



# Towards an algorithmic synthesis of thermofluid systems

Jonas B. Weber<sup>1</sup> · Michael Hartisch<sup>1</sup> · Alexander D. Herbst<sup>1</sup> · Ulf Lorenz<sup>1</sup>

Received: 21 October 2019 / Revised: 9 September 2020 / Accepted: 9 September 2020 /  
Published online: 18 September 2020  
© The Author(s) 2020

## Abstract

Individual technical components are usually well optimized. However, the design process of entire technical systems, especially in its early stages, is still dominated by human intuition and the practical experience of engineers. In this context, our vision is the widespread availability of software tools to support the human-driven design process with the help of modern mathematical methods. As a contribution to this, we consider a selected class of technical systems, so-called thermofluid systems. From a technical point of view, these systems comprise fluid distribution as well as superimposed heat transfer. Based on models for simple fluid systems as extensively studied in literature, we develop model extensions and algorithmic methods directed towards the optimized synthesis of thermofluid systems to a practical extent. Concerning fluid systems, we propose a Branch-and-Bound framework, exploiting problem-specific characteristics. This framework is then further analyzed using the application example of booster stations for high-rise buildings. In addition, we demonstrate the application of Quantified Programs to meet possible resilience requirements with respect to the systems generated. In order to model basic thermofluid systems, we extend the existing formulation for fluid systems by including heat transfer. Since this consideration alone is not able to deal with dynamic system behavior, we face this challenge separately by providing a more sophisticated representation dealing with the temporal couplings that result from storage components. For the considered case, we further show the advantages of this special continuous-time representation compared to the more common representation using discrete time intervals.

**Keywords** Engineering software tool · System synthesis problem · Thermofluid system · Mixed-integer linear programming · Resilience

---

✉ Jonas B. Weber  
jonas.weber@uni-siegen.de

Extended author information available on the last page of the article

## 1 Introduction

In light of the European Union's (EU) greenhouse gas emission reduction goal and its commitment under the climate agreement reached at the COP21 climate conference in Paris, the heating and cooling sector holds great potential for achieving these objectives. According to a report prepared by the Executive Agency for Small and Medium-Sized Enterprises (2016), heating and cooling accounted the EU's biggest energy use with 50% of final energy consumption in 2012, which corresponds to 546 Mtoe<sup>1</sup> and it is expected to remain that way. In this regard, the 'EU Heating and Cooling Strategy' announced by the European Commission points out that demand reduction and the deployment of renewable energy and other sustainable sources can reduce fossil fuel import and guarantee energy supply security, while ensuring an affordable supply of energy for the end user. In the EU, 45% of the energy consumed for heating and cooling is used in the private sector, 36% in industry and 18% in services. The authors of the report assume that each of these sectors has the potential to reduce demand and increase efficiency, especially considering that 75% of the fuel these sectors consume still comes from fossil sources. The decarbonization of the heating and cooling sector is therefore essential to meet the EU's energy and climate change objectives. However, the sector is currently still fragmented and characterized by outdated and inefficient equipment, thus offering a high degree of potential for improvement (Executive Agency for Small and Medium-Sized Enterprises 2016). At the same time, the distribution of fluids, especially water, also has a high potential for saving energy and reducing emissions. The European Commission identified that water pumps in commercial buildings, for drinking water supply, in the food industry and in agriculture alone consumed an estimated quantity of 169 TWh/a in the EU-25 countries in 2000 (Betz 2017; Falkner 2008). In his dissertation, Betz (2017) further explains that the energy consumption of all pumps is reported with 300 TWh/a and the savings potential is estimated to be 123 TWh/a which indicates that, compared to a net electricity production of 2777 TWh/a in the EU-25 countries in 2000, about 10.8% of the net electricity production is currently fed into pump drives.

With regard to the design of technical systems, empirical studies suggest that the initial decisions, i.e. combining the intended functionality, layout, used components as well as the expected loads for the future use, make up 70–85% of a system's total lifespan costs (VDI 2884 2015). In this setting, two important approaches can be identified. On one hand, there is the mathematical optimization which has produced solid results in the area of design-related tasks regarding technical (flow-based) systems using both linear and non-linear optimization. Examples are the optimization of water (D'Ambrosio et al. 2015) and gas networks (Domschke et al. 2011). All those have in common that mathematicians and engineers have chosen a specific subtask with the goal of designing a technical system which was extensively examined regarding special characteristics in the problem structure and solved

---

<sup>1</sup> Tonne of oil equivalent (toe) is a unit of energy, defined as the amount of energy released by burning one tonne (1000 kg) of crude oil. One megatone (Mtoe) corresponds to one million toe.

successfully with the help of advanced algorithms. On the other hand, there is the system simulation in engineering sciences. Simulation tools like Modelica<sup>2</sup> or Matlab/SimuLink<sup>3</sup> are often used in this context. For these, components of technical systems are typically described by differential equations and component catalogs consisting of templates are used and adapted to the considered application. Furthermore, it is important that the system topology is largely predefined.

However, despite the efforts made in both areas, the design of technical systems, especially in early stages of development, is still dominated by human intuition. Hence, our aim is to provide tools for engineers to guide their intuition by the use of quantitative, modern mathematical methods during the system design process in the form of applicable software. The advantage of this approach is that the considered methods, in contrast to conventional procedures, guarantee for global optimality within the model. The use of the tool should be similar to known simulation environments. Therefore, the goal is not to identify and solve an engineering problem but to provide engineers with tools to describe and solve their problems. In particular, they should be able to formulate tasks in their own technical language. The difference, however, is that there is no simulation of an existing system but the selection of a system from a large number of implicitly described systems, as it is done on the basis of individual studies in the field of mathematical optimization. Yet, in contrast to the thematically strongly focused individual studies, our long term goal is the development of a generic tool.

Currently, we focus on a class of technical systems designed to cover applications where a combination of fluid distribution and heat transfer is required. These systems incorporating the subtasks of heating and cooling as well as transporting fluids can be summarized under the general term ‘thermofluid systems’. However, our starting point is a simplification of thermofluid systems, so-called fluid systems which are restricted to the distribution of fluids. These provide the foundation for this paper since they have been subject of extensive research in the past. In this regard, mathematical models as well as algorithms for the design and operation have been developed. We therefore provide a short overview of this topic. Note that although gas networks generally belong to the class of fluid systems, further considerations primarily focus on the application to water networks.

In general, the optimization tasks considered in literature involve optimal operation problems, optimal design problems and combinations of both. The optimal operation task aims at operating fixed components over a certain time horizon in such a way that the customer demands are satisfied while the operation costs, typically arising from the components’ power consumption, are minimized. For the optimal design task, it has to be noted that in literature the design problem usually also assumes a fixed underlying network topology (D’Ambrosio et al. 2015). The design task is therefore restricted to component sizing, e.g. choosing appropriate diameters for the pipes of the network. However, for the remainder of this paper the term ‘design’ also refers to the task of finding optimal network topologies, which is often

<sup>2</sup> <https://modelica.org> (accessed September 20, 2019).

<sup>3</sup> <https://mathworks.com/products/simulink.html> (accessed September 20, 2019).

also referred to as layout problem (De Corte and Sørensen 2013) or synthesis level (Frangopoulos et al. 2002).

In the context of optimal operation decisions, Gleixner et al. (2012) examine the optimal stationary operation of water supply networks. The optimization of dynamic water supply systems with a given layout is addressed by Morsi et al. (2012) who introduce a Mixed-Integer Linear Programming (MILP) approach based on the piecewise linearization of non-linear constraints. Geißler et al. (2011) use a similar approach for the optimization of dynamic transport networks which, in addition to water supply network optimization, has also been applied to the example of transient gas network optimization. For a deeper insight into the optimal operation of water supply networks, we refer to Martin et al. (2012) where the topic is investigated in detail. The optimal design of water distribution networks is studied by Bragalli et al. (2012) who use a MILP approach to select pipe diameters from a predefined finite set of possibilities. Besides optimizing water networks with a given layout, Fügