindhoven, a solution with an optimality gap of 1% would generally be satisfactory. For the 7 nodes case, the solver is quickly able to find a better solution than that. For the 28 nodes case, the 1% gap is almost achieved after 23 hours. Yet for the 110 nodes case, a gap of 30.8% is the best that can be found by the supercomputer within 24 hours. Note that the best integer solution that the solver has found might already be the optimal integer solution. However, since the optimality gap is not zero, it has not yet proven that this is the optimal solution. Therefore, a solution with such a large gap would be unacceptable to use for an urban decision maker. This underlines the reason that this thesis aims to lower the gap, in a shorter amount of time. To improve the optimality gap using column generation, either a better integer solution should be found should be found within the same amount of time, or a solution to the relaxed problem that is a better lower bound.

It is interesting to note that the objective function value of the solution for the 28 nodes case is much lower than for the 7 nodes case. A possible reason for this is investigated in the next section, where the solutions found are further analyzed. This analysis is also used for a comparison with the greedy algorithm.

7.1.2. Solution Gurobi 7 nodes case

The solutions found by the Gurobi solver are now further analyzed by looking into the investments that are made in these solutions. The solution to the 7 nodes case is further analyzed in this section, and the solutions to larger cases are further inspected in the next section.

In order to get a better insight into the solutions, besides just the total costs, bar plots of the investments in the solutions are analyzed. The total number of investments in different asset types in the solution for the 7 nodes case are shown in Figure 7.1. In this bar plot, the total number of investments in each asset type in a decade are added together, and represented by one bar(part). The bars representing investments in assets of the same category (distribution, conversion, and storage) are stacked on top of each other. The bars representing the investments in supply assets, namely wind and solar supply assets, are not stacked, because they are very large.

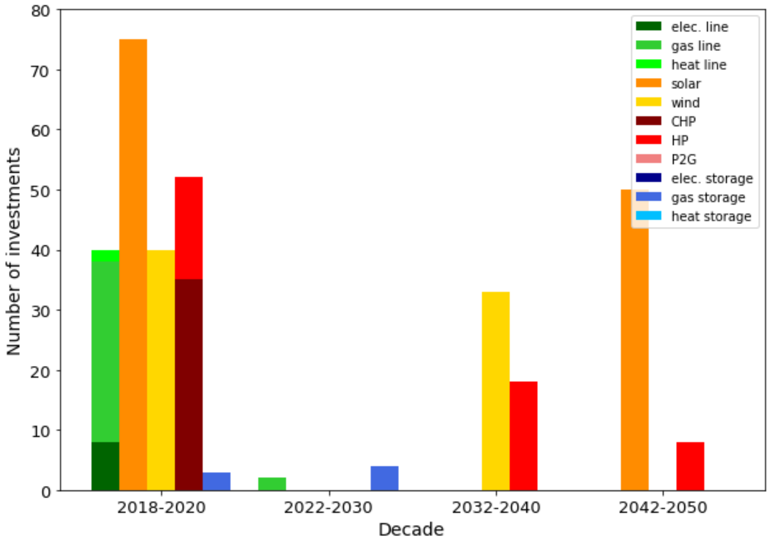


Figure 7.1: Number of investments made in the different asset types in the solution for the 7 nodes case, found by the Gurobi solver after 23 hours. The total number of investments per decade are shown. The bars representing investments in distribution assets (electricity lines, gas pipelines and heat pipelines) are green and stacked on top of each other. The orange and yellow bars represent investments in solar an wind assets, respectively. The bars representing investments in conversion assets are red and stacked, and those representing investments in storage assets are blue and stacked.

In Figure 7.1, it can be observed that the total number of investments is the largest in the first decade.

This can be explained by the fact that the case studies consist of greenfield data, meaning no data on prior investments is included. Therefore, many assets are built in the first time period. After the first time period, the investments are driven by the reduction of available gas. As the gas supply reduces, more electricity supply assets and heat pumps are needed to satisfy the heat demand. Therefore, in the third and fourth decade, an increasing number of electricity supply assets are invested in, as well as heat pumps. In addition, some gas storage is built in the first and second decade. These storage assets can store leftover gas from the first time periods, which can be used to satisfy the gas demand in later time periods. Presumably, the extra gas distribution assets that are invested in during the second decade are needed to distribute gas to these storage assets.

Another noticeable result in Figure 7.1 is that in the last decade, many solar supply assets are invested in, whereas in the third decade only wind assets are invested in. A possible reason for this is the following. In the first decade, many CHPs are invested in and used to convert gas to heat and electricity. This electricity is then distributed to other nodes via electricity lines. In the third decade, less gas is available. Since electricity from wind supply assets is cheaper per unit energy than from solar assets, wind assets are invested in, and this electricity is distributed via the lines that electricity from the CHPs was previously distributed on. This theory is supported by the fact that no distribution assets are invested in after the first decade. In the last decade, the amount of electricity that is needed for the HPs increases further. Solar supply assets have a much smaller capacity than wind supply assets, and they can be invested in at every node. Therefore, they are better suited for tailoring to the increasing electricity demand than large wind supply assets in the last decade. This makes solar supply assets a better investment in the last decade than wind supply assets.

It is also interesting to look into which assets are not invested in. No P2G assets are invested in, and the same holds for electricity and heat storage. Apparently, these technologies are not efficient enough yet or too expensive to invest in. And indeed, electricity storage has a huge loss factor of 96% and a very small capacity, which makes it extremely expensive. Heat storage is more than 13 times as expensive as gas storage in the first time period, and has a loss factor of 10% compared to 0.02%. P2G is also has an extremely high cost in the first time periods. Due to a high technological development rate, the cost is much lower in later time periods, but apparently it is not competitive with gas storage for obtaining gas. Furthermore, after the first decade, barely any more distribution assets are invested in, except for a few gas and electricity lines in the second decade and third decade. This can be explained by the fact that the total demand for heat and electricity does not change much from 2018 to 2050. Therefore, no more distribution assets are needed in later years. There are also no investments in CHPs after the first decade. This can be explained by the decrease in available gas over the time periods, which is what drives CHP conversion. Thus, less gas can be converted in CHPs, and more HPs are needed to satisfy the heat demand.

7.1.3. Solution Gurobi 28 and 110 nodes cases

In this section, the results of the solutions found by the Gurobi solver for the 28 and 110 nodes cases are investigated. To identify the main differences in the solutions compared to the results of the 7 nodes case, the total number of investments in each asset type in each solution are given in Table 7.2. The differences are further analyzed by using Figure 7.2 and 7.3, in which the solutions of the 28 and 110 nodes case are plotted, respectively.

The main differences between the solutions for the three cases that can be identified from Table 7.2 are the following. Firstly, the number of investments in electricity and gas lines in solutions of the larger cases are much higher than in the solution of the 7 nodes case. This is an obvious result, since there are simply many more edges in the larger cases, thus more lines are needed.

Furthermore, more investments in CHPs are made in the larger cases. The reason being that CHPs are relatively cheap and efficient, and therefore a good investment. Note that no CHPs are invested in during later time periods, thus they are apparently only a good investment in the first time period. In order to find a practically implementable solution, the maximum number of CHPs that can be invested in was set to 5 in each node, in each time period. In the solution for the 7 nodes case, the maximum number of investments ($7 \cdot 5 = 35$) are made in the first time period. For the larger two cases, the maximum total number of assets that can be invested in during the first time period is higher, since there are simply more nodes. Since CHPs are such a good investment in the first time period, more CHPs are invested in, and less investments in other assets are needed. This results in a cheaper solution overall, which is why the objective function value of the solution for the 28 nodes case is lower than that of the 7 nodes case, as can be observed in Table 7.1.

Table 7.2: Total number of investments made from 2018-2050 in the different assets types in the solutions to the 7, 28 and 110 nodes cases, as found by Gurobi after the maximum run time.

	Case		
Asset type	7 nodes	28 nodes	110 nodes
Electricity line	8	32	120
Gas pipeline	32	91	189
Heat pipeline	2	7	2
Solar supply	125	48	290
Wind supply	73	91	0
CHP	35	41	66
HP	43	48	113
P2G	0	0	7
Electricity storage	0	0	0
Gas storage	7	7	65
Heat storage	0	0	0

From Table 7.2, it can also be concluded that the solution for the 110 nodes case is very different from the smaller cases. There are large differences in the number of investments in solar and wind supply assets, HP and P2G conversion assets and gas storage assets. The reason for these large differences is investigated later in this section.

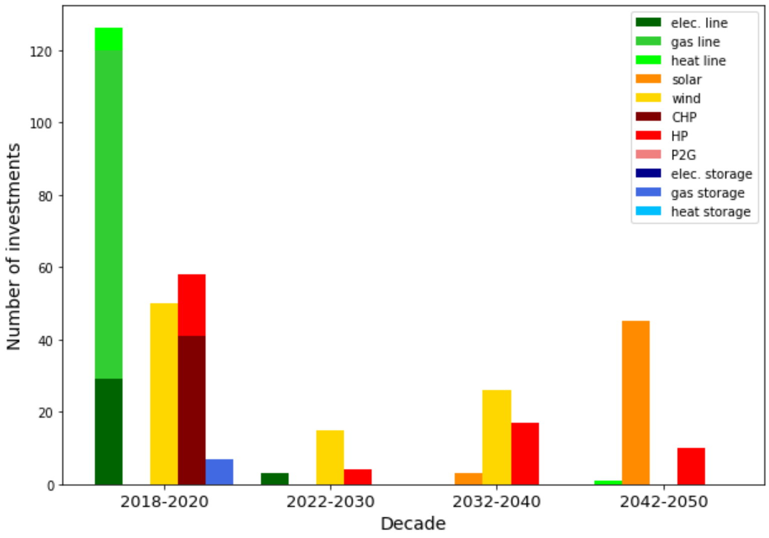


Figure 7.2: Number of investments made in the different asset types in the solution for the 28 nodes case, found by the Gurobi solver after 23 hours. The total number of investments per decade are shown, and the bars representing investments in assets of the same category (distribution, conversion, and storage) are stacked on top of each other.

More complicated differences are better observed by investigating the barplots of the solution. The solution found by the solver for the 28 nodes case is plotted in Figure 7.2. As mentioned earlier, many CHPs are invested in during the first time period. In almost every node, one or more CHPs are invested in. Therefore, electricity and heat are produced locally, and not many distribution assets are needed. However, many gas lines are invested in. These gas distribution assets are needed to distribute gas from the supply nodes to all other nodes, to convert it in the CHPs.

Furthermore, more investments in wind assets are made in the first time period in the 28 nodes case than in the 7 nodes case. On the other hand, no investments in solar assets are made, whereas 80 investments in solar assets are made in the solution for the 7 nodes case. This is probably due to the fact that, in the 28 nodes case, many more wind assets can be invested in during the first time period than in the 7 nodes case, since constraints restrict the number of investments per node. Using wind supply assets to obtain electricity is cheaper per unit energy than using solar supply assets, thus more wind supply assets are invested in during the first time period. Enough wind supply assets are invested in to fulfill the electricity demand, thus no additional investments in solar supply assets are needed in the first decade. Electricity distribution assets are invested in to distribute this electricity from the nodes where investments in wind supply assets are made to all the other nodes.

The reason that the total costs (equal to the objective function value) of the solution for the 28 nodes are lower than for the 7 nodes case, probably also comes from the fact that more CHPs and wind supply assets can be invested in during the first time period. CHPs are a very good investment in the first time period, since they produce electricity and heat efficiently by converting gas, which is available for free. Less HPs and electricity supply assets are then needed to satisfy the heat and electricity demand, resulting in lower total costs. Furthermore, wind assets are cheaper per produced unit of electricity than solar supply assets. Thus, being able to invest in more wind supply asset significantly decreases the costs.

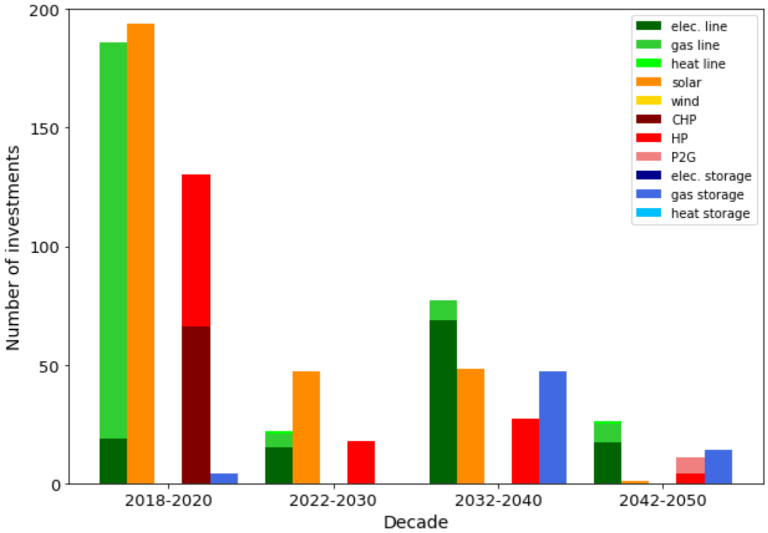


Figure 7.3: Number of investments made in the different asset types in the solution for the 110 nodes case, found by the Gurobi solver after 23 hours. The total number of investments per decade are shown, and the bars representing investments in assets of the same category (distribution, conversion, and storage) are stacked on top of each other.

The solution found by the solver for the 110 nodes case is plotted in Figure 7.3. This solution is very different from those for the other cases. It should be noted that this solution has a much higher optimality gap, and might therefore be a much worse solution. The main reason to still investigate this result, is to be able to compare it with the results of the greedy algorithm.

In the first decade, significantly more CHPs are invested in, and many gas lines are invested in, but no heat lines are invested in, as can also be seen in Table 7.2. Furthermore, as opposed to the 28-node case, no investments are made in wind supply assets. Instead, all renewable energy is generated using solar supply assets. The absence of investments in heat lines means that heat is produced locally. One or more CHPs are built in around half of the nodes. Gas distribution assets are needed to distribute gas from the nodes with available gas supply to all nodes with a CHP. In the other nodes, investments in HPs are made to fulfill the heat demand locally. Similarly, relatively little electricity distribution assets are needed, since electricity is produced locally by CHPs and solar supply assets. Though there are many more edges in the 110 nodes case than in the 28 nodes case, less electricity lines are invested in during the first decade.

In the smaller cases, wind supply assets seemed a better investment in the first decade than solar supply assets. The reason that it cheaper to produce electricity locally using solar supply assets in the 110 nodes case, rather than by using wind supply assets, is the following. Wind supply assets can only be invested in at very few nodes, namely 8 of the 110 nodes, and in even less nodes in the smaller cases. These wind supply assets have a very high capacity, thus a large amount of electricity can be produced. To distribute this electricity to all other nodes, many electricity distribution assets need to be invested in. A difference between the cases is that the demand in each node is smaller in the 110 nodes case, since the same amount of electricity demand as in the smaller cases is now divided over 110 nodes. Therefore, solar supply assets, which have a much smaller capacity, probably fit the smaller demand in each of the 110 nodes better than wind supply assets combined with many distribution assets. Thus, in the 110 nodes case, it is cheaper to produce energy locally using solar supply assets than it is to invest in wind supply assets and electricity distribution assets. And indeed, in the first time period, one or more solar supply assets are built in nearly every node.

It can also be observed that many more investments in HPs are made in the solution of the 110 nodes case, as can also be seen in Table 7.2. This is probably related to the fact that many solar supply assets are invested in locally (in almost every node). Thus, electricity is produced locally, which makes it cheaper to obtain heat by investing in HPs locally. Especially in the third decade, many more HPs are invested in, likely to be able to fulfill the heat demand, when less gas is available for the CHPs. Many more electricity distribution assets and solar supply are also invested in during the third decade. The solar supply assets are needed to produce more electricity, which is needed for the extra HPs. The distribution assets are probably needed to distribute electricity from the nodes where the extra solar supply assets are invested in, to the nodes where the extra HPs are invested in.

Furthermore, many investments in gas storage assets are made in later time periods, as well as gas distribution assets. At more than half of the nodes, one gas storage asset is invested in. Therefore, the gas lines are needed to distribute gas from the nodes where gas supply is available to the nodes with a gas storage asset.

A last observation that can be made is that investments in P2G assets are made in the solution of the 110 nodes case, unlike the solutions of the other cases. This is likely the result of more CHPs being invested in during the first time period, which are able to convert more gas, thus leaving less gas to put into storage in the first decades, to save gas for the last decade. Therefore, P2G assets are needed in the last decade to fulfill the gas demand.

7.2. Results greedy algorithm

In this section, the results of the greedy algorithm are evaluated. The maximum energy path length is an input parameter to the algorithm. A longer maximum length results in a longer needed run time, and results in a different solution. Therefore, the effect of different maximum lengths of energy paths is first evaluated in Section 7.2.1. The maximum path length that results in the best solution for each case is identified, and thereafter called the ‘best maximum energy path length’. The best results of the greedy algorithm are then further analyzed in Section 7.2.2 by comparing bar plots of the solutions to those of the Gurobi solver. This gives an insight into the way the greedy algorithm works and possible shortcomings. The comparison between the greedy algorithm and the solver is also needed to decide which of the two is better suited for producing an initial solution for column generation, both in terms of the quality of the solution, as well as the needed computation time. In this section, the question arises whether including the option of investing in storage has a positive effect on the solution. Therefore, in Section 7.2.3, it is investigated whether storage should be included or not to obtain a better solution.

7.2.1. Effect maximum energy path length

In this section, the effect of the maximum length of energy paths is investigated, such that the best maximum energy path length can be identified. The longest length that a path can have is $|V|+|MA|-1$, which represents the number of nodes plus the number of conversion assets minus 1, as explained in Chapter 4. For the 7 nodes case, this is equal to $7+3-1=9$. Therefore, the maximum path length for the 7 nodes case can be set to any value from 1 to 9. The objective function values that the greedy algorithm finds for each of these path lengths are given in Table 7.3, as well as the amount of time that is needed by the algorithm to find this solution.

As can be seen in Table 7.3, the maximum allowed path length needs to be four or higher in order to be able to obtain a feasible solution for this case. It can be concluded that the greedy algorithm finds the best solution when the maximum path length is set to 6, thus this is the best maximum length for the 7 nodes case. It can also be seen in this table that the needed run time increases as the maximum path length is increased. This can be explained by the fact that there are many more possible paths that need to be calculated, thus more time is needed.

Table 7.3: Objective function value of the solution found by the greedy algorithm and the needed run time, for different maximum path lengths.

Max. path length	Objective function value [M€]	Needed run time [s]
1	infeasible	-
2	infeasible	-
3	infeasible	-
4	863.47	14
5	859.94	62
6	859.35	210
7	860.40	617
8	859.67	1458
9	859.67	1968

It is interesting that the solution with a maximum path length of 6 is better than the solutions with 4 and 5 as the maximum, but then worsens as the maximum length is set to a higher value. The reason why the solution for a maximum path length of 6 is better than the solutions of 4 and 5, is probably just that the possibility of longer paths eventually results in a better overall solution. However, it might not be as obvious why the objective function value worsens as the maximum path length increases, since the number of possibly good paths is larger when the maximum path length is larger. It can likely be explained by the fact that the algorithm sets the costs to zero when an asset not used to its maximum capacity. For example, assume that the cheapest path at some point is to invest in solar supply assets in node v. Assume 10 solar assets can be invested in, and $4^1/_2$ assets are enough to satisfy the remaining electricity demand in node v. Then, 5 solar supply assets are invested in, but half of the capacity of one asset is not used. This results in an energy path for supplying electricity in node v with a capacity that is equal to half of the capacity of the solar supply asset, which has a cost of zero. Then, it is likely that in the next iteration, the cheapest energy path is to supply solar energy in node v, and distribute it to node v'. Again, a distribution asset that is then invested in may not be used to its full capacity. Therefore, adding another asset to this path again results in a cheap path. This continuing process results in very long paths that are relatively cheap, whilst that is likely not the overall optimal solution. However, whether the maximum path length is set to 8 or 9 does not affect the solution anymore, according to Table 7.3.

Next, the effect of the maximum path length on the 28 and 110 nodes cases is shortly analyzed. The objective function values of the solutions that the greedy algorithm finds for different maximum path lengths for the 28 and 110 nodes cases are given in Table 7.4 and Table 7.5, respectively. For the larger cases, the greedy algorithm requires significantly more time to find solutions for higher maximum path lengths, since the number of possible paths is much higher. Due to the 24-hour run time limit of tests on the supercomputer, the maximum path length for the 28 nodes case is limited to 4, and to 3 for the 110 nodes case.

Table 7.4: Objective function value of the solution to the 28 nodes case found by the greedy algorithm and the needed run time, for different maximum path lengths.

Max. path length	Objective function value [M€]	Needed run time [s]
1	infeasible	-
2	infeasible	-
3	769.38	344
4	781.04	6468
5	-	>24 hours

Table 7.5: Objective function value of the solution to the 110 nodes case found by the greedy algorithm and the needed run time, for different maximum path lengths.

Max. path length	Objective function value [M€]	Needed run time [s]
1	infeasible	-
2	1072.65	785
3	1109.74	68,653
4	-	>24 hours

From Table 7.4, it can be concluded that the maximum allowed path length needs to be at least 3 to obtain a feasible solution. Yet the best maximum path length equals 3. The computation time needed for this case is much higher than for the 7 nodes case, and the solutions with a maximum path length of 5 or higher cannot be calculated within 24 hours on the supercomputer.

For the 110 nodes case, it can be concluded from Table 7.5 that the maximum allowed path length in the greedy algorithm needs to be at least 2 to obtain a feasible solution. The greedy algorithm finds the best solution within 24 hours when the maximum energy path length is set to 2. Furthermore, it can be concluded that the objective function value, equal to the total costs of the solution, is much higher for the 110 nodes case than for the smaller cases, regardless of the maximum path length. This is likely due to the capacity of the assets included in the case study. The demand in each node is much smaller in the 110 nodes case than in the smaller cases, since the same amount of demand is divided over more nodes. Therefore, the capacity does not match the demand in the 110 nodes case in the same way as in the smaller cases. The assets are more likely to have an overcapacity. Thus, regarding assets with smaller capacities for the 110 nodes case is likely to result in less overcapacity, and lower total costs.

In the next section, these results are further investigated, and compared to the results of the Gurobi solver.

7.2.2. Greedy algorithm vs. Gurobi

In this section, the best solutions found by the greedy algorithm are further analyzed and compared to the solutions found by the Gurobi solver. The deeper analysis gives an insight into the way the greedy algorithm works, and what flaws it may contain. From the comparison with the solutions obtained by the Gurobi solver, it can be decided which of the two is better suited for producing an initial solution for column generation. To conclude the results so far, the best found solutions by the greedy algorithm for each case are given in Table 7.6. For a fair comparison of objective function values, the solutions found by the Gurobi solver for each case within the same amount of time (rounded up) are also given in Table 7.6, except for the 110 nodes case. The greedy algorithm finds the solution in around 800 seconds, whereas it takes the Gurobi solver around 2000 seconds to find one feasible solution to the original model. The solution of the Gurobi solver given in the table is this first found feasible solution.

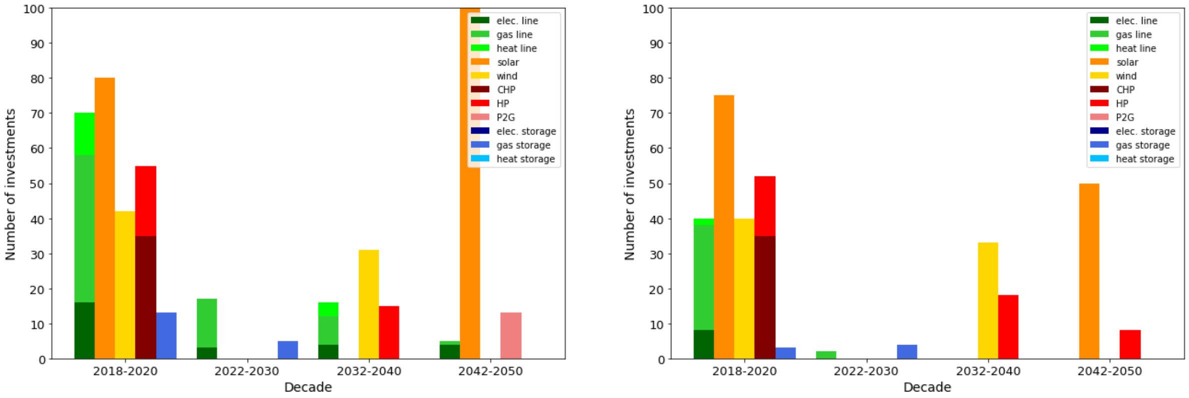
Table 7.6: Objective function values of the best found solutions found by the greedy algorithm for each case and the objective function values of the solutions found by the Gurobi solver in the approximately the same amount of time.

# nodes	Obj. greedy algorithm [M€]	Obj. Gurobi solver [M€]	Difference with greedy [%]
7	838.76	711.50	-15.2
28	769.38	658.82	-14.4
110	1107.02	922.35	-14.0

It can be concluded from Table 7.6 that the greedy algorithm performs worse than the Gurobi solver. For the 110 nodes case, it is able to find a solution faster than the Gurobi solver, but the Gurobi solver finds a much better solution after 1200 seconds more. It can also be observed that performance of the greedy algorithm is relatively worse for smaller cases than for larger cases. To be able to explain the higher objective function values of the solutions found by the greedy algorithm, bar plots are again used to gain a better insight into the solutions, starting with the 7 nodes case.

The 7 nodes case

The investments in the solution found by the greedy algorithm for the 7 nodes case are displayed in a bar plot in Figure 7.4a. To enable an easy comparison, Figure 7.4b (same as Figure 7.1) shows the solution found by the Gurobi solver.



(a) Solution obtained by the greedy algorithm with the maximum path length set to the best length. The total number of investments made in solar supply assets in the fourth decade is 460.

(b) Solution obtained by the Gurobi solver, same as Figure 7.1.

Figure 7.4: Number of investments made in the different asset types in the solution for the 7 nodes case, found by the greedy algorithm (left) and the Gurobi solver (right). The total number of investments per decade are shown, and the bars representing investments in assets of the same category (distribution, conversion, and storage) are stacked on top of each other.

The results obtained by the greedy algorithm and the Gurobi solver are similar in the first decades, as shown in Figure 7.4. However, significantly more investments in distribution assets are made in the solution found by the greedy algorithm. Furthermore, a major difference between the solutions occurs in the last decade. The greedy algorithm incurs far more investments in solar supply assets and P2G assets in the last decade. Both of these differences explain why the objective function value of the solution found by the greedy algorithm is so much higher. More distribution assets result in a higher total cost. But more importantly, P2G assets are relatively very expensive, thus these costs mainly contribute to a higher objective function value. The reason that these assets are invested in is related to the nature of the greedy algorithm. It is now explained why this is the case.

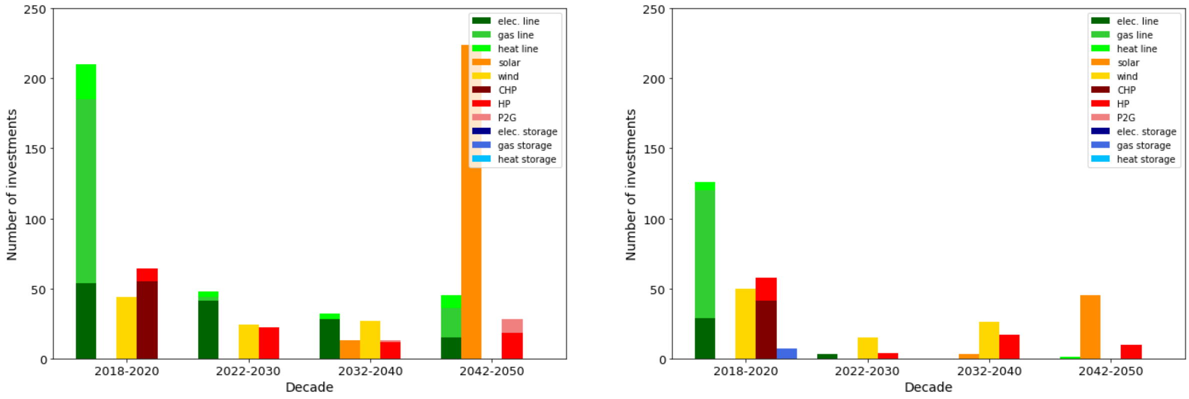
The reason that the number of investments in conversion and supply assets are similar in both solutions in the first decades, but many more distribution assets are needed in the greedy solution, is likely due to the fact that the greedy algorithm makes locally optimal choices, based on the lowest cost. The costs of investments in conversion and energy supply assets are relatively much higher than the costs of distribution assets. Therefore, the greedy algorithm mainly bases its choices for the solution on the investments of conversion and supply assets. Thus, the greedy algorithm is likely to find a similar investment plan for these assets as the Gurobi solver. However, due to the design of the greedy algorithm, it might not do the investments in the best places. Then many more distribution assets are required to distribute the energy to the locations where it is needed. The number of investments in distribution assets in the solution found by the greedy algorithm for the 7 nodes case is also much higher than in the solution found by the Gurobi solver, though the number of investments in CHPs is the same in both solutions, which supports this theory. This higher number of needed investments in distribution assets results in a higher total cost, thus this is part of the reason that the objective function value of the solution found by the greedy algorithm is higher.

Furthermore, P2G assets are needed to satisfy the gas demand in the last decade, and the solar supply assets are likely needed to supply electricity for these P2G assets, even though gas storage assets are invested in during the first time period. The reason that this is needed, also comes from the fact that the algorithm is a greedy algorithm. In the first three decades, the cheapest energy paths are constantly chosen, without considering later time periods or the overall solution. Gas is available for free, so energy paths that start with gas supply are always cheaper than those starting with electricity supply assets. Therefore, these paths are often chosen by the greedy algorithm, and as much gas as possible is used up in the all time periods. The leftover gas can be stored and used in later time periods. In those later time periods, it is still relatively cheap to use stored gas, since the costs only consist of the storage assets, which have a relatively low loss factor. Therefore, these paths involving gas storage are often chosen by the greedy algorithm, and the gas in the storage assets is used up quickly. As can be seen in Figure 7.4, P2G assets are then still needed in the last decade in order to be able to satisfy the gas and heat demand. These P2G assets are relatively very expensive. Therefore, the total costs (objective function value) of the solutions found by the greedy algorithm are much higher than those found by the Gurobi solver.

The question then arises whether including storage in the algorithm has a positive effect on the total costs of the solution or not. Gas that is put in storage is used up quickly in the time periods thereafter, and many P2G assets are still needed in the last time periods. Therefore, in the next section, it is investigated what the effect is of excluding storage from the greedy algorithm is. But first, the solutions found by the greedy algorithm for the larger cases are further analyzed.

The 28 nodes case

In contrast to the solution for the 7 nodes case, no investments in storage assets are made in the solution for the 28 nodes case. This can be seen in Figure 7.5a, where the solution for the 28 nodes case found by the greedy algorithm is plotted. For comparison, the solution found by the Gurobi solver is again plotted next to it in Figure 7.5b. Apparently, in the 28 nodes case, an energy path containing a storage asset is never the cheapest energy path. It was also concluded from Table 7.6 that the objective function value of the solution of the greedy algorithm is much higher than the solution found by the Gurobi solver. An explanation for these two things can be found when investigating the two plots in Figure 7.5.



(a) Solution obtained by the greedy algorithm with the maximum path length set to the best length.

(b) Solution obtained by the Gurobi solver, same as Figure 7.2.

Figure 7.5: Number of investments made in the different asset types in the solution for the 28 nodes case, found by the greedy algorithm (left) and the Gurobi solver (right).

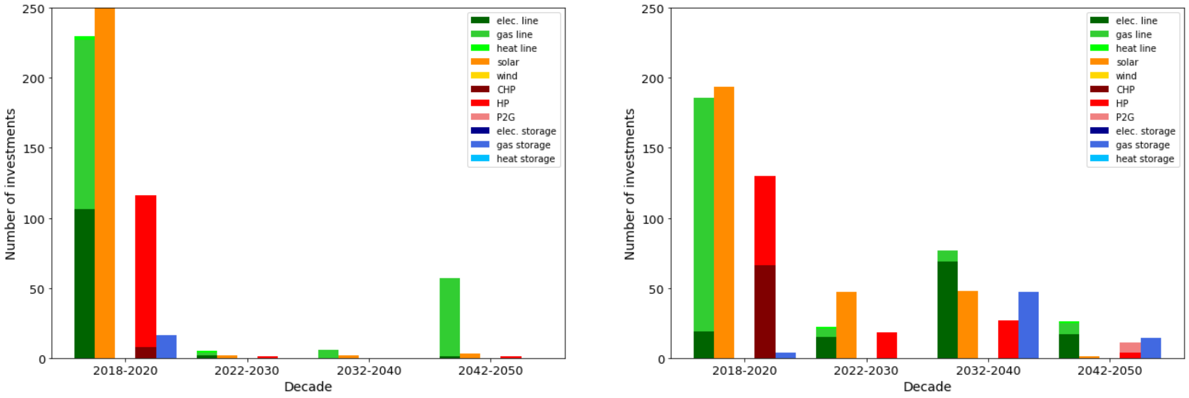
First of all, it can be concluded that the number of investments in wind supply assets, CHPs and HPs is comparable in both solutions. A small difference in number of investments can be observed in the first time period, where more investments in CHPs are made in the greedy solution. Furthermore, a little less HPs are invested in during the first and second decade of the greedy solution. The reason for this is probably also that it is a greedy algorithm, and gas supply has a cost of zero in the case study. Therefore, CHPs are a cheap way to obtain heat, thus energy paths containing CHPs are often selected by the algorithm. This reduces the need for HPs to obtain heat, thus explains the lower number of investments in HPs in the greedy solution.

A big difference between the two figures is the number of investments made in distribution assets, in all four decades. This number is much higher in the greedy solution. Part of the reason is probably that more investments in CHPs are made in the greedy solution. Distribution assets are needed to distribute more gas to these CHPs from nodes where gas supply is available, and to distribute the electricity and heat produced by these CHPs to other nodes. Another reason for the much higher number of investments in distribution assets is again the nature of the greedy algorithm, as explained earlier when analyzing the solution for the 7 nodes case. This higher number of investments in distribution assets is part of the reason why the objective function value of the solution found by the greedy algorithm is higher.

Another big difference between the two figures, which is another reason that the objective function value of the greedy algorithm is higher, is that P2G assets are needed in the last decade to fulfill the gas demand, just like in the solution for the 7 nodes case. In the solution for the 28 nodes case, even more investments in CHPs are made, thus even more gas can be converted. Since it is a greedy algorithm, it is likely that all leftover gas, that is not used to fulfill gas demand, is used for conversion in CHPs. Then there is no gas left to put into storage. This is probably the reason that including storage in the greedy algorithm has no effect on the solution for the 28 nodes case. When there is no gas left in the last decade, P2G assets are needed to fulfill gas demand. These are very expensive, thus this results in a higher total cost of the solution. Moreover, they require electricity, which explains the much higher number of investments in solar supply assets in the solution obtained by the greedy algorithm.

The 110 nodes case

The results of the solution for the 110 nodes case as found by the greedy algorithm are plotted in Figure 7.6a. The solution found by the Gurobi solver is again plotted next to this figure in Figure 7.6b.



(a) Solution obtained by the greedy algorithm with the maximum path length set to the best length. The total number of investments made in solar supply assets in the first decade is 352.

(b) Solution obtained by the Gurobi solver, same as Figure 7.3.

Figure 7.6: Number of investments made in the different asset types in the solution for the 110 nodes case, found by the greedy algorithm (left) and the Gurobi solver (right).

When comparing the results in Figure 7.6, it can be observed that the number of investments in electricity lines is much higher in the solution found by the greedy algorithm. This is probably again an unfortunate result of the design of the greedy algorithm.

Moreover, many more large differences between the two solutions can be observed. In the solution found by the greedy algorithm, the number of investments in CHPs is much lower, as well as the number of investments in gas lines in the first decade, whilst the number of investments in electricity lines, solar supply assets, heat pumps and gas storage are much higher. Moreover, hardly any investments are made in the last three decades, except for some investments in gas lines in the last decade. This all can be explained by the fact that the maximum length of energy paths is set to 2, by the following reasoning.

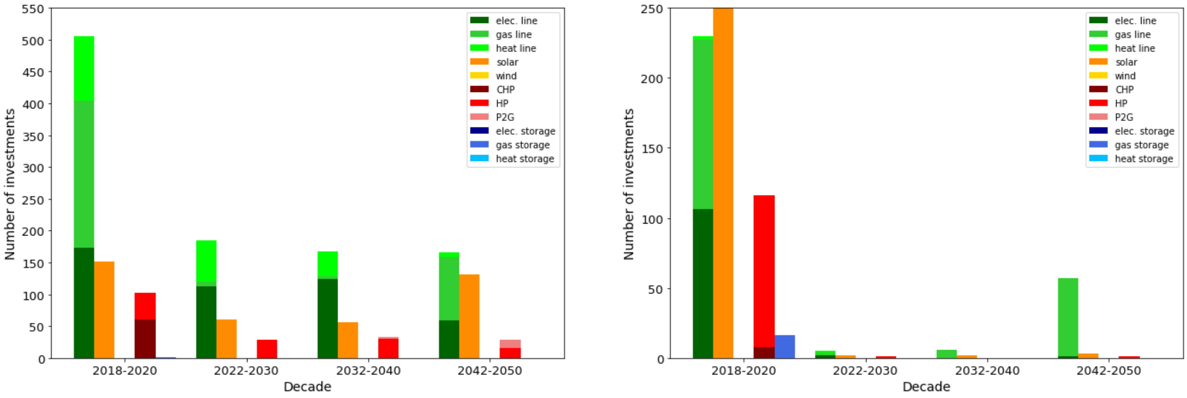
In the solution found by the Gurobi solver, many investments are made in CHPs, as well as in gas distribution assets to distribute the needed gas from nodes where gas supply is available. However, this requires an energy path length of at least 3, since gas is supplied, distributed to a different node, and then converted by a CHP. Therefore, CHPs are only an option for the greedy algorithm when they are placed in the same nodes as where the available gas supply is located. This explains the much lower number of investments in CHPs, and the slightly lower number of investments in gas lines.

In other nodes, the only way to obtain heat is by using HPs. Therefore, the number of investments in HPs is very high, as well as the number of investments in solar supply assets, which are needed to produce electricity for the HPs. Since these assets do not depend on available gas supply, which decreases of the years, they can still be used in the later time periods, thus hardly any more investments need to be made in the last three decades.

Furthermore, since less gas is used in CHPs, more gas is leftover, which can be put into storage and used in later time periods, which explains the higher number of investments in gas storage in the first decade. This is likely also the reason that many investments are made in gas lines in the last decade. These assets are needed to distribute gas taken out from storage assets, which are located in the same nodes as gas supply, to all other nodes.

The reason that this solution has a higher objective function value than the solution found by the Gurobi solver, is probably because the maximum energy path length is so restricted. Therefore, CHPs cannot be invested in at nodes where there is no gas supply, whilst this is a cheaper way to obtain electricity and heat than using solar supply assets and HPs in the first time periods.

What remains interesting, is why the solution found by the greedy algorithm with a maximum path length of 2 is better than when it is set to 3, when storage is included. To be able to compare the two results, the solution found by the greedy algorithm when the maximum path length is set to 3 is given in Figure 7.7a, and the solution for a maximum path length of 2 is again plotted in Figure 7.7b. This solution is entirely different than the solution with a maximum path length of 2. The main difference lies in the fact that investments in CHPs can be made in all nodes, rather than just the nodes where gas supply is available. As CHPs are a good investment in the first time period, many of these investments are made, which changes the entire solution. The reason that this solution is more costly in the end is likely that as much gas as possible is converted in CHPs, thus less gas is left that can be put into storage, thus P2G assets are again needed in the last time period, which are very expensive. This results in a higher total cost.



(a) Solution obtained by the greedy algorithm with the maximum path length set to 3.

(b) Solution obtained by the greedy algorithm with the maximum path length set to the best length. Same as Figure 7.6a. The total number of investments made in solar supply assets in the first decade is 352.

Figure 7.7: Number of investments made in the different asset types in the solution for the 110 nodes case found by the greedy algorithm, with the maximum path length set to 3 (left), and set to 2, the best length (right).

7.2.3. Effect excluding storage

As explained in the previous section when analyzing the solution of the 7 nodes case, excluding storage from the algorithm may have a positive effect on the solution found by the greedy algorithm. In this section, the results of removing storage as an option in the algorithm are given. The results for the 7 nodes case are first plotted in Figure 7.8, to gain insight into the trends and differences. The exact results are given in Table E.1, which can be found in Appendix E. It can be concluded from Figure 7.9 that the greedy algorithm indeed finds a better solution when storage is excluded, and that the needed run time is shorter. This holds for all maximum path lengths. The algorithm finds the best solution when storage is excluded, and the maximum path length is set to 5. Thus, the best maximum path length for the 7 nodes case is 5.

The fact that the algorithm needs less run time when storage is excluded can easily be explained by the fact that there are less possible paths that need to be calculated. The reason that including storage in the greedy algorithm results in even higher total costs can be found when inspecting the barplot of the solution in Figure 7.9, where both the solutions with and without storage are plotted. In this figure, it can be observed that the number of investments for all assets is almost exactly the same, but more distribution assets are invested in when storage is included. Thus, when gas storage is included, costs

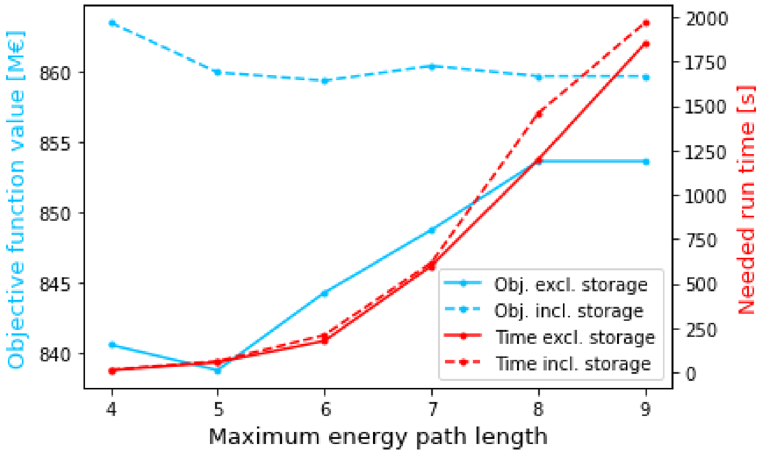
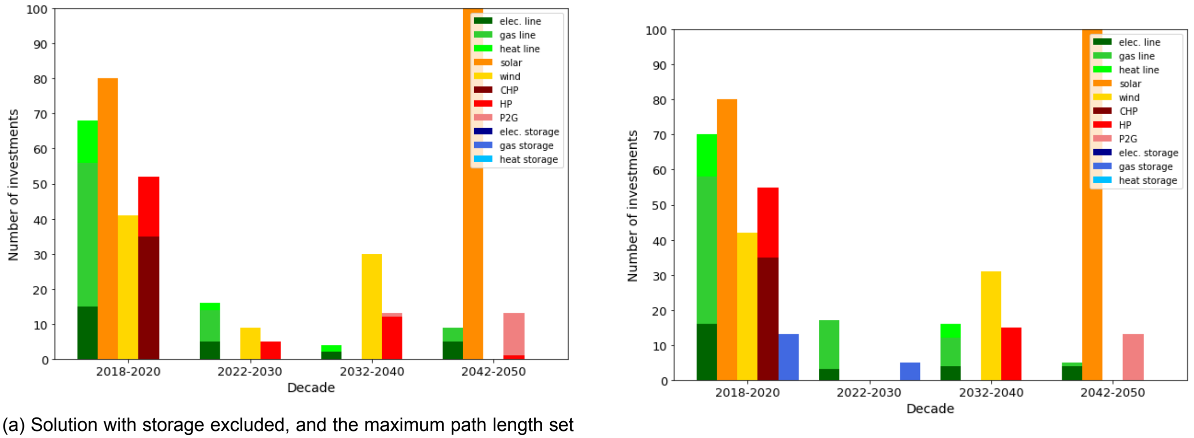


Figure 7.8: Results of the greedy algorithm for different path lengths, with storage included (full line) and excluded (dotted line). The objective function values of the solutions are plotted (in blue), as well as the needed run times (in red).

of investing in these assets and extra needed distribution assets are added to the total costs. However, the gas stored in these assets is used quickly in following time periods, due to the greedy nature of the algorithm. Therefore, the total number of P2G assets needed in the last time periods to fulfill the gas demand is exactly the same. Thus, by including storage, only some costs are saved by being able to store gas and use it in CHPs a little later, but the costs of storage assets and needed distribution assets are added. Eventually, the total costs are higher than when excluding storage, thus it does not have a positive effect on the solution. This is an interesting result, since storage is included in relatively good solutions to the model found by the Gurobi solver.



(a) Solution with storage excluded, and the maximum path length set to 4. The total number of investments made in solar supply assets in the fourth decade is 413.

(b) Solution with storage included, same as Figure 7.4a. The total number of investments made in solar supply assets in the fourth decade is 460.

Figure 7.9: Number of investments made in the different asset types in the solutions for the 7 nodes case, found by the greedy algorithm with the best maximum path length, with storage excluded (left) and included (right).

As concluded in the previous section, storage is not chosen in the solution found by the greedy algorithm for the 28 nodes case. Apparently, an energy path containing a storage asset is never the cheapest energy path. Thus, the best maximum path length is still 3. However, the run time of the algorithm is shorter when storage is not included, namely only 264 seconds instead of 344. Therefore, it is also better to exclude storage of the algorithm for the 28 nodes case.

However, this does not hold for the 110 nodes case, for which the results are shown in Table 7.7. From this table, it can be concluded that the greedy algorithm finds the best solution when storage is included, and the maximum energy path length is set to 2.

Table 7.7: Objective function value of the solution to the 110 nodes case found by the greedy algorithm and the needed run time, for different maximum path lengths, with storage excluded and with storage included.

Max. path length	Obj. excl. storage [M€]	Run time [s]	Obj. incl. storage [M€]	Run time [s]
1	infeasible	-	infeasible	-
2	1134.63	533	1072.65	785
3	1107.02	47,908	1109.74	68,653
4	-	>24 hours	-	>24 hours

In conclusion of the results of the greedy algorithm, the best found solutions are given in Table 7.8, where it is also indicated whether storage should be included or not, and what the best maximum path length is for each case. The fact remains that the greedy algorithm cannot find a better solution than the Gurobi solver in the same amount of time for the 7 and 28 nodes cases. For the 110 nodes case, the Gurobi solver is is able to find a better solution, but it takes around 2000 seconds to find this solution, whereas the greedy algorithm only needs around 800 seconds.

Table 7.8: Parameters for finding the best solution with the greedy algorithm for each case, the objective function value of this best solution and the needed run time.

# nodes	Opt. max. path length	Storage included	Obj. function value [M€]	Run time [s]
7	5	No	838.76	56
28	3	No	769.38	264
110	2	Yes	1072.65	785

7.3. Results column generation

In this section, the performances of the four different applications of column generation on the MES model are evaluated. As concluded in the previous section, the Gurobi solver is able to find better solutions than the greedy algorithm, in (nearly) the same amount of time. Therefore, the performance of the different applications of column generation is evaluated using the Gurobi solver to generate initial columns. First, verifications are done to ensure that column generation was implemented correctly in Section 7.3.1, by running it on small testcases. Then, the four different applications are analyzed in terms of size of the master problems and pricing subproblems in Section 7.3.2, when applied to the case study from Chapter 6.1, as this is very likely to have an effect on the performance of the four applications. The theoretical results of this analysis are also compared to the practical results of the small testcases. These results can explain the performance of the four applications of column generation, which are evaluated thereafter in Section 7.3.3. The results of the four applications on the 7, 28 and 110 nodes cases are given and compared to each other. The improvement in the Relaxed Master Problem (RMP) is investigated, as well as the improvement of the integer solution at the end, when the Integer Master Problem (IMP) is solved. Since the problem is very complex, it is expected that the run time of column generation is very long. Therefore, in Section 7.3.4, the effect of removing unused columns is investigated, as this may lower the computation time.

7.3.1. Verifying performance implementations

Deriving the master and pricing subproblem formulations can be rather difficult for a model as complex as the MES model. The same holds for programming these formulations. Therefore, it is important to verify that the formulations have been derived and programmed correctly. The verifications that were done in this research are explained in this section, as well as the results of these verifications.

Verifying the implementations

The most important indicator that was used in this research to verify that the derivation and implementation of the formulations is correct, is if the column generation algorithm finds the correct solution to a small example. Therefore, two even smaller cases are derived from the 7 nodes case. Both testcases consists of only 3 nodes. Furthermore, the first testcase consists of 2 time periods, and the second of 4 time periods. These cases are constructed in the following way. The demand, supply and location data on nodes 1-3 from the original 7 nodes case are used. Similarly, the demand and supply data on the first two and the first four time periods is used, and all data on the other time periods is removed. The cost, capacity and efficiency data on the assets are the same as in the original 7 nodes case.

The four implementations of column generation are then run on these two testcases, until the objectives of all pricing subproblems is zero. The fact that they are able to achieve a reduced cost of zero for all pricing subproblems, and do not get stuck in a loop, already indicates the no mistakes were made in the derivation of the formulation of the master and pricing subproblems. The Gurobi solver is used to find an initial solution for the column generation algorithm. The results for the first and second testcase are given in Table 7.9 and 7.10, respectively.

Table 7.9: Results of applying all four implementations of column generation to the first testcase, consisting of two nodes and two time periods. The optimal integer solution found by the Gurobi solver is used for the initial columns. The objective function values of the initial solution, the RMP in the last iteration, and the IMP are given.

Column generation	Obj. initial solution [M€]	Obj. RMP [M€]	Obj. IMP [M€]
Time period	182.91	182.91	182.91
Energy carrier	182.91	179.78	182.91
Asset type	182.91	179.78	182.91
Node	182.91	179.78*	182.91

*For the first implementation option. For the other two options, the algorithm did not converge within 24 hours.

Table 7.10: Results of applying all four implementations of column generation to the second testcase, consisting of two nodes and four time periods. The optimal integer solution found by the Gurobi solver is used for the initial columns. The objective function values of the initial solution, the RMP in the last iteration, and the IMP are given.

Column generation	Obj. initial solution [M€]	Obj. RMP [M€]	Obj. IMP [M€]
Time period	183.16	183.12*	183.16
Energy carrier	183.16	179.78	183.16
Asset type	183.16	179.78	183.16
Node	183.16	179.78**	183.16

*Did not converge within 24 hours.

**For the first implementation option. For the other two options, the algorithm did not converge within 24 hours.

Firstly, the objective function value of the initial solution is given in these tables. Since these cases are so small, the Gurobi solver is able to find the optimal solution of the cases within a few seconds, thus the optimal solution is used for the initial columns.

Secondly, the objective function value of the RMP at the end of the algorithm is given in these tables, when the objectives of all pricing subproblems are zero. If column generation is implemented correctly, the expected outcome is that this objective function value is equal to the objective of the relaxed version of the original model, or higher. It can be concluded that this holds for all four of the implementations of column generation, for the first testcase. The objective function value of the RMP of column generation per energy carrier, asset type and node are all equal to the objective function value of the relaxed version of the original model. The same holds when they are applied to the second testcase.

However, the objective function value of the RMP of column generation per time period for the first testcase is equal to the objective function value of the integer solution of the original model. When the optimal solution to the integer problem, found by the Gurobi solver, is used for the initial columns, column generation per time period already stops after one iteration. Thus, column generation per time period can find a better bound for the solution to the original model than the other implementations of column generation. However, for the second testcase, column generation per time period is not able to converge within 24 hours. This means that is not able to find the right columns within 24 hours, such that the objective function value of all pricing subproblems is zero.

The reason that column generation per time period does not converge for the second testcase, but that it does for the first testcase, even when the initial solution is the optimal integer solution, lies in the content of this optimal solution, and nature of the pricing subproblems. In the optimal integer solution to the first testcase, all investments are made in the first time period, and none in the second time period. Due to the capacity constraints that are included only in the pricing subproblem of the first time period, reduced cost of this subproblem is instantly zero. However, in the optimal solution to the second testcase, some investments are made in the third time period. However, the pricing subproblem for the third time period does not contain capacity constraints. Therefore, the reduced cost is not instantly zero. Instead, a new column is generated by this pricing subproblem, and added to the master problem. This changes the values of the dual variables, which eventually should ensure that the algorithm converges. However, it is apparently not able to converge within 24 hours, not even for such a small case.

Column generation per node is not able to converge for either of the testcases when the second or third option for implementation is used, which are described in Section 5.5.2. In the next section, the four applications of column generations are further analyzed, revealing a possible reason for this issue. Thereafter, only the first implementation option for column generation per node is considered, as this option has the most potential to perform well on the larger cases.

7.3.2. Analysis four different implementations

Before investigating the result of implementing column generation, something can already be said about how the case study affects the four different implementations. For the performance of column generation, the number of pricing subproblems and the size of the master problem in terms of number of constraints is important, as well as the difficulty of the joint constraints. Therefore, these three aspects are analyzed for all four implementations of column generation, when applied to the case study of Eindhoven.

First, the total number of joint constraints for each implementation of column generation and for each case is determined. The calculations can be found in Appendix D, and the results are given in Table 7.11.

Table 7.11: Number of constraints in the master problem of each implementation of column generation, when applied to the case study of Eindhoven, for each number of nodes.

	Time period	Energy carrier	Asset	Node
Formula dependent on $\|V\|$	$17 + 128\|V\| + 48\|V\|^2$	$3 + 51\|V\|$	$11 + 51\|V\|$	$52\|V\|$*
# nodes	# joint constraints			
7	3,265	360	368	364*
28	41,233	1,431	1,439	1,456*
110	594,897	5,613	5,621	5,720*

*For the first implementation option. For the second implementation option, the formula for the total number of joint constraints is $|V| + 51|V|^2$, and the actual numbers of joint constraints are 2,506, 40,012 and 617,210. For the third option, the total number is $-24.5|V| + 76.5|V|^2$ resulting in 3,577, 59,290 and 922,955 joint constraints.

From Table 7.11, it can be concluded that the master problem of column generation per time period contains far more joint constraints than the other implementations, for all cases. This is partly due to the fact that the number of joint constraints depends quadratically on the number of nodes, whereas this relation is linear for the other implementations. This is most likely the reason for the poor performance of column generation per time period for the second testcase, as shown in the previous section. The same holds for the second and third options for implementation for column generation per node. For the second implementation option, some of these joint constraints are equality constraints. The second implementation option only contains more joint constraints than column generation per time period for the largest case, but a little less for the smaller cases. However, it contains far more joint constraints than column generation per energy carrier and per asset type. The third implementation option consists of the most joint constraints over all applications for every case.

The number of pricing subproblems in each iteration for different implementations of column generation, when applied to the case study of Eindhoven, for each number of nodes, is given in Table 7.12. From this table, it can be concluded that column generation per energy carrier has the lowest number of pricing subproblems. Column generation per time period has the highest number of pricing subproblems for the 7 nodes case. However, for the larger cases, column generation per node has the most pricing subproblems by far, since this number is dependent on the number of nodes.

Table 7.12: Number of pricing problems per iteration for each implementation of column generation, when applied to the case study of Eindhoven, for each number of nodes.

	Time period	Energy carrier	Asset	Node
Formula	$\|T\|$	$\|E\|$	$\|A\|$	$\|V\|$
# nodes	# pricing subproblems			
7	17	3	12	7
28	17	3	12	28
110	17	3	12	110

The number of pricing subproblems, and the complexity of these subproblems, is likely to have an effect on the duration and performance of the column generation algorithm. This effect can indeed be seen, when investigating the number of iterations and the run time that the algorithms need to converge for the testcases. These results are given in Table 7.13 and Table 7.14, respectively. When disregarding the results of column generation per time period, it can be concluded that column generation per asset type needs the least iterations to converge. This is probably the result of the larger number of pricing subproblems. Furthermore, it can be concluded that column generation per asset type needs much less time for the larger testcase than the other applications of column generation. Therefore, column generation per asset type has the most potential to perform well on the larger cases consisting of 7, 28 and 110 nodes. The results of the performance of all four applications of column generation on these larger cases are given in the next section.

Table 7.13: Number of iterations that column generation needs to converge for the test cases.

Testcase	Time period	Energy carrier	Asset	Node
1	1	59	22	79**
2	-*	326	60	367**

*Did not converge within 24 hours.

**For the first implementation option, the other implementation options did not converge within 24 hours.

Table 7.14: Amount of time [s] that column generation needs to converge for the test cases.

Testcase	Time period	Energy carrier	Asset	Node
1	0.7	44	57	73*
2	>24 hours	575	177	738*

*For the first implementation option, the other implementation options did not converge within 24 hours.

7.3.3. Results four applications

The results of the four implementations of column generation on the MES model are presented in this section. The Gurobi solver produces a better initial solution than the greedy algorithm, within the same amount of time. However, since it is interesting how column generation performs when the initial solution is much worse, both methods are used to generate a feasible solution for the initial columns. The tests are run for 23 hours on the supercomputer, on the 7, 28, and 110 nodes cases. First, the results of column generation per time period are discussed. At the end, the results of the four applications are compared and discussed.

One remark on column generation applied to the 110 nodes case should be made beforehand. Due to the many complicated computations and very large or very small numbers, the algorithm runs into issues for the 110 nodes case. The solution that is produced by the Gurobi solver as an initial solution for the column generation algorithm cannot be used in practice. The column generation algorithm gives an error in the first iteration that the master problem with the initial columns is infeasible, due to rounding mistakes in the computations. This same issue is found when applying the other column generation methods to the 110 nodes case. Even when all numbers in the data are rounded to 4 significant digits, and the initial solution is rounded to 4 significant digits, this problem still occurs. When the greedy algorithm is used to generate an initial solution, this is not an issue. The same holds for the larger cases.

Column generation per time period

As discussed in the previous section, column generation per time period is not able to converge for the testcase consisting of three nodes and four time periods. Therefore, it is unlikely that it performs well for larger cases. And indeed, the column generation algorithm per time period performs even worse for the larger cases. Regardless of the method that was used to obtain the initial solution for the 7 nodes case, the very first pricing subproblem cannot be solved to optimality within 24 hours on the supercomputer. Thus, it is not able to find any new columns, let alone find a lower objective function value for the RMP or IMP.

Column generation per energy carrier

The results for column generation per energy carrier are given in Table 7.15, when it is run for 23 hours on the supercomputer. Though column generation per energy carrier does converge when applied to the testcases, it is not able to converge within 23 hours on the supercomputer. It can be observed that it is not able to find a significantly better solution to the RMP than the initial solution for any of the cases. This even holds when the greedy algorithm is used to obtain an initial solution, though this solution is much worse. No better integer solution is found either, when solving the IMP with all the generated columns at the end of the algorithm. Therefore, no barplots of the solutions are investigated.

Table 7.15: Results of applying column generation per energy carrier to the cases and running it for 23 hours, using both the Gurobi solver and the greedy algorithm to find a feasible solution for the initial columns. The objective function values of the initial solution, the RMP in the last iteration, and the IMP at the end are given.

Method init. sol.	# nodes	Obj. initial sol. [M€]	Obj. RMP [M€]	Obj. IMP [M€]
Gurobi	7	711.50	711.50	711.50
	28	658.82	658.82	658.82
	110	-	-	-
Greedy	7	838.76	838.76	838.76
	28	769.38	769.38	769.38
	110	1072.65	1072.65	1072.65

Table 7.16: Results of applying column generation per asset type to the cases and running it for 23 hours, using both the Gurobi solver and the greedy algorithm to find a feasible solution for the initial columns. The objective function values of the initial solution, the RMP in the last iteration, and the IMP at the end are given.

Method init. sol.	# nodes	Obj. initial sol. [M€]	Obj. RMP [M€]	Obj. IMP [M€]
Gurobi	7	711.50	709.85	711.50
	28	658.82	658.82	658.82
	110	-	-	-
Greedy	7	838.76	736.19	838.76
	28	769.38	769.38	769.38
	110	1072.65	1072.65	1072.65

Column generation per asset type

Column generation per asset type is, however, able to lower the objective function value for the RMP significantly. The results of column generation per energy carrier are given in Table 7.16. It can be concluded from this table that column generation per asset finds a lower objective function value for the 7 nodes case within the maximum run time. However, it did not converge within 23 hours, thus no better lower bound to the problem is found. Moreover, no better solution to the IMP is found within 23 hours either.

For the 28 nodes case, column generation per asset type is unable to find a solution with a lower objective function value for the RMP, regardless of the method used to produce the initial solution. For the 110 nodes case, column generation per asset type is also unable to find a solution with a lower objective function value for the RMP, when the greedy algorithm is used for the initial solution.

Column generation per node

Similar to column generation per energy carrier, the first implementation option for column generation per node is not able to find a significantly better solution to the RMP than the initial solution, for any of the cases, as can be observed in Table 7.17. No better integer solution is found either, when solving the IMP with all the generated columns at the end of the algorithm.

One additional remark on these results should be made. In Section 5.5.2, it was explained that the first implementation option may result in a different optimal solution, since the constraints are different. And indeed, the initial solution that the Gurobi solver generates for this case within the maximum time is slightly different than for the other cases. Remarkably, the found objective function value is slightly lower. The fact that the problem is different, also means the greedy algorithm could not be used to generate an initial solution as it is. This would be possible with slight adaptations, but it is not likely to change the performance of column generation per node within 24 hours. Therefore, it is not investigated in this research.

Table 7.17: Results of applying the first implementation option for column generation per node to the cases and running it for 23 hours, using the Gurobi solver to find a feasible solution for the initial columns. The objective function values of the initial solution, the RMP in the last iteration, and the IMP at the end are given.

Method init. sol.	# nodes	Obj. initial sol. [M€]	Obj. RMP [M€]	Obj. IMP [M€]
Gurobi	7	701.63	701.63	701.63
	28	655.63	655.63	655.63
	110	-	-	-

Comparison results four applications

It can be concluded that column generation per asset type shows the most potential out of the four applications, as it is the only one able to lower the objective function value of the RMP within 24 hours on the supercomputer. The results also raise several questions. Firstly, why does this application perform the best? Secondly, how can the performance of the applications, especially for the larger cases, be improved? And thirdly, why is no better integer solution found by any of the applications, for any of the cases?

The reason why column generation per asset type performs the best, can likely be explained by the results found in Section 7.3.2. The number of joint constraints is low for both column generation per energy carrier and per asset type, but the number of pricing subproblems is higher for column generation per asset type. The effect of the number of joint constraints and pricing subproblems can be investigated by inspecting the average computation time of both, for all four applications, applied to all three cases. The results are given in Table 7.18, where the greedy algorithm is used to obtain the initial solution, and the column generation algorithms are run on the supercomputer for one hour. The average computation time of one pricing subproblem is given, as well as the average computation time of all pricing subproblems together in one iteration. This is equal to the average computation time of one pricing subproblem, times the number of subproblems as given in Table 7.12. The results of column generation per time period are excluded, since it was already unable to solve the first pricing subproblem to optimality.

Note that the average computation time of the master problem is higher for the smaller cases. The reason being that the master problem takes longer to solve when more columns are added to the solution space. For the smaller cases, more iterations of the algorithm are performed in one hour than for the larger cases. Therefore, more columns have been added to the solution space at the end of the hour for the smaller case, thus the master problem takes longer to solve.

It can be concluded that the average computation time of the master problem is the lowest for column generation per energy carrier. For the larger cases, column generation per asset type is a close second. Column generation per energy carrier has the lowest number of joint constraints, as well as the lowest number of columns added in each iteration, since it has the lowest number of pricing subproblems. This explains the low computation time of the master problem.

The computation time per pricing subproblem is the lowest for column generation per node. This can be explained by the fact that it has the most pricing subproblems, so the same constraints are divided over more pricing subproblems, making the pricing subproblems less complicated to solve. However, as a result, the total time that it takes to solve all pricing subproblems in one iteration of the column generation algorithm is extremely high for the larger cases, compared to the other two applications of column generation.

Table 7.18: Average computation time [s] of the master problem and the pricing subproblems (per individual problem and all problems in one iteration together) for column generation per energy carrier, per asset type and per node, for all three cases. The greedy algorithm is used to obtain the initial solution, and the test are run for one hour on the supercomputer.

Avg. comp. time of	# nodes	Energy carrier	Asset type	Node
Master problem	7	0.71	1.57	1.76
	28	0.29	0.37	3.85
	110	0.45	0.46	5.02
1 pricing subproblem	7	0.43	0.32	0.29
	28	2.79	1.44	1.03
	110	38.21	16.34	10.75
All pricing subproblems	7	1.29	3.85	2.04
	28	8.37	17.27	28.71
	110	114.64	196.04	1182.42

The computation time of all pricing subroblems in one iteration is around 1.7-3 times higher for column generation per asset type than for column generation per energy carrier, depending on the case. However, column generation per asset type produces 12 columns in each iteration, which is 4 times more columns than column generation per energy carrier. This explains why column generation per asset type performs the best.

To improve the performance of the applications, especially for the larger cases, a reduction of the computation time of the master and pricing subproblems is needed. An additional method can be used to decrease the computation time of the master problem, namely removing unused columns. This method is further explained and applied in the next section. It is only tested on column generation per asset type, since this is the only application of column generation that is able to find columns that result in a lower objective function value to the RMP. A solution for decreasing the computation time needed to solve the pricing subproblems, could be to not solve the pricing subproblems to optimality. However, this likely increases the needed number of iterations to converge, thus its effect on the overall solution is uncertain. Therefore, this method is not applied in this thesis, but left as a suggestion for further research.

The other question that was raised in this section regards the method of obtaining an integer solution combined with column generation in this research. No better integer solution is found for the 7, 28 and 110 nodes cases by solving the integer master problem with all the generated columns at the end of the algorithm. Interestingly, this also holds for the testcases, which were described in Section 7.3.1. When the greedy algorithm is used to generate an initial solution for the two testcases, columnn generation per energy carrier, asset type and node are all able to converge within a short amount of time (max. 15 minutes). However, no better integer solution can be found with all the generated columns. Thus, it can be concluded that the chosen method to obtain an integer solution from column generation does not work for these cases for the MES model. The reason must be related to the columns that are produced by column generation, and the joint constraints. Apparently, no combination of new columns, where one column is selected per subproblem, satisfies the joint constraint. It is currently unclear why the produced columns are not suitable. Further research could investigate this problem. Adding more valid inequalities to the pricing subproblems may improve the suitability of the produced columns.

7.3.4. Results removing unused columns

An additional method that can be used to reduce the computation time of column generation, is removing unused columns, of which the effect is analyzed in this section. A column is not used in the RMP when it has a value of zero in the optimal solution to the RMP. The idea is to remove columns from the solution space, when they have not been used for a certain number of iterations. This number of iterations can attain different values, and is given as a parameter I to the algorithm. The aim is to lower the computation time of the master problem, and thus speeds up the algorithm. In Table 7.19, the results of removing unused columns for column generation per asset type after different numbers of iterations for the 7 nodes case are given, where the Gurobi solver is used to generate the initial solution. To get an impression of the performance, the value of I is set to values from 20 to 100, with steps of 5 or 10.

Table 7.19: Results of removing unused columns after different numbers of iterations I for column generation per asset type on the 7 nodes case, when running it for 23 hours. The Gurobi solver is used to generate the initial solution. The total number of iterations is given, as well as the objective function value of the initial solution and the objective function value of the RMP at the end.

I	# iterations	Obj. initial sol. [M€]	Obj. RMP [M€]
20	8569	711.50	711.50
25	5623	711.50	704.39
30	4022	711.50	703.35
35	2972	711.50	703.45
40	3191	711.50	709.60
50	2360	711.50	709.80
75	2360	711.50	709.81
100	2360	711.50	709.82

From Table 7.19 it can be concluded that the algorithm can perform many more iterations within the same amount of time when columns are removed. And the sooner columns are removed, meaning the lower the value of I, the more iterations it can perform. This also has the desired effect on the found objective function value of the solution to the RMP. This value is lower when columns are removed. After 23 hours on the used supercomputer, the lowest objective function value is found when columns are removed after 30 iterations. Note that removing columns after 26-34 iterations may result in an even better solution. Moreover, the algorithm has not yet converged, thus no better lower bound is found. No better integer solution was found either, when solving the IMP at the end.

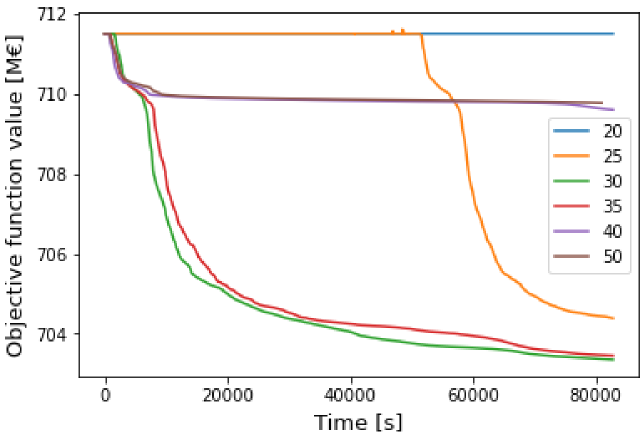


Figure 7.10: Results of removing unused columns after different numbers of iterations for the 7 nodes case, where the Gurobi solver is used to generate the initial solution.

The performance over 23 hours is also plotted in Figure 7.10 for several different values for I. From this figure it can be concluded that the number of iterations after which an unused column is removed makes a big difference for the outcome. If this number is too low, no improvement can be made in the objective function value. If this number is too high, the algorithm also gets stuck at a certain objective function value at some point. If this number is just right, the algorithm performs much better.

It is interesting to investigate if it is always best to remove columns after 30 iterations for this case, independent of the initial solution. Therefore, it is now analyzed what the effect of removing unused columns is, when the greedy algorithm is used to generate the initial solution. The results are shown in Table 7.20. To get an impression of the performance, the value of I is set to values from 25 to 140, with steps of 10 or larger.

Table 7.20: Results of removing unused columns after different numbers of iterations I for column generation per asset type on the 7 nodes case, when running it for 23 hours. The greedy algorithm is used to generate the initial solution. The total number of iterations is given, as well as the objective function value of the initial solution and the objective function value of the RMP at the end.

I	# iterations	Obj. initial sol. [M€]	Obj. RMP [M€]
25	8569	838.76	838.76
50	4022	838.76	741.11
60	3191	838.76	725.60
70	2360	838.76	727.16
80	2360	838.76	718.90
90	2360	838.76	719.81
100	2360	838.76	719.77
140	2239	838.76	723.05

Note again for the results in Table 7.20 that the algorithm has not yet converged, and no better integer solution was found when solving the IMP at the end. It can be concluded from this table that column generation per asset type using the greedy algorithm for the initial solution finds the best solution after 23 hours on the supercomputer when unused columns are removed after 80 columns. Note that removing columns after 71-89 iterations may result in an even better solution. This is a different outcome than when the solution obtained by the Gurobi solver was used, thus the best value of I depends on the initial solution.

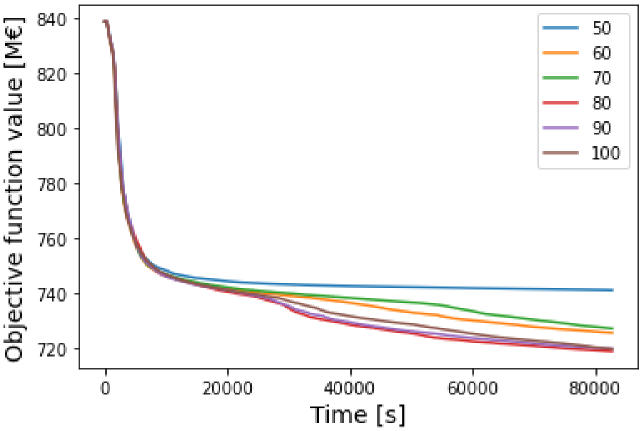


Figure 7.11: Results of removing unused columns after different numbers of iterations for the 7 nodes case, where the Gurobi solver is used to generate the initial solution.

In Figure 7.11, the performance over 23 hours is again plotted for several different values for I, when the greedy algorithm is used for the initial solution. From this figure, it can also be concluded that if I is too low, the algorithm gets stuck at a certain objective function value at some point. If this number is just right, the algorithm performs better. However, the results for I=70 is not in line with the other results. The objective function at the end is expected to be a little lower, such that it falls in between those belonging to I=60 and I=80. This unexpected outcome can be the result of small inconsistencies in the performance of the supercomputer. Therefore, for a more consistent result, it would be better run all tests several times, before determining what value for I results in the best performance.

In order to evaluate the effect of removing unused columns for the 28 nodes case, tests are run with I set to values from 25 to 150, with steps of 25. These numbers are based on the total number of iterations in 23 hours when no columns are removed, which is around 400. When using the greedy algorithm to obtain the initial solution, no lower objective function value to the RMP or IMP is found within 24 hours. When using the Gurobi solver to obtain the initial solution, a strange problem occurs. After a, seemingly random, number of iterations, the master problem is suddenly infeasible. This only happens when I is set to 50, 75, 100, 125 and 150, but not when it is set to 25, or when no unused columns are removed.

This problem does not occur for the 110 nodes case. Similar as for the 28 nodes case, tests are run with I set to values from 25 to 150, with steps of 25, using both the greedy algorithm and the Gurobi solver to generate an initial solution. No lower objective function value to the RMP or IMP is found within 24 hours.

8 Conclusion and discussion

The aim of this research was to investigate whether column generation can improve the quality of the solution to the MES model, found by the Gurobi solver, withing a certain amount of time. To this end, four different applications of column generation to the MES model were formulated in Chapter 5. In addition, a greedy algorithm was developed in Chapter 4, which could be used to generate an initial solution to the MES model for column generation. In Chapter 6, the case study of Eindhoven was introduced, which was used to test the performance of the methods on. The results of these tests were given in Chapter 7. These results are concluded in Section 8.1, and discussed in Section 8.2. Some directions for future research are also given in this section.

8.1. Conclusions

The conclusions that can be drawn from Chapter 7 are the following.

Gurobi solver
It is concluded in Section 7.1.1 that the Gurobi solver is able to find a relatively good solution for the smaller cases. However, for the 110 nodes case, the found optimality gap after 23 hours is 30%. Furthermore, the objective function value of the solution that the Gurobi solver finds for the 7 nodes case is much lower than for the 28 nodes case. The likely reason for this is discussed later in Section 8.2.

Greedy algorithm
In Section 7.2, it is concluded that the Gurobi solver can find a better feasible integer solution for the MES model for the 7 and 28 nodes cases than the greedy algorithm within the same amount of time. It takes a more time for Gurobi to find a feasible solution for the 110 nodes case than for the greedy algorithm, but the objective function value is again much lower. Thus, the Gurobi solver is better suited for producing initial columns for column generation. It is also concluded that excluding storage as an option from the greedy algorithm improves the solution found for the 7 nodes case. Furthermore, setting the maximum path length to its largest possible value does not necessarily result in the best solution.

Column generation
From all the results in Section 7.3, it can be concluded concluded that column generation per asset type performs the best out of the four different applications. It is the only application of column generation that can find a significantly better solution for the 7 nodes case within 24 hours for the relaxed master problem. This is likely because of the low number of joint constraints in the master problem, which is only linearly dependent on the number of nodes. Moreover, it contains more pricing subproblems than column generation per energy carrier. Therefore, the subproblems are easier to solve, and more columns are produced in each iteration. This benefits the number of needed iterations to converge.

Thus, a general conclusion that can be taken from this research is that column generation performs better when the number of joint constraints is lower, and the number of pricing subproblems is higher. More specifically for large cases, column generation performs better if the number of joint constraints is not quadratically dependent on the number of nodes. This can be concluded from the fact that column generation per time period and the second and third implementation option for column generation per node perform very poorly. Column generation also performs better for the larger cases if the number of pricing subproblems is higher, but not dependent on the number of nodes. This can be concluded from the fact that column generation per node performs worse than column generation per asset type.

Removing columns after a certain number of iterations further improved the run time of column generation per asset type. This effect is different, depending on which initial solution is used for the initial columns.

The best result found in this research for the 7 nodes case was found by column generation per asset type, when the Gurobi solver is used to generate the solution for the initial columns, and when columns are removed after being unused for 30 iterations. However, it was still not able to converge for the 7 nodes case within 24 hours on the supercomputer, let alone converge for the larger cases. Thus, it was not able to find a better lower bound for the optimal solution, regardless of what method is used to generate the initial solution.

Since it was not able to converge within 24 hours, it is not possible to apply branch-and-price to find a better integer solution. Solving the integer master problem at the end of the algorithm also did not result in a better integer solution. Thus, no better integer solution to the MESDP was found in this research.

8.2. Discussion and recommendations

Some points of discussion and recommendations for future research are now elaborated upon.

Case study

The objective function value of the solution that the Gurobi solver finds for the 7 nodes case is much lower than for the 28 nodes case. This is likely due to the fact that the constraints on the number of investments of CHPs in each node are currently set to the same number, in both of the cases. Therefore, the maximum number of investments that can be made in CHPs in one time period in the 28 nodes case is much higher. Since CHPs are a good investment in the first time period, this results in lower total costs. The same holds for investments in wind supply assets. In further research, it is recommended that these constraints are adjusted for the different cases, such that the total number of investments that can be done in CHPs and wind supply assets is the same for each case, regardless of the number of nodes.

Both the Gurobi solver and greedy algorithm found that the objective function value of the solution for the 110 nodes case is much higher than for the other two cases. This means that the total costs for the 110 nodes case is higher. The reason is likely related to the capacities of the assets considered in the case study. The demand in each node is much smaller in the 110 nodes case than in the smaller cases, since the same amount of demand is divided over more nodes. Therefore, the capacity of the assets does not match the demand in the 110 nodes case in the same way as in the smaller cases. The assets are more likely to have an overcapacity. Thus, regarding assets with smaller capacities for the 110 nodes case in future research is likely to result in less overcapacity, and lower total costs.

Greedy algorithm

The fact that the total costs of the solution found by the greedy algorithm are higher, especially when storage is included for the 7 nodes case, is likely caused by the greedy nature of the algorithm, and the fact that gas is free to use in the case study but has limited availability, and P2G assets are very expensive. Since the greedy algorithm makes its decision based on the cheapest cost, all available gas is used up for the gas demand and producing heat with CHPs in the first time periods. Then, no gas is left for the last time periods, not even when storage is included in the greedy algorithm. This results in a need for P2G assets in the last time periods, which are very expensive and thus lead to a high total cost. In order to improve the greedy algorithm, further research can aim to change this. This research should investigate how a greedy algorithm can use a limited available resource better, such as gas in this case. When the algorithm decides whether it uses a limited resource in a certain time period or not, the effect on future time periods should be taken into account.

Another suggestion for further research on the greedy algorithm involves the needed run time. The solutions to the 28 and 110 nodes cases cannot be found within 24 hours on the used supercomputer, when the maximum energy path length is set to 5 or 4, respectively, or higher. A higher maximum energy path length may result in a better objective function value of the solution. This motivates research into speeding up the algorithm, by making it more efficient, or into running the algorithm for longer than 24 hours.

Column generation

Out of the four different applications, column generation per asset type has the most potential to be able to find a better lower bound or integer solution. It is the only column generation application that is able to find a lower objective function value for the relaxed master problem for the 7 nodes case. Therefore, it is recommended that future research into finding a better solution to the MES model for large cases focuses on this application of column generation.

The long computation time of column generation, when applied to the MES model, is currently a limiting factor for all cases, but especially for the 28 and 110 nodes cases. To be able to investigate the performance on these large cases better, a longer allowed run time on a supercomputer is desired. Moreover, parallel computing could be used to optimize the different pricing subproblems in one iteration simultaneously.

Removing unused columns successfully increased the speed of the algorithm, by decreasing the needed computation time of the master problem. Another method that can be applied in future research is not solving the pricing subproblems to optimality, but setting a run time or optimality gap limit. This may also improve the performance of the column generation algorithm.

Future research could also investigate why solving the integer master problem with the generated columns at the end did not result in a better integer solution in this research. No better integer solution was found for any of the cases, not even for the testcases, regardless of what method was used for the initial solution. The addition of valid inequalities to the pricing subproblems may improve the suitability of the produced columns for a better integer solution.

Lastly, some errors can be investigated in future research, such as the problem of infeasibility when the Gurobi solver is used to generate an initial solution for the 110 nodes case. Future research can investigate the rounding issues further. Furthermore, the infeasiblity problem in the 28 nodes case, when using the Gurobi solver as an initial solution solution and removing unused columns, can be further investigated.

Bibliography

Ayre, J. (2015). *Another low-solar-price record: Saudi electric company lands solar ppa under 5/kwh* (CleanTechnica, Ed.). https://tinyurl.com/gspko7v

Balinski, M. (1965). Integer programming: Methods, uses, computations. *Management Science*, *12*(3), 253–313. https://EconPapers.repec.org/RePEc:inm:ormnsc:v:12:y:1965:i:3:p:253-313

Bonenkamp, N. (2020). *Designing multi-energy systems* (Master's thesis). TU Delft. http://resolver.tudelft.nl/uuid:85050fef-cb23-45ab-a5fd-af505963ae91

Dantzig, G. B., & Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research*, *8*(1), 101–111.

(DHPC), D. H. P. C. C. (2022). DelftBlue Supercomputer (Phase 1) [https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase1].

Enexis open data [https://www.enexis.nl/over-ons/open-data]. (2022).

Flores-Quiroz, A., Palma-Behnke, R., Zakeri, G., & Moreno, R. (2016). A column generation approach for solving generation expansion planning problems with high renewable energy penetration. *Electric Power Systems Research*, *136*, 232–241. https://doi.org/10.1016/j.epsr.2016.02.011

Fraiture, J. (2020). *The robustness of energy systems* (Master's thesis). TU Delft. http://resolver.tudelft.nl/uuid:84cd083a-8bf8-4931-a75a-276e2d54c5cc

Gong, P. (1996). *Capacitated network design with column generation* (Doctoral dissertation). Georgia Institute of Technology.

Gurobi - the fastest solver [https://www.gurobi.com/]. (2022).

(IEA), I. E. A. (2022). Iea sankey diagram [www.iea.org/sankey].

Klose, A., & Görtz, S. (2004). An exact column generation approach to the capacitated facility location problem. https://doi.org/10.1007/978-3-642-17020-1_1

L. R. Ford, J., & Fulkerson, D. R. (1958). A suggested computation for maximal multi-commodity network flows. *Management Science*, *5*(1), 97–101. https://EconPapers.repec.org/RePEc:inm:ormnsc:v:5:y:1958:i:1:p:97-101

Municipality of Eindhoven. (2016). *Eindhoven Klimaatplan 2016-2020* (tech. rep.) [In Dutch]. Municipality of Eindhoven. Eindhoven, the Netherlands.

Municipality of Eindhoven. (2020). *Klimaatplan 2021-2025 - Naar een klimaatneutraal Eindhoven* (tech. rep.) [In Dutch]. https://www.eindhovenduurzaam.nl/sites/default/files/2021-01/WEBTX_Klimaatplan_0.pdf

Muts, P., Bruche, S., Nowak, I., Wu, O., Hendrix, E. M. T., & Tsatsaronis, G. (2021). A column generation algorithm for solving energy system planning problems. *Optimization and Engineering*. https://doi.org/10.1007/s11081-021-09684-2

Paris agreement [UNTC XXVII 7.d]. (2015, December 12). https://treaties.un.org/pages/ViewDetails.aspx?src=TREATY&mtdsg_no=XXVII-7-d&chapter=27&clang=_en

Pyomo. (2022). About pyomo [http://www.pyomo.org/about].

Rahmaniani, R., & Ghaderi, A. (2015). An algorithm with different exploration mechanisms: Experimental results to capacitated facility location/network design problem. *Expert Systems with Applications*, *42*(7), 3790–3800.

Stant, T. (2019). *Decomposition methods for the design of an integrated energy infrastructure* (Master's thesis). Erasmus University Rotterdam.

Thio, L. (2020). *Decision making under uncertainty for multi-energy systems* (Master's thesis). Erasmus University Rotterdam.

van Beuzekom, I. (2022). *Optimizing investment planning of integrated multi-energy systems: To support urban decision makers design an energy transition pathway* (Doctoral dissertation) [Proefschrift.]. Electrical Engineering. Eindhoven University of Technology.

van Beuzekom, I., Gibescu, M., & Slootweg, J. (2015). A review of multi-energy system planning and optimization tools for sustainable urban development. https://doi.org/10.1109/PTC.2015.7232360 van Beuzekom, I., Hodge, B., & Slootweg, H. (2021). Framework for optimization of long-term, multi-period investment planning of integrated urban energy systems [Publisher Copyright: © 2021 The Authors Copyright: Copyright 2021 Elsevier B.V., All rights reserved.]. *Applied Energy*, *292*. https://doi.org/10.1016/j.apenergy.2021.116880

van Beuzekom, I., Mazairac, L., Gibescu, M., & Slootweg, J. (2016). Optimal design and operation of an integrated multi-energy system for smart cities [2016 IEEE International Energy Conference, EnergyCon 2016, EnergyCon 2016 ; Conference date: 04-04-2016 Through 08-04-2016]. https://doi.org/10.1109/ENERGYCON.2016.7514030

van Beuzekom, I., Pinson, P., Gibescu, M., & Slootweg, J. (2017). Optimal planning of integrated multi-energy systems [12th IEEE PES PowerTech Conference, PowerTech 2017 ; Conference date: 18-06-2017 Through 22-06-2017]. https://doi.org/10.1109/PTC.2017.7980886

van der Veen, G. M. (2021). *Design of a short-term weather effects integration module for a multi-energy system* (Master's thesis). University of Groningen.

van Kampen, F. N. (2022). *Influence of uncertain weather conditions on optimizing a multi-energy system* (Master's thesis). University of Groningen.

A

Notation overview

An alphabetical overview of the notation of all parameters and variables introduced in Chapter 3:

A	set of all assets
B_{avt}	number of investments done in asset $a \in A$ at node $v \in V$ in time period $t \in T$
B_{dut}	number of investments done in distribution asset $d \in DA$ on edge $u \in U$ in time period $t \in T$
c_{avt}	costs representing technological development of asset $a \in A$ at node $v \in V$ in time period $t \in T$
c_{dut}	costs representing technological development of distribution asset $d \in DA$ on edge $u \in U$ in time period $t \in T$
d_{evt}	demand of energy carrier $e \in E$ in node $v \in V$ in time period $t \in T$
DA	set of all distribution assets
DA_e	set of all distribution assets distributing energy carrier $e \in E$
DU	set of directed edges $u = (v, v')$ with $v, v' \in V$
E	set of energy carriers
F_{edut}	amount of energy carrier $e \in E$ that is distributed on directed edge $u \in DU$ using distribution asset $d \in DA_e$ in time period $t \in T$
M_{emvt}	amount of energy carrier $e \in E$ that is converted in node $v \in V$ using conversion asset $m \in MA_e$ in time period $t \in T$
MA	set of all conversion assets
MA_e	set of all conversion assets converting energy carrier $e \in E$
N	maximum number of investments in assets of one type, at one node/edge, in one time period
o^{fix}_{avt}	fixed cost of opening an asset $a \in A \setminus DA$ at node $v \in V$ in time period $t \in T$
o^{fix}_{dut}	fixed cost of opening a distribution asset $d \in DA$ on edge $u \in U$ in time period $t \in T$
o^{var}_{avt}	variable cost of using an asset $a \in SA \cup MA$ at node $v \in V$ in time period $t \in T$
o^{var}_{dut}	variable cost of using a distribution asset $d \in DA$ on edge $u \in U$ in time period $t \in T$
$o^{+,var}_{wvt}$	variable cost of charging energy to asset $w \in WA$ at node $v \in V$ in time period $t \in T$
$o^{-,var}_{wvt}$	variable cost of discharging energy from asset $w \in WA$ at node $v \in V$ in time period $t \in T$
S_{esvt}	amount of energy carrier $e \in E$ that is supplied in node $v \in V$ using supply asset $s \in SA_e$ in time period $t \in T$
SA	set of all supply assets
SA_e	set of all supply assets supplying energy carrier $e \in E$

Continued on the next page...

Continuation:

T	set of time periods
t_0	the first time period in T
t_{end}	the last time period in T
U	set of edges $u = \{v, v'\}$ with $v, v' \in V$
V	set of nodes
W^{+}_{ewvt}	amount of energy carrier $e \in E$ that is stored in storage asset $w \in WA_e$ at node $v \in V$ in time period $t \in T$
W^{-}_{ewvt}	amount of energy carrier $e \in E$ that is discharged from storage asset $w \in WA_e$ at node $v \in V$ in time period $t \in T$
WA	set of all storage assets
WA_e	set of all storage assets storing energy carrier $e \in E$
γ_a	capacity of asset $a \in A$
δ	social discount rate
$\eta_{d\{v,v'\}}$	loss factor of distribution asset $d \in DA$ on edge $\{v, v'\}$
$\eta_{mee'}$	efficiency of converting energy carrier $e \in E$ to energy carrier $e' \in E$ with conversion asset $m \in MA_e$
η^{+}_{w}	charging efficiency of storage asset $w \in WA$
η^{-}_{w}	discharging efficiency of storage asset $w \in WA$
η^{sl}_{w}	standing losses of storage asset $w \in WA$
σ_{st}	external factor affecting supply asset $s \in SA$ in time period $t \in T$
ϕ_a	technological development rate of asset $a \in A$

B

MILP formulations

B.1. Original MILP formulation

The MILP formulation of the original MES model as described in Chapter 3:

$$
\begin{aligned}
\min \quad & \sum_{t\in T}\sum_{e\in E}\Bigg\{\sum_{d\in DA_e}\left[\sum_{u\in U} B_{dut}(c_{dut}+o_{dut}^{fix}) + \sum_{u\in DU} o_{dut}^{var}\cdot F_{edut}\right] \\
& + \sum_{v\in V}\Bigg[\sum_{s\in SA_e}(B_{svt}(c_{svt}+o_{svt}^{fix}) + o_{svt}^{var}\cdot S_{esvt}) \\
& + \sum_{m\in MA_e}(B_{mvt}(c_{mvt}+o_{mvt}^{fix}) + o_{mvt}^{var}\cdot M_{emvt}) \\
& + \sum_{w\in WA_e}(B_{wvt}(c_{wvt}+o_{wvt}^{fix}) + o_{wvt}^{+,var}\cdot W_{ewvt}^{+} + o_{wvt}^{-.var}\cdot W_{ewvt}^{-}))\Bigg]\Bigg\}
\end{aligned} \tag{3.18}
$$

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v',v)t}\cdot \eta_{d\{v,v'\}} + \sum_{s\in SA_e} S_{esvt}\cdot\sigma_{st} + \sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_{e'}} M_{e'mvt}\cdot\eta_{me'e} + \sum_{w\in WA_e} W_{ewvt}^{-}\cdot\eta_{w}^{-} \\
& \geq\ d_{evt} + \sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v,v')t} + \sum_{m\in MA_e} M_{emvt} + \sum_{w\in WA_e} W_{ewvt}^{+}, \qquad \forall e\in E, v\in V, t\in T
\end{aligned} \tag{3.3}
$$

$$F_{ed(v,v')t} \leq \gamma_d \sum_{i=t_0}^{t} B_{d\{v,v\}i}, \quad \forall e\in E, d\in DA_e, v, v'\in V, t\in T \tag{3.4}$$

$$S_{esvt} \leq \gamma_s \sum_{i=t_0}^{t} B_{svi}, \quad \forall e\in E, s\in SA_e, v\in V, t\in T \tag{3.5}$$

$$M_{emvt} \leq \gamma_m \sum_{i=t_0}^{t} B_{mvi}, \quad \forall e\in E, m\in MA_e, v\in V, t\in T \tag{3.6}$$

$$W_{ewvt}^{+}\cdot\eta_w^{+} \leq \gamma_w \sum_{i=t_0}^{t} B_{wvi} - W_{ewvt}^{tot}, \quad \forall e\in E, w\in WA_e, v\in V, t\in T \tag{3.7}$$

$$W_{ewvt}^{-} \leq W_{ewvt}^{tot}, \quad \forall e\in E, w\in WA_e, v\in V, t\in T \tag{3.8}$$

$$W_{ewvt_0}^{tot} = 0, \quad \forall e\in E, w\in WA_e, v\in V \tag{3.9}$$

$$
\begin{aligned}
W_{ewvt}^{tot} &= (W_{ewv(t-1)}^{tot} + W_{ewv(t-1)}^{+}\cdot\eta_w^{+} - W_{ewv(t-1)}^{-})\cdot(1-\eta_w^{sl}), \\
& \qquad \forall e\in E, w\in WA_e, v\in V, t\in T\setminus\{t_0\}
\end{aligned} \tag{3.10}
$$

$$B_{dut} \leq N,\ B_{dut}\in\mathbb{Z}^{+}, \quad \forall d\in DA, u\in U, t\in T \tag{3.11}$$

$$B_{avt} \leq N,\ B_{avt}\in\mathbb{Z}^{+}, \quad \forall a\in A\setminus DA, v\in V, t\in T \tag{3.12}$$

$$F_{edut}\in\mathbb{R}^{+}, \quad \forall e\in E, d\in DA_e, u\in DU, t\in T \tag{3.14}$$

$$S_{esvt}\in\mathbb{R}^{+}, \quad \forall e\in E, s\in SA_e, v\in V, t\in T \tag{3.15}$$

$$M_{emvt}\in\mathbb{R}^{+}, \quad \forall e\in E, m\in MA_e, v\in V, t\in T \tag{3.16}$$

$$W_{ewvt}^{+},\ W_{ewvt}^{-},\ W_{ewvt}^{tot}\in\mathbb{R}^{+}, \quad \forall e\in E, w\in WA_e, v\in V, t\in T \tag{3.17}$$

B.2. Column generation per time period

B.2.1. Original MILP formulation with colors

The original MILP formulation, where the terms containing multiple variables concerning different time periods are marked.

$$
\begin{aligned}
\min \quad & \sum_{t\in T}\sum_{e\in E}\Bigg\{\sum_{d\in DA_e}\Bigg[\sum_{u\in U} B_{dut}(c_{dut}+o_{dut}^{fix}) + \sum_{u\in DU} o_{dut}^{var}\cdot F_{edut}\Bigg] \\
& + \sum_{v\in V}\Bigg[\sum_{s\in SA_e} (B_{svt}(c_{svt}+o_{svt}^{fix}) + o_{svt}^{var}\cdot S_{esvt}) \\
& + \sum_{m\in MA_e} (B_{mvt}(c_{mvt}+o_{mvt}^{fix}) + o_{mvt}^{var}\cdot M_{emvt}) \\
& + \sum_{w\in WA_e} (B_{wvt}(c_{wvt}+o_{wvt}^{fix}) + o_{wvt}^{+,var}\cdot W_{ewvt}^{+} + o_{wvt}^{-.var}\cdot W_{ewvt}^{-}))\Bigg]\Bigg\}
\end{aligned}
\tag{3.18}
$$

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v',v)t}\cdot \eta_{d\{v,v'\}} + \sum_{s\in SA_e} S_{esvt}\cdot\sigma_{st} + \sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_{e'}} M_{e'mvt}\cdot\eta_{me'e} + \sum_{w\in WA_e} W_{ewvt}^{-}\cdot\eta_w^{-} \\
& \geq d_{evt} + \sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v,v')t} + \sum_{m\in MA_e} M_{emvt} + \sum_{w\in WA_e} W_{ewvt}^{+}, \qquad \forall e\in E, v\in V, t\in T
\end{aligned}
\tag{3.3}
$$

$$F_{ed(v,v')t} \leq \gamma_d \sum_{i=t_0}^{t} B_{d\{v,v'\}i}\,, \quad \forall e\in E, d\in DA_e, v,v'\in V, t\in T \tag{3.4}$$

$$S_{esvt} \leq \gamma_s \sum_{i=t_0}^{t} B_{svi}\,, \quad \forall e\in E, s\in SA_e, v\in V, t\in T \tag{3.5}$$

$$M_{emvt} \leq \gamma_m \sum_{i=t_0}^{t} B_{mvi}\,, \quad \forall e\in E, m\in MA_e, v\in V, t\in T \tag{3.6}$$

$$W_{ewvt}^{+}\cdot\eta_w^{+} \leq \gamma_w \sum_{i=t_0}^{t} B_{wvi} - W_{ewvt}^{tot}\,, \quad \forall e\in E, w\in WA_e, v\in V, t\in T \tag{3.7}$$

$$W_{ewvt}^{-} \leq W_{ewvt}^{tot}\,, \quad \forall e\in E, w\in WA_e, v\in V, t\in T \tag{3.8}$$

$$W_{ewvt_0}^{tot} = 0, \quad \forall e\in E, w\in WA_e, v\in V \tag{3.9}$$

$$
\begin{aligned}
& W_{ewvt}^{tot} = (W_{ewv(t-1)}^{tot} + W_{ewv(t-1)}^{+}\cdot\eta_w^{+} - W_{ewv(t-1)}^{-})\cdot(1-\eta_w^{sl})\,, \\
& \qquad\qquad\qquad \forall e\in E, w\in WA_e, v\in V, t\in T\setminus\{t_0\}
\end{aligned}
\tag{3.10}
$$

$$B_{dut} \leq N,\ B_{dut}\in\mathbb{Z}^{+}, \quad \forall d\in DA, u\in U, t\in T \tag{3.11}$$

$$B_{avt} \leq N,\ B_{avt}\in\mathbb{Z}^{+}, \quad \forall a\in A\setminus DA, v\in V, t\in T \tag{3.12}$$

$$F_{ed(v,v')t}\in\mathbb{R}^{+}, \quad \forall e\in E, d\in DA_e, u\in DU, t\in T \tag{3.14}$$

$$S_{esvt}\in\mathbb{R}^{+}, \quad \forall e\in E, s\in SA_e, v\in V, t\in T \tag{3.15}$$

$$M_{emvt}\in\mathbb{R}^{+}, \quad \forall e\in E, m\in MA_e, v\in V, t\in T \tag{3.16}$$

$$W_{ewvt}^{+},\ W_{ewvt}^{-},\ W_{ewvt}^{tot}\in\mathbb{R}^{+}, \quad \forall e\in E, w\in WA_e, v\in V, t\in T \tag{3.17}$$

B.2.2. Master problem

$$
\begin{aligned}
\min \quad & \sum_{t\in T}\sum_{k\in K_t}\sum_{e\in E}\Bigg\{\sum_{d\in DA_e}\Bigg[\sum_{u\in U}\lambda_t^k B_{dut}^k(c_{dut}+o_{dut}^{fix}) + \sum_{u\in DU} o_{dut}^{var}\cdot \lambda_t^k F_{edut}^k\Bigg] \\
& + \sum_{v\in V}\Bigg[\sum_{s\in SA_e}(\lambda_t^k B_{svt}^k(c_{svt}+o_{svt}^{fix}) + o_{svt}^{var}\cdot\lambda_t^k S_{esvt}^k) \\
& + \sum_{m\in MA_e}(\lambda_t^k B_{mvt}^k(c_{mvt}+o_{mvt}^{fix}) + o_{mvt}^{var}\cdot\lambda_t^k M_{emvt}^k) \\
& + \sum_{w\in WA_e}(\lambda_t^k B_{wvt}^k(c_{wvt}+o_{wvt}^{fix}) + o_{wvt}^{+,var}\cdot\lambda_t^k W_{ewvt}^{k,+} + o_{wvt}^{-,var}\cdot\lambda_t^k W_{ewvt}^{k,-})\Bigg]\Bigg\}
\end{aligned}
\tag{B.1}
$$

$$
\text{s.t.} \quad \sum_{k\in K_t}\lambda_t^k F_{ed(v,v')t}^k \le \gamma_d \sum_{i=t_0}^{t}\sum_{k\in K_i}\lambda_i^k B_{d\{v,v'\}i}^k, \quad \forall e\in E, d\in DA_e, v,v'\in V, t\in T\setminus\{t_0\} \qquad (\pi_{ed(v,v')t}^{DA})
\tag{B.2}
$$

$$
\sum_{k\in K_t}\lambda_t^k S_{esvt}^k \le \gamma_s \sum_{i=t_0}^{t}\sum_{k\in K_i}\lambda_i^k B_{svi}^k, \quad \forall e\in E, s\in SA_e, v\in V, t\in T\setminus\{t_0\} \qquad (\pi_{esvt}^{SA})
\tag{B.3}
$$

$$
\sum_{k\in K_t}\lambda_t^k M_{emvt}^k \le \gamma_m \sum_{i=t_0}^{t}\sum_{k\in K_i}\lambda_i^k B_{mvi}^k, \quad \forall e\in E, m\in MA_e, v\in V, t\in T\setminus\{t_0\} \qquad (\pi_{emvt}^{MA})
\tag{B.4}
$$

$$
\begin{aligned}
& \sum_{k\in K_t}\lambda_t^k W_{ewvt}^{k,+}\cdot\eta_w^+ \le \gamma_w \sum_{k\in K_t}\lambda_t^k B_{wvt}^k \\
& \quad + \sum_{i=t_0}^{t-1}\sum_{k\in K_i}\lambda_i^k\Bigg(B_{wvi}^k\cdot\gamma_w - (W_{ewvi}^{k,+}\cdot\eta_w^+ - W_{ewvi}^{k,-})\cdot(1-\eta_w^{sl})^{t-i}\Bigg), \qquad (\pi_{ewvt}^{WA+}) \\
& \qquad \forall e\in E, w\in WA_e, v\in V, t\in T\setminus\{t_0\}
\end{aligned}
\tag{B.5}
$$

$$
\begin{aligned}
& \sum_{k\in K_t}\lambda_t^k W_{ewvt}^{k,-} \le \sum_{i=t_0}^{t-1}\sum_{k\in K_i}\lambda_i^k(W_{ewvi}^{k,+}\cdot\eta_w^+ - W_{ewvi}^{k,-})\cdot(1-\eta_w^{sl})^{t-i}, \qquad (\pi_{ewvt}^{WA-}) \\
& \qquad \forall e\in E, w\in WA_e, v\in V, t\in T\setminus\{t_0\}
\end{aligned}
\tag{B.6}
$$

$$
\sum_{k\in K_t}\lambda_t^k = 1, \quad \forall t\in T \qquad (\mu_t)
\tag{B.7}
$$

$$
\lambda_t^k \in \{0,1\}, \quad \forall t\in T, k\in K_t
\tag{B.8}
$$

B.2.3. Pricing subproblems

For each $t \in T \setminus \{t_0\}$:

$$
\begin{aligned}
\min \quad & \sum_{e \in E} \Bigg\{ \sum_{d \in DA_e} \Bigg[\sum_{u \in U} B_{dut}(c_{dut} + o_{dut}^{fix}) + \sum_{u=(v,v') \in DU} \Bigg(o_{dut}^{var} \cdot F_{edut} - \pi_{edut}^{DA} \cdot F_{edut} + (B_{d\{v,v'\}t} \cdot \gamma_d) \cdot \sum_{i=t}^{t_{end}} \pi_{edui}^{DA} \Bigg) \\
& + \sum_{v \in V} \Bigg[\sum_{s \in SA_e} \Bigg(B_{svt}(c_{svt} + o_{svt}^{fix}) + o_{svt}^{var} \cdot S_{esvt} - \pi_{esvt}^{SA} \cdot S_{esvt} + (B_{svt} \cdot \gamma_s) \cdot \sum_{i=t}^{t_{end}} \pi_{esvi}^{SA} \Bigg) \\
& + \sum_{m \in MA_e} \Bigg(B_{mvt}(c_{mvt} + o_{mvt}^{fix}) + o_{mvt}^{var} \cdot M_{emvt} - \pi_{emvt}^{MA} \cdot M_{emvt} + (B_{mvt} \cdot \gamma_m) \cdot \sum_{i=t}^{t_{end}} \pi_{emvi}^{MA} \Bigg) \\
& + \sum_{w \in WA_e} \Bigg(B_{wvt}(c_{wvt} + o_{wvt}^{fix}) + o_{wvt}^{+,var} \cdot W_{ewvt}^{+} + o_{wvt}^{-,var} \cdot W_{ewvt}^{-} \\
& - \pi_{ewvt}^{WA+}\big(W_{ewvt}^{+} \cdot \eta_w^{+} - B_{wvt} \cdot \gamma_w\big) \\
& + \sum_{i=t+1}^{t_{end}} \pi_{ewvi}^{WA+}\big(B_{wvt} \cdot \gamma_w - (W_{ewvt}^{+} \cdot \eta_w^{+} - W_{ewvt}^{-}) \cdot (1 - \eta_w^{sl})^{i-t}\big) \\
& - \pi_{ewvt}^{WA-} \cdot W_{ewvt}^{-} + \sum_{i=t+1}^{t_{end}} \pi_{ewvi}^{WA-} \cdot (W_{ewvt}^{+} \cdot \eta_w^{+} - W_{ewvt}^{-}) \cdot (1 - \eta_w^{sl})^{i-t} \Bigg) \Bigg] \Bigg] \Bigg\} \\
& - \mu_t
\end{aligned}
\tag{B.9}
$$

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{d \in DA_e} \sum_{v' \in V} F_{ed(v',v)t} \cdot \eta_{d\{v,v'\}} + \sum_{s \in SA_e} S_{esvt} \cdot \sigma_{st} + \sum_{\substack{e' \in E \\ e' \neq e}} \sum_{m \in MA_{e'}} M_{e'mvt} \cdot \eta_{me'e} + \sum_{w \in WA_e} W_{ewvt}^{-} \cdot \eta_w^{-} \\
& \geq \; d_{evt} + \sum_{d \in DA_e} \sum_{v' \in V} F_{ed(v,v')t} + \sum_{m \in MA_e} M_{emvt} + \sum_{w \in WA_e} W_{ewvt}^{+}, \qquad \forall e \in E, v \in V
\end{aligned}
\tag{B.10}
$$

$$W_{ewvt}^{+} \leq (t - t_0) \cdot N \cdot \gamma_w, \quad \forall e \in E, w \in WA_e, v \in V \tag{B.11}$$

$$W_{ewvt}^{-} \leq (t - t_0 - 1) \cdot N \cdot \gamma_w, \quad \forall e \in E, w \in WA_e, v \in V \tag{B.12}$$

$$B_{dut} \leq N, \; B_{dut} \in \mathbb{Z}^{+}, \quad \forall d \in DA, u \in U \tag{B.13}$$

$$B_{avt} \leq N, \; B_{avt} \in \mathbb{Z}^{+}, \quad \forall a \in A \setminus DA, v \in V \tag{B.14}$$

$$F_{edut} \in \mathbb{R}^{+}, \quad \forall e \in E, d \in DA_e, u \in DU \tag{B.15}$$

$$S_{esvt} \in \mathbb{R}^{+}, \quad \forall e \in E, s \in SA_e, v \in V \tag{B.16}$$

$$M_{emvt} \in \mathbb{R}^{+}, \quad \forall e \in E, m \in MA_e, v \in V \tag{B.17}$$

$$W_{ewvt}^{+}, \, W_{ewvt}^{-} \in \mathbb{R}^{+}, \quad \forall e \in E, w \in WA_e, v \in V \tag{B.18}$$

Objective function of the pricing subproblem for $t = t_0$:

$$\begin{aligned}
\min \quad & \sum_{e\in E}\Bigg\{\sum_{d\in DA_e}\Bigg[\sum_{u\in U} B_{dut_0}(c_{dut_0}+o^{fix}_{dut_0}) + \sum_{u=(v,v')\in DU}\Bigg(o^{var}_{dut_0}\cdot F_{edut_0} + (B_{d\{v,v'\}t_0}\cdot\gamma_d)\cdot\sum_{i=t_0+1}^{t_{end}}\pi^{DA}_{edui}\Bigg) \\
& + \sum_{v\in V}\Bigg[\sum_{s\in SA_e}\Bigg(B_{svt_0}(c_{svt_0}+o^{fix}_{svt_0}) + o^{var}_{svt_0}\cdot S_{esvt_0} + (B_{svt_0}\cdot\gamma_s)\cdot\sum_{i=t_0+1}^{t_{end}}\pi^{SA}_{esvi}\Bigg) \\
& + \sum_{m\in MA_e}\Bigg(B_{mvt_0}(c_{mvt_0}+o^{fix}_{mvt_0}) + o^{var}_{mvt_0}\cdot M_{emvt_0} + (B_{mvt_0}\cdot\gamma_m)\cdot\sum_{i=t_0+1}^{t_{end}}\pi^{MA}_{emvi}\Bigg) \\
& + \sum_{w\in WA_e}\Bigg(B_{wvt_0}(c_{wvt_0}+o^{fix}_{wvt_0}) + o^{+,var}_{wvt_0}\cdot W^{+}_{ewvt_0} + o^{-,var}_{wvt_0}\cdot W^{-}_{ewvt_0} \\
& + \sum_{i=t_0+1}^{t_{end}}\pi^{WA+}_{ewvi}\Big(B_{wvt_0}\cdot\gamma_w - (W^{+}_{ewvt_0}\cdot\eta^{+}_{w} - W^{-}_{ewvt_0})\cdot(1-\eta^{sl}_{w})^{i-t_0}\Big) \\
& + \sum_{i=t_0+1}^{t_{end}}\pi^{WA-}_{ewvi}\cdot(W^{+}_{ewvt_0}\cdot\eta^{+}_{w} - W^{-}_{ewvt_0})\cdot(1-\eta^{sl}_{w})^{i-t_0}\Bigg)\Bigg]\Bigg\} \\
& - \mu_{t_0}
\end{aligned} \tag{B.19}$$

Constraints added for $t = t_0$:

$$F_{ed(v,v')t_0} \leq B_{d(\{v,v'\}t_0}\cdot\gamma_d, \quad \forall e\in E, d\in DA_e, v,v'\in V \tag{B.20}$$

$$S_{esvt_0} \leq B_{svt_0}\cdot\gamma_s, \quad \forall e\in E, s\in SA_e, v\in V \tag{B.21}$$

$$M_{emvt_0} \leq B_{mvt_0}\cdot\gamma_m, \quad \forall e\in E, m\in MA_e, v\in V \tag{B.22}$$

$$W^{+}_{ewvt_0}\cdot\eta^{+}_{w} \leq B_{wvt_0}\cdot\gamma_w, \quad \forall e\in E, w\in WA_e, v\in V \tag{B.23}$$

$$W^{-}_{ewvt_0} = 0, \quad \forall e\in E, w\in WA_e, v\in V \tag{B.24}$$

B.3. Column generation per energy carrier type

B.3.1. Original MILP formulation with colors

The original MILP formulation, where the terms containing multiple variables concerning different energy carriers are marked.

$$
\begin{aligned}
\min \quad & \sum_{t\in T}\sum_{e\in E}\Bigg\{\sum_{d\in DA_e}\Bigg[\sum_{u\in U} B_{dut}(c_{dut}+o^{fix}_{dut})+\sum_{u\in DU} o^{var}_{dut}\cdot F_{edut}\Bigg] \\
& +\sum_{v\in V}\Bigg[\sum_{s\in SA_e}(B_{svt}(c_{svt}+o^{fix}_{svt})+o^{var}_{svt}\cdot S_{esvt}) \\
& +\sum_{m\in MA_e}(B_{mvt}(c_{mvt}+o^{fix}_{mvt})+o^{var}_{mvt}\cdot M_{emvt}) \\
& +\sum_{w\in WA_e}(B_{wvt}(c_{wvt}+o^{fix}_{wvt})+o^{+,var}_{wvt}\cdot W^{+}_{ewvt}+o^{-.var}_{wvt}\cdot W^{-}_{ewvt}))\Bigg]\Bigg]\Bigg\}
\end{aligned}
\tag{3.18}
$$

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v',v)t}\cdot\eta_{d\{v,v'\}}+\sum_{s\in SA_e} S_{esvt}\cdot\sigma_{st}+\sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_{e'}} M_{e'mvt}\cdot\eta_{me'e}+\sum_{w\in WA_e} W^{-}_{ewvt}\cdot\eta^{-}_{w} \\
& \geq\ d_{evt}+\sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v,v')t}+\sum_{m\in MA_e} M_{emvt}+\sum_{w\in WA_e} W^{+}_{ewvt}, \qquad \forall e\in E, v\in V, t\in T
\end{aligned}
\tag{3.3}
$$

$$F_{ed(v,v')t}\leq\gamma_d\sum_{i=t_0}^{t} B_{d\{v,v'\}i}, \quad \forall e\in E, d\in DA_e, v,v'\in V, t\in T \tag{3.4}$$

$$S_{esvt}\leq\gamma_s\sum_{i=t_0}^{t} B_{svi}, \quad \forall e\in E, s\in SA_e, v\in V, t\in T \tag{3.5}$$

$$M_{emvt}\leq\gamma_m\sum_{i=t_0}^{t} B_{mvi}, \quad \forall e\in E, m\in MA_e, v\in V, t\in T \tag{3.6}$$

$$W^{+}_{ewvt}\cdot\eta^{+}_{w}\leq\gamma_w\sum_{i=t_0}^{t} B_{wvi}-W^{tot}_{ewvt}, \quad \forall e\in E, w\in WA_e, v\in V, t\in T \tag{3.7}$$

$$W^{-}_{ewvt}\leq W^{tot}_{ewvt}, \quad \forall e\in E, w\in WA_e, v\in V, t\in T \tag{3.8}$$

$$W^{tot}_{ewvt_0}=0, \quad \forall e\in E, w\in WA_e, v\in V \tag{3.9}$$

$$
\begin{aligned}
W^{tot}_{ewvt} &= (W^{tot}_{ewv(t-1)}+W^{+}_{ewv(t-1)}\cdot\eta^{+}_{w}-W^{-}_{ewv(t-1)})\cdot(1-\eta^{sl}_{w}), \\
& \qquad \forall e\in E, w\in WA_e, v\in V, t\in T\setminus\{t_0\}
\end{aligned}
\tag{3.10}
$$

$$B_{dut}\leq N,\ B_{dut}\in\mathbb{Z}^{+}, \quad \forall d\in DA, u\in U, t\in T \tag{3.11}$$

$$B_{avt}\leq N,\ B_{avt}\in\mathbb{Z}^{+}, \quad \forall a\in A\setminus DA, v\in V, t\in T \tag{3.12}$$

$$F_{edut}\in\mathbb{R}^{+}, \quad \forall e\in E, d\in DA_e, u\in DU, t\in T \tag{3.14}$$

$$S_{esvt}\in\mathbb{R}^{+}, \quad \forall e\in E, s\in SA_e, v\in V, t\in T \tag{3.15}$$

$$M_{emvt}\in\mathbb{R}^{+}, \quad \forall e\in E, m\in MA_e, v\in V, t\in T \tag{3.16}$$

$$W^{+}_{ewvt},\ W^{-}_{ewvt},\ W^{tot}_{ewvt}\in\mathbb{R}^{+}, \quad \forall e\in E, w\in WA_e, v\in V, t\in T \tag{3.17}$$

B.3.2. Master problem

$$\begin{aligned}
\min \quad & \sum_{t\in T}\sum_{e\in E}\sum_{k\in K_e}\Bigg\{\sum_{d\in DA_e}\left[\sum_{u\in U}\lambda_e^k B_{dut}^k(c_{dut}+o_{dut}^{fix}) + \sum_{u\in DU} o_{dut}^{var}\cdot\lambda_e^k F_{edut}^k\right] \\
& + \sum_{v\in V}\Bigg[\sum_{s\in SA_e}(\lambda_e^k B_{svt}^k(c_{svt}+o_{svt}^{fix}) + o_{svt}^{var}\cdot\lambda_e^k S_{esvt}^k) \\
& + \sum_{m\in MA_e}(\lambda_e^k B_{mvt}^k(c_{mvt}+o_{mvt}^{fix}) + o_{mvt}^{var}\cdot\lambda_e^k M_{emvt}^k) \\
& + \sum_{w\in WA_e}(\lambda_e^k B_{wvt}^k(c_{wvt}+o_{wvt}^{fix}) + o_{wvt}^{+,var}\cdot\lambda_e^k W_{ewvt}^{k,+} + o_{wvt}^{+,var}\cdot\lambda_e^k W_{ewvt}^{k,-})\Bigg]\Bigg\}
\end{aligned} \tag{B.25}$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_{d\in DA_e}\sum_{v'\in V}\sum_{k\in K_e}\lambda_e^k F_{ed(v',v)t}^k\cdot\eta_{d\{v,v'\}} + \sum_{s\in SA_e}\sum_{k\in K_e}\lambda_e^k S_{esvt}^k\cdot\sigma_{st} \\
& + \sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_{e'}}\sum_{k\in K_{e'}}\lambda_{e'}^k M_{e'mvt}^k\cdot\eta_{me'e} + \sum_{w\in WA_e}\sum_{k\in K_e}\lambda_e^k W_{ewvt}^{k,-}\cdot\eta_w^- \\
& \geq d_{evt} + \sum_{d\in DA_e}\sum_{v'\in V}\sum_{k\in K_e}\lambda_e^k F_{ed(v,v')t}^k + \sum_{m\in MA_e}\sum_{k\in K_e}\lambda_e^k M_{emvt}^k + \sum_{w\in WA_e}\sum_{k\in K_e}\lambda_e^k W_{ewvt}^{k,+}, \\
& \qquad \forall e\in E, v\in V, t\in T \qquad (\pi_{evt})
\end{aligned} \tag{B.26}$$

$$\sum_{k\in K_e}\lambda_e^k = 1, \quad \forall e\in E \qquad (\mu_e) \tag{B.27}$$

$$\lambda_e^k\in\{0,1\}, \quad \forall e\in E, k\in K_e \tag{B.28}$$

B.3.3. Pricing subproblems

For each $e \in E$:

$$
\begin{aligned}
\min \quad & \sum_{t\in T}\Bigg\{\sum_{d\in DA_e}\Bigg(\sum_{u\in U} B_{dut}(c_{dut}+o^{fix}_{dut}) + \sum_{u\in DU} o^{var}_{dut}\cdot F_{edut}\Bigg) \\
& + \sum_{v\in V}\Bigg[\sum_{s\in SA_e}\Bigg(B_{svt}(c_{svt}+o^{fix}_{svt}) + o^{var}_{svt}\cdot S_{esvt}\Bigg) \\
& + \sum_{m\in MA_e}\Bigg(B_{mvt}(c_{mvt}+o^{fix}_{mvt}) + o^{var}_{mvt}\cdot M_{emvt}\Bigg) \\
& + \sum_{w\in WA_e}\Bigg(B_{wvt}(c_{wvt}+o^{fix}_{wvt}) + o^{+,var}_{wvt}\cdot W^{+}_{ewvt} + o^{-,var}_{wvt}\cdot W^{-}_{ewvt}\Bigg) \\
& - \pi_{evt}\Bigg(\sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v',v)t}\cdot \eta_{d\{v,v'\}} + \sum_{s\in SA_e} S_{esvt}\cdot\sigma_{st} + \sum_{w\in WA_e} W^{-}_{ewvt}\cdot\eta^{-}_{w} \\
& - \sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v,v')t} - \sum_{m\in MA_e} M_{emvt} - \sum_{w\in WA_e} W^{+}_{ewvt}\Bigg) \\
& - \sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_e} \pi_{e'vt}\cdot M_{emvt}\cdot\eta_{mee'}\Bigg]\Bigg\} \\
& - \mu_e
\end{aligned}
\tag{B.29}
$$

s.t.

$$F_{ed(v,v')t} \leq \gamma_d \sum_{i=t_0}^{t} B_{d\{v,v'\}i}, \quad \forall d\in DA_e, v, v'\in V, t\in T \tag{B.30}$$

$$S_{esvt} \leq \gamma_s \sum_{i=t_0}^{t} B_{svi}, \quad \forall s\in SA_e, v\in V, t\in T \tag{B.31}$$

$$M_{emvt} \leq \gamma_m \sum_{i=t_0}^{t} B_{mvi}, \quad \forall m\in MA_e, v\in V, t\in T \tag{B.32}$$

$$W^{+}_{ewvt}\cdot\eta^{+}_{w} \leq \gamma_w \sum_{i=t_0}^{t} B_{wvi} - W^{tot}_{ewvt}, \quad \forall w\in WA_e, v\in V, t\in T \tag{B.33}$$

$$W^{-}_{ewvt} \leq W^{tot}_{ewvt}, \quad \forall w\in WA_e, v\in V, t\in T \tag{B.34}$$

$$W^{tot}_{ewvt_0} = 0, \quad \forall w\in WA_e, v\in V \tag{B.35}$$

$$W^{tot}_{ewvt} = (W^{tot}_{ewv(t-1)} + W^{+}_{ewv(t-1)}\cdot\eta^{+}_{w} - W^{-}_{ewv(t-1)})\cdot(1-\eta^{sl}_{w}), \quad \forall w\in WA_e, v\in V, t\in T\setminus\{t_0\} \tag{B.36}$$

$$B_{dut}\leq N,\ B_{dut}\in\mathbb{Z}^{+}, \quad \forall d\in DA_e, u\in U, t\in T \tag{B.37}$$

$$B_{svt}\leq N,\ B_{svt}\in\mathbb{Z}^{+}, \quad \forall s\in SA_e, v\in V, t\in T \tag{B.38}$$

$$B_{mvt}\leq N,\ B_{mvt}\in\mathbb{Z}^{+}, \quad \forall m\in MA_e, v\in V, t\in T \tag{B.39}$$

$$B_{wvt}\leq N,\ B_{wvt}\in\mathbb{Z}^{+}, \quad \forall w\in WA_e, v\in V, t\in T \tag{B.40}$$

$$F_{edut}\in\mathbb{R}^{+}, \quad \forall d\in DA_e, u\in DU, t\in T \tag{B.41}$$

$$S_{esvt}\in\mathbb{R}^{+}, \quad \forall s\in SA_e, v\in V, t\in T \tag{B.42}$$

$$M_{emvt}\in\mathbb{R}^{+}, \quad \forall m\in MA_e, v\in V, t\in T \tag{B.43}$$

$$W^{+}_{ewvt},\ W^{-}_{ewvt},\ W^{tot}_{ewvt}\in\mathbb{R}^{+}, \quad \forall w\in WA_e, v\in V, t\in T \tag{B.44}$$

B.4. Column generation per asset type

B.4.1. Original MILP formulation with colors

The original MILP formulation, where the terms containing variables concerning distribution (assets) are marked green, variables concerning supply (assets) are marked yellow, variables concerning conversion (assets) are marked red, and variables concerning storage (assets) are marked blue.

$$\min \sum_{t\in T}\sum_{e\in E}\Bigg\{\sum_{d\in DA_e}\Big[\sum_{u\in U} B_{dut}(c_{dut}+o_{dut}^{fix}) + \sum_{u\in DU} o_{dut}^{var}\cdot F_{edut}\Big]$$

$$+\sum_{v\in V}\Big[\sum_{s\in SA_e}(B_{svt}(c_{svt}+o_{svt}^{fix}) + o_{svt}^{var}\cdot S_{esvt}) + \sum_{m\in MA_e}(B_{mvt}(c_{mvt}+o_{mvt}^{fix}) + o_{mvt}^{var}\cdot M_{emvt})$$

$$+\sum_{w\in WA_e}(B_{wvt}(c_{wvt}+o_{wvt}^{fix}) + o_{wvt}^{+,var}\cdot W_{ewvt}^{+} + o_{wvt}^{-.var}\cdot W_{ewvt}^{-}))\Big]\Bigg\}$$

s.t. $$\sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v',v)t}\cdot \eta_{d\{v,v'\}} + \sum_{s\in SA_e} S_{esvt}\cdot\sigma_{st} + \sum_{\substack{e'\in E:\\ e'\neq e}}\sum_{m\in MA_{e'}} M_{e'mvt}\cdot\eta_{me'e} + \sum_{w\in WA_e} W_{ewvt}^{-}\cdot\eta_{w}^{-}$$

$$\geq d_{evt} + \sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v,v')t} + \sum_{m\in MA_e} M_{emvt} + \sum_{w\in WA_e} W_{ewvt}^{+}, \qquad \forall e\in E, v\in V, t\in T$$

$$F_{ed(v,v')t} \leq \gamma_d \sum_{i=t_0}^{t} B_{d\{v,v'\}i}, \qquad \forall e\in E, d\in DA_e, v, v'\in V, t\in T$$

$$S_{esvt} \leq \gamma_s \sum_{i=t_0}^{t} B_{svi}, \qquad \forall e\in E, s\in SA_e, v\in V, t\in T$$

$$M_{emvt} \leq \gamma_m \sum_{i=t_0}^{t} B_{mvi}, \qquad \forall e\in E, m\in MA_e, v\in V, t\in T$$

$$W_{ewvt}^{+}\cdot\eta_w^{+} \leq \gamma_w \sum_{i=t_0}^{t} B_{wvi} - W_{ewvt}^{tot}, \qquad \forall e\in E, w\in WA_e, v\in V, t\in T$$

$$W_{ewvt}^{-} \leq W_{ewvt}^{tot}, \qquad \forall e\in E, w\in WA_e, v\in V, t\in T$$

$$W_{ewvt_0}^{tot} = 0, \qquad \forall e\in E, w\in WA_e, v\in V$$

$$W_{ewvt}^{tot} = (W_{ewv(t-1)}^{tot} + W_{ewv(t-1)}^{+}\cdot\eta_w^{+} - W_{ewv(t-1)}^{-})\cdot(1-\eta_w^{sl}),$$

$$\forall e\in E, w\in WA_e, v\in V, t\in T\setminus\{t_0\}$$

$$B_{dut}\leq N,\ B_{dut}\in\mathbb{Z}^{+}, \qquad \forall d\in DA, u\in U, t\in T$$

$$B_{svt}\leq N,\ B_{svt}\in\mathbb{Z}^{+}, \qquad \forall s\in SA, v\in V, t\in T$$

$$B_{mvt}\leq N,\ B_{mvt}\in\mathbb{Z}^{+}, \qquad \forall m\in MA, v\in V, t\in T$$

$$B_{wvt}\leq N,\ B_{wvt}\in\mathbb{Z}^{+}, \qquad \forall w\in WA, v\in V, t\in T$$

$$F_{edut}\in\mathbb{R}^{+}, \qquad \forall e\in E, d\in DA_e, u\in DU, t\in T$$

$$S_{esvt}\in\mathbb{R}^{+}, \qquad \forall e\in E, s\in SA_e, v\in V, t\in T$$

$$M_{emvt}\in\mathbb{R}^{+}, \qquad \forall e\in E, m\in MA_e, v\in V, t\in T$$

$$W_{ewvt}^{+},\ W_{ewvt}^{-},\ W_{ewvt}^{tot}\in\mathbb{R}^{+}, \qquad \forall e\in E, w\in WA_e, v\in V, t\in T$$

B.4.2. Master problem

$$
\begin{aligned}
\min \quad & \sum_{t\in T}\sum_{e\in E}\Bigg\{ \sum_{d\in DA_e}\sum_{k\in K_d}\left[\lambda_d^k B_{dut}^k (c_{dut}+o_{dut}^{fix}) + \sum_{u\in DU} o_{dut}^{var}\cdot\lambda_d^k F_{edut}^k\right] \\
& \quad + \sum_{v\in V}\Bigg[\sum_{s\in SA_e}\sum_{k\in K_s}(\lambda_s^k B_{svt}^k(c_{svt}+o_{svt}^{fix}) + o_{svt}^{var}\cdot\lambda_s^k S_{esvt}^k) \\
& \quad + \sum_{m\in MA_e}\sum_{k\in K_m}(\lambda_m^k B_{mvt}^k(c_{mvt}+o_{mvt}^{fix}) + o_{mvt}^{var}\cdot\lambda_m^k M_{emvt}^k) \\
& \quad + \sum_{w\in WA_e}\sum_{k\in K_w}(\lambda_w^k B_{wvt}^k(c_{wvt}+o_{wvt}^{fix}) + o_{wvt}^{+,var}\cdot\lambda_w^k W_{ewvt}^{k,+} + o_{wvt}^{-,var}\cdot\lambda_w^k W_{ewvt}^{k,-})\Bigg]\Bigg\}
\end{aligned}
\tag{B.45}
$$

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{d\in DA_e}\sum_{v'\in V}\sum_{k\in K_d}\lambda_d^k F_{ed(v',v)t}^k\cdot\eta_{d\{v,v'\}} + \sum_{s\in SA_e}\sum_{k\in K_s}\lambda_s^k S_{esvt}^k\cdot\sigma_{st} \\
& \quad + \sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_{e'}}\sum_{k\in K_m}\lambda_m^k M_{e'mvt}^k\cdot\eta_{me'e} + \sum_{w\in WA_e}\sum_{k\in K_w}\lambda_w^k W_{ewvt}^{k,-}\cdot\eta_w^- \\
& \quad \geq d_{evt} + \sum_{d\in DA_e}\sum_{v'\in V}\sum_{k\in K_d}\lambda_d^k F_{ed(v,v')t}^k + \sum_{m\in MA_e}\sum_{k\in K_m}\lambda_m^k M_{emvt}^k + \sum_{w\in WA_e}\sum_{k\in K_w}\lambda_w^k W_{ewvt}^{k,+}, \\
& \qquad\qquad\qquad\qquad \forall e\in E, v\in V, t\in T \qquad (\pi_{evt})
\end{aligned}
\tag{B.46}
$$

$$
\sum_{k\in K_a}\lambda_a^k = 1, \quad \forall a\in A \qquad (\mu_a)
\tag{B.47}
$$

$$
\lambda_a^k \in \{0,1\}, \quad \forall a\in A, k\in K_a
\tag{B.48}
$$

B.4.3. Pricing subproblem for distribution asset types

For each $e \in E,\ d \in DA_e$:

$$\begin{aligned} \min \quad & \sum_{t \in T} \Bigg[\sum_{u \in U} B_{dut}(c_{dut} + o_{dut}^{fix}) + \sum_{u \in DU} o_{dut}^{var} \cdot F_{edut} \\ & - \sum_{v \in V} \pi_{evt} \Bigg(\sum_{v' \in V} F_{ed(v',v)t} \cdot \eta_{d\{v,v'\}} - \sum_{v' \in V} F_{ed(v,v')t} \Bigg) \Bigg] - \mu_d \end{aligned} \tag{B.49}$$

$$\text{s.t.} \quad F_{ed(v,v')t} \le \gamma_d \sum_{i=t_0}^{t} B_{d\{v,v'\}i}, \quad \forall v, v' \in V, t \in T \tag{B.50}$$

$$B_{dut} \le N,\ B_{dut} \in \mathbb{Z}^+, \quad \forall u \in U, t \in T \tag{B.51}$$

$$F_{edut} \in \mathbb{R}^+, \quad \forall u \in DU, t \in T \tag{B.52}$$

B.4.4. Pricing subproblem for supply asset types

For each $e \in E, s \in SA_e$:

$$\min \quad \sum_{t \in T} \sum_{v \in V} \Big[B_{svt}(c_{svt} + o_{svt}^{fix}) + o_{svt}^{var} \cdot S_{esvt} - \pi_{evt} \cdot S_{esvt} \cdot \sigma_{st} \Big] - \mu_s \tag{B.53}$$

$$\text{s.t.} \quad S_{esvt} \le \gamma_s \sum_{i=t_0}^{t} B_{svi}, \quad \forall v \in V, t \in T \tag{B.54}$$

$$B_{svt} \le N,\ B_{svt} \in \mathbb{Z}^+, \quad \forall v \in V, t \in T \tag{B.55}$$

$$S_{esvt} \in \mathbb{R}^+, \quad \forall v \in V, t \in T \tag{B.56}$$

B.4.5. Pricing subproblem for conversion asset types

For each $e \in E, m \in MA_e$:

$$\begin{aligned} \min \quad & \sum_{t \in T} \sum_{v \in V} \Bigg[B_{mvt}(c_{mvt} + o_{mvt}^{fix}) + o_{mvt}^{var} \cdot M_{emvt} \\ & + \pi_{evt} \cdot M_{emvt} - \sum_{\substack{e' \in E: \\ e' \neq e}} \pi_{e'vt} \cdot M_{emvt} \cdot \eta_{mee'} \Bigg] - \mu_m \end{aligned} \tag{B.57}$$

$$\text{s.t.} \quad M_{emvt} \le \gamma_m \sum_{i=t_0}^{t} B_{mvi}, \quad \forall v \in V, t \in T \tag{B.58}$$

$$B_{mvt} \le N,\ B_{mvt} \in \mathbb{Z}^+, \quad \forall v \in V, t \in T \tag{B.59}$$

$$M_{emvt} \in \mathbb{R}^+, \quad \forall v \in V, t \in T \tag{B.60}$$

B.4.6. Pricing subproblem for storage asset types

For each $e \in E, w \in WA_e$:

$$\begin{aligned} \min \quad & \sum_{t \in T} \sum_{v \in V} \Bigg[B_{wvt}(c_{wvt} + o_{wvt}^{fix}) + o_{wvt}^{+,var} \cdot W_{ewvt}^{+} + o_{wvt}^{-,var} \cdot W_{ewvt}^{-} \\ & - \pi_{evt} \Big(W_{ewvt}^{-} \cdot \eta_w^{-} - W_{ewvt}^{+} \Big) - \mu_w \end{aligned} \tag{B.61}$$

$$\text{s.t.} \quad W_{ewvt}^{+} \cdot \eta_w^{+} \leq \gamma_w \sum_{i=t_0}^{t} B_{wvi} - W_{ewvt}^{tot}, \quad \forall v \in V, t \in T \tag{B.62}$$

$$W_{ewvt}^{-} \leq W_{ewvt}^{tot}, \quad \forall v \in V, t \in T \tag{B.63}$$

$$W_{ewvt_0}^{tot} = 0, \quad \forall v \in V \tag{B.64}$$

$$W_{ewvt}^{tot} = (W_{ewv(t-1)}^{tot} + W_{ewv(t-1)}^{+} \cdot \eta_w^{+} - W_{ewv(t-1)}^{-}) \cdot (1 - \eta_w^{sl}), \quad \forall v \in V, t \in T \setminus \{t_0\} \tag{B.65}$$

$$B_{wvt} \leq N, \; B_{wvt} \in \mathbb{Z}^{+}, \quad \forall v \in V, t \in T \tag{B.66}$$

$$W_{ewvt}^{+}, \; W_{ewvt}^{-}, \; W_{ewvt}^{tot} \in \mathbb{R}^{+}, \quad \forall v \in V, t \in T \tag{B.67}$$

B.5. Column generation per node

B.5.1. Original MILP formulation with colors

The original MILP formulation, where the terms containing multiple variables concerning different directed edges are marked.

$$
\begin{aligned}
\min \quad & \sum_{t\in T}\sum_{e\in E}\Bigg\{\sum_{d\in DA_e}\Bigg[B_{dut}(c_{dut}+o^{fix}_{dut})+\sum_{u\in DU} o^{var}_{dut}\cdot F_{edut}\Bigg] \\
& +\sum_{v\in V}\Bigg[\sum_{s\in SA_e}(B_{svt}(c_{svt}+o^{fix}_{svt})+o^{var}_{svt}\cdot S_{esvt}) \\
& +\sum_{m\in MA_e}(B_{mvt}(c_{mvt}+o^{fix}_{mvt})+o^{var}_{mvt}\cdot M_{emvt}) \\
& +\sum_{w\in WA_e}(B_{wvt}(c_{wvt}+o^{fix}_{wvt})+o^{+,var}_{wvt}\cdot W^{+}_{ewvt}+o^{-.var}_{wvt}\cdot W^{-}_{ewvt}))\Bigg]\Bigg\}
\end{aligned}
\tag{3.18}
$$

$$
\begin{aligned}
\text{s.t.} \quad & \sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v',v)t}\cdot\eta_{d\{v,v'\}}+\sum_{s\in SA_e} S_{esvt}\cdot\sigma_{st}+\sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_{e'}} M_{e'mvt}\cdot\eta_{me'e}+\sum_{w\in WA_e} W^{-}_{ewvt}\cdot\eta^{-}_{w}, \\
& \geq d_{evt}+\sum_{d\in DA_e}\sum_{v'\in V} F_{ed(v,v')t}+\sum_{m\in MA_e} M_{emvt}+\sum_{w\in WA_e} W^{+}_{ewvt} \\
& \qquad \forall e\in E, v\in V, t\in T
\end{aligned}
\tag{3.3}
$$

$$F_{ed(v,v')t}\leq\gamma_d\sum_{i=t_0}^{t} B_{d\{v,v'\}i},\quad \forall e\in E, d\in DA_e, v,v'\in V, t\in T \tag{3.4}$$

$$S_{esvt}\leq\gamma_s\sum_{i=t_0}^{t} B_{svi},\quad \forall e\in E, s\in SA_e, v\in V, t\in T \tag{3.5}$$

$$M_{emvt}\leq\gamma_m\sum_{i=t_0}^{t} B_{mvi},\quad \forall e\in E, m\in MA_e, v\in V, t\in T \tag{3.6}$$

$$W^{+}_{ewvt}\cdot\eta^{+}_{w}\leq\gamma_w\sum_{i=t_0}^{t} B_{wvi}-W^{tot}_{ewvt},\quad \forall e\in E, w\in WA_e, v\in V, t\in T \tag{3.7}$$

$$W^{-}_{ewvt}\leq W^{tot}_{ewvt},\quad \forall e\in E, w\in WA_e, v\in V, t\in T \tag{3.8}$$

$$W^{tot}_{ewvt_0}=0,\quad \forall e\in E, w\in WA_e, v\in V \tag{3.9}$$

$$
\begin{aligned}
& W^{tot}_{ewvt}=(W^{tot}_{ewv(t-1)}+W^{+}_{ewv(t-1)}\cdot\eta^{+}_{w}-W^{-}_{ewv(t-1)})\cdot(1-\eta^{sl}_{w}), \\
& \qquad \forall e\in E, w\in WA_e, v\in V, t\in T\setminus\{t_0\}
\end{aligned}
\tag{3.10}
$$

$$B_{dut}\leq N,\ B_{dut}\in\mathbb{Z}^{+},\quad \forall d\in DA, u\in U, t\in T \tag{3.11}$$

$$B_{avt}\leq N,\ B_{avt}\in\mathbb{Z}^{+},\quad \forall a\in A\setminus DA, v\in V, t\in T \tag{3.12}$$

$$F_{edut}\in\mathbb{R}^{+},\quad \forall e\in E, d\in DA_e, u\in DU, t\in T \tag{3.14}$$

$$S_{esvt}\in\mathbb{R}^{+},\quad \forall e\in E, s\in SA_e, v\in V, t\in T \tag{3.15}$$

$$M_{emvt}\in\mathbb{R}^{+},\quad \forall e\in E, m\in MA_e, v\in V, t\in T \tag{3.16}$$

$$W^{+}_{ewvt},\ W^{-}_{ewvt},\ W^{tot}_{ewvt}\in\mathbb{R}^{+},\quad \forall e\in E, w\in WA_e, v\in V, t\in T \tag{3.17}$$

B.5.2. Master problem

The master problem for column generation per node, where the parts that are only included for the second implementation option are marked blue, and the parts only included for the third implementation option are yellow.

$$\begin{aligned}\min \quad & \sum_{t\in T}\sum_{e\in E}\Bigg\{\sum_{v\in V}\sum_{k\in K_v}\Bigg[\sum_{v'\in V}\sum_{d\in DA_e}\Big(\frac{1}{2}\lambda_v^k B_{d(v,v')t}^k(c_{d\{v,v'\}t}+o_{d\{v,v'\}t}^{fix})+o_{d\{v,v'\}t}^{var}\cdot\lambda_v^k F_{ed(v,v')t}^k\Big)\\ & \quad +\sum_{s\in SA_e}(\lambda_v^k B_{svt}^k(c_{svt}+o_{svt}^{fix})+o_{svt}^{var}\cdot\lambda_v^k S_{esvt}^k)\\ & \quad +\sum_{m\in MA_e}(\lambda_v^k B_{mvt}^k(c_{mvt}+o_{mvt}^{fix})+o_{mvt}^{var}\cdot\lambda_v^k M_{emvt}^k)\\ & \quad +\sum_{w\in WA_e}(\lambda_v^k B_{wvt}^k(c_{wvt}+o_{wvt}^{fix})+o_{wvt}^{+,var}\cdot\lambda_v^k W_{ewvt}^{k,+}+o_{wvt}^{-,var}\cdot\lambda_v^k W_{ewvt}^{k,-})\Bigg]\Bigg\}\end{aligned} \tag{B.68}$$

$$\begin{aligned}\text{s.t.}\quad & \sum_{d\in DA_e}\sum_{\substack{v',u:\\ u=(v',v)\in U_v^{in}}}\sum_{k\in K_{v'}}\lambda_{v'}^k F_{ed(v',v)t}^k\cdot\eta_{d\{v,v'\}}+\sum_{s\in SA_e}\sum_{k\in K_v}\lambda_v^k S_{esvt}^k\cdot\sigma_{st}\\ & +\sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_{e'}}\sum_{k\in K_v}\lambda_v^k M_{e'mvt}^k\cdot\eta_{me'e}+\sum_{w\in WA_e}\sum_{k\in K_v}\lambda_v^k W_{ewvt}^{k,-}\cdot\eta_w^-\\ & \geq d_{evt}+\sum_{d\in DA_e}\sum_{v'\in V}\sum_{k\in K_v}\lambda_v^k F_{ed(v,v')t}^k+\sum_{m\in MA_e}\sum_{k\in K_v}\lambda_v^k M_{emvt}^k+\sum_{w\in WA_e}\sum_{k\in K_v}\lambda_v^k W_{ewvt}^{k,+},\\ & \qquad \forall e\in E, v\in V, t\in T \qquad (\pi_{evt})\end{aligned} \tag{B.69}$$

$$\sum_{k\in K_v}\lambda_v^k B_{d(v,v')t}=\sum_{k\in K_{v'}}\lambda_{v'}^k B_{d(v',v)t}, \quad \forall d\in DA,(v,v')\in DU, t\in T \qquad (\rho_{d(v,v')t}) \tag{B.70}$$

$$\begin{aligned}& \sum_{k\in K_v}\lambda_v^k F_{ed(v,v')t}^k\leq\gamma_d\sum_{i=t_0}^{t}\Bigg(\sum_{k\in K_v}\lambda_v^k B_{d(v,v')i}^k+\sum_{k\in K_{v'}}\lambda_{v'}^k B_{d(v',v)i}^k\Bigg),\\ & \qquad \forall e\in E, d\in DA_e,(v,v')\in DU, t\in T \qquad (\tau_{ed(v,v')t})\end{aligned} \tag{B.71}$$

$$\sum_{k\in K_v}\lambda_v^k B_{d(v,v')t}^k+\sum_{k\in K_{v'}}\lambda_{v'}^k B_{d(v',v)t}^k\leq N, \quad \forall d\in DA,\{v,v'\}\in U, t\in T \qquad (\xi_{d\{v,v'\}t}) \tag{B.72}$$

$$\sum_{k\in K_v}\lambda_v^k=1, \quad \forall v\in V \qquad (\mu_v) \tag{B.73}$$

$$\lambda_v^k\in\{0,1\}, \quad \forall v\in V, k\in K_v \tag{B.74}$$

B.5.3. Pricing subproblems

The pricing subproblem for each $v \in V$ for column generation per node, where the parts that are only included for the second implementation option are marked blue, and the parts only included for the third implementation option are yellow:

$$
\begin{aligned}
\min \quad & \sum_{t\in T}\sum_{e\in E}\biggl\{\sum_{v'\in V}\sum_{d\in DA_e}\biggl(\frac{1}{2}B_{dut}(c_{dut}+o_{dut}^{fix})+o_{dut}^{var}\cdot F_{edut}\biggr) \\
& +\sum_{s\in SA_e}\biggl(B_{svt}(c_{svt}+o_{svt}^{fix})+o_{svt}^{var}\cdot S_{esvt}\biggr) \\
& +\sum_{m\in MA_e}\biggl(B_{mvt}(c_{mvt}+o_{mvt}^{fix})+o_{mvt}^{var}\cdot M_{emvt}\biggr) \\
& +\sum_{w\in WA_e}\biggl(B_{wvt}(c_{wvt}+o_{wvt}^{fix})+o_{wvt}^{+,var}\cdot W_{ewvt}^{+}+o_{wvt}^{-,var}\cdot W_{ewvt}^{-}\biggr) \\
& -\pi_{evt}\biggl(\sum_{s\in SA_e}S_{esvt}\cdot\sigma_{st}+\sum_{\substack{e'\in E\\ e'\neq e}}\sum_{m\in MA_{e'}}M_{e'mvt}\cdot\eta_{me'e}+\sum_{w\in WA_e}W_{ewvt}^{-}\cdot\eta_{w}^{-} \\
& -\sum_{d\in DA_e}\sum_{v'\in V}F_{ed(v,v')t}-\sum_{m\in MA_e}M_{emvt}-\sum_{w\in WA_e}W_{ewvt}^{+}\biggr) \\
& -\sum_{d\in DA}\sum_{\substack{v',u:\\ u=(v',v)\in U_v^{in}}}\pi_{ev't}\cdot F_{ed(v',v)t}\cdot\eta_{d\{v,v'\}}\biggr\} \\
& +\sum_{d\in DA}\sum_{v'\in V}\sum_{t\in T}\biggl(-\rho_{d(v,v')t}\cdot B_{d(v,v')t}+\rho_{d(v',v)t}\cdot B_{d(v,v')t}\biggr) \\
& +\sum_{d\in DA_e}\sum_{v'\in V}\sum_{t\in T}\biggl\{\sum_{e\in E}\biggl[\tau_{ed(v,v')t}\biggl(-F_{ed(v,v')t}+\gamma_d\sum_{i=t_0}^{t}B_{d(v,v')i}\biggr)+\tau_{ed(v',v)t}\cdot\gamma_d\sum_{i=t_0}^{t}B_{d(v,v')i}\biggr] \\
& -\xi_{d\{v,v'\}}\cdot B_{d(v,v')t}\biggr\} \\
& -\mu_v
\end{aligned}
\tag{B.75}
$$

The constraints included in the pricing subproblem for node $v \in V$ for all the implementation options:

$$\text{s.t.} \quad S_{esvt} \leq \gamma_s \sum_{i=t_0}^{t} B_{svi}, \quad \forall e \in E, s \in SA_e, t \in T \tag{B.76}$$

$$M_{emvt} \leq \gamma_m \sum_{i=t_0}^{t} B_{mvi}, \quad \forall e \in E, m \in MA_e, t \in T \tag{B.77}$$

$$W_{ewvt}^{+} \cdot \eta_w^{+} \leq \gamma_w \sum_{i=t_0}^{t} B_{wvi} - W_{ewvt}^{tot}, \quad \forall e \in E, w \in WA_e, t \in T \tag{B.78}$$

$$W_{ewvt}^{-} \leq W_{ewvt}^{tot}, \quad \forall e \in E, w \in WA_e, t \in T \tag{B.79}$$

$$W_{ewvt_0}^{tot} = 0, \quad \forall e \in E, w \in WA_e \tag{B.80}$$

$$W_{ewvt}^{tot} = (W_{ewv(t-1)}^{tot} + W_{ewv(t-1)}^{+} \cdot \eta_w^{+} - W_{ewv(t-1)}^{-}) \cdot (1 - \eta_w^{sl}), \quad \forall e \in E, w \in WA_e, t \in T \setminus \{t_0\} \tag{B.81}$$

$$B_{d(v,v')t} \leq N,\ B_{d(v,v')t} \in \mathbb{Z}^{+}, \quad \forall e \in E, d \in DA_e, v' \in V, t \in T \tag{B.82}$$

$$B_{svt} \leq N,\ B_{svt} \in \mathbb{Z}^{+}, \quad \forall e \in E, s \in SA_e, t \in T \tag{B.83}$$

$$B_{mvt} \leq N,\ B_{mvt} \in \mathbb{Z}^{+}, \quad \forall e \in E, m \in MA_e, t \in T \tag{B.84}$$

$$B_{wvt} \leq N,\ B_{wvt} \in \mathbb{Z}^{+}, \quad \forall e \in E, w \in WA_e, t \in T \tag{B.85}$$

$$F_{ed(v,v')t} \in \mathbb{R}^{+}, \quad \forall e \in E, d \in DA_e, v' \in V, t \in T \tag{B.86}$$

$$S_{esvt} \in \mathbb{R}^{+}, \quad \forall e \in E, s \in SA_e, t \in T \tag{B.87}$$

$$M_{emvt} \in \mathbb{R}^{+}, \quad \forall e \in E, m \in MA_e, t \in T \tag{B.88}$$

$$W_{ewvt}^{+},\ W_{ewvt}^{-},\ W_{ewvt}^{tot} \in \mathbb{R}^{+}, \quad \forall e \in E, w \in WA_e, t \in T \tag{B.89}$$

Extra included constraints for the first and second implementation option of column generation per node:

$$F_{ed(v,v')t} \leq \gamma_d \sum_{i=t_0}^{t} B_{d(v,v')i}, \quad \forall e \in E, d \in DA_e, v' \in V, t \in T$$

C
Data details

Tables on the costs, technological development rates, capacities and efficiencies/loss factors of the assets.

Table C.1: Initial costs, technological development rates, capacities and loss factors of the distribution assets.

asset type	cost c_{dut_0} [Eur/m]	tech. dev. rate ϕ_d [%]	capacity γ_d [PJ/yr]	loss factor $\eta_{d\{v,v'\}}$ [%]
electricity line	65	0.0	0.179	2.5
gas pipeline	100	0.0	0.123	0.1
heat pipeline	550	0.0	0.284	5

Table C.2: Initial costs, technological development rates and capacities of electricity supply assets.

asset type	cost c_{svt_0} [mEur]	tech. dev. rate ϕ_s [%]	capacity γ_s [PJ/yr]
solar supply	2.56	5.0	0.0116
wind supply	6.62	2.2	0.0543

Table C.3: Initial costs, technological development rates, capacities and efficiencies of the conversion assets.

asset type	cost c_{mvt_0} [mEur]	tech. dev. rate ϕ_m [%]	capacity γ_m [PJ/yr]	efficiency $\eta_{mee'}$ [%]
HP	0.75	1.0	0.0267	400
P2G	18	7.9	0.3974	55
CHP	2.73	0.0	0.1419	54 to heat, 36 to electricity

Table C.4: Initial costs, technological development rates, capacities and loss factors of the storage assets.

asset type	cost c_{wvt_0} [mEur]	tech. dev. rate ϕ_w [%]	capacity γ_w [PJ/yr]	loss factor η_w^{sl} [%]
electricity storage	1.12	5.0	0.0000288	96
gas storage	1.99	0.0	0.36	0.02
heat storage	0.8	1.6	0.0108	10

D
Number of joint constraints

In the case study of Eindhoven, the sets are the following size:

$$
\begin{aligned}
|E| &= 3\\
|DA| &= 3, \text{ with } |DA_e| = 1\ \forall e \in E\\
|SA| = |SA_{electricity}| &= 2\\
|SA_{gas}| = |SA_{heat}| &= 0\\
|MA| &= 3, \text{ with } |MA_{electricity}| = 2 \text{ and } |MA_{gas}| = 1\\
|DA| &= 3, \text{ with } |DA_e| = 1\ \forall e \in E\\
|A| = |DA| + |SA| + |MA| + |WA| &= 11\\
|V| &= 7, 28\text{or}110\\
|U| = \frac{1}{2}|V|(|V|-1) &= 21, 378 \text{ or } 5995\\
|DU| = |V|(|V|-1) &= 42, 756 \text{ or } 11, 990\\
|T| &= 17
\end{aligned}
$$

In the master problem of column generation per time period, joint constraint B.2 (the flow capacity constraint) holds $\forall e \in E, d \in DA_e, u \in DU, t \in T \setminus \{t_0\}$. This adds up to a total number of constraints of:

$$\sum_{e \in E} |DA_e| \cdot |DU| \cdot (|T|-1) = (1+1+1) \cdot |DU| \cdot (17-1) = 48|DU|$$

Similarly, the number of supply capacity constraints is $32|V|$, the number of conversion capacity constraints is $48|V|$, and the number of storage capacity constraints is $2 \cdot 48|V|$. Lastly, the number of convexity constraints in the master problem is $|T| = 17$. Thus, the total number of joint constraints in the master problem of column generation per time period is:

$$17 + 176|V| + 48|DU| = 17 + 176|V| + 48|V|(|V|-1) = 17 + 128|V| + 48|V|^2$$

Joint constraint (B.26), the energy balance constraint, in the master problem of column generation per energy carrier holds $\forall e \in E, v \in V, t \in T$. This adds up to $51|V|$. The same is true for column generation per asset type and per node. Furthermore, each master problem contains a convexity constraint. For column generation per energy carrier, there are $|E|$ convexity constraints, so the total number of constraints in the master problem is $3 + 51|V|$. For column generation per asset type, there are $|A|$ convexity constraints, so the total number of constraints in the master problem is $11 + 51|V|$.

For column generation per node, there are $|V|$ convexity constraints, so the total number of constraints in the master problem is $52|V|$ for the first option for implementation. For the second implementation option, the joint constraint corresponding to dual variable $\rho_{dut}\ \forall d \in DA, u \in DU, t \in T$ is added. Thus, the total number of joint constraints is for the second implementation option is:

$$52|V| + |DA| \cdot |DU| \cdot |T| = 52|V| + 51|DU| = 52|V| + 51|V|(|V|-1) = |V| + 51|V|^2$$

For the third option, the joint constraints corresponding to dual variables τ_{edut} $\forall e \in E, d \in DA_e, u \in DU, t \in T$ and ξ_{dut} $\forall d \in DA, u \in U, t \in T$ are added. Therefore, the total number of joint constraints is:

$$52|V| + 51|DU| + 51|U| = -24.5|V| + 76.5|V|^2$$

E

Greedy maximum path length

The results of the greedy algorithm when applied to the 7 nodes case are given in Table E.1. The results for both including and excluding storage are given, for all possible maximum path lengths.

Table E.1: Objective function value of the solution found by the greedy algorithm and the needed run time, for different maximum path lengths, with storage excluded and with storage included.

Max. path length	Obj. excl. storage [M€]	Run time [s]	Obj. incl. storage [M€]	Run time [s]
1	infeasible	-	infeasible	-
2	infeasible	-	infeasible	-
3	infeasible	-	infeasible	-
4	840.57	12	863.47	14
5	838.76	56	859.94	62
6	844.27	177	859.35	210
7	848.73	599	860.40	617
8	853.61	1197	859.67	1458
9	853.61	1851	859.67	1968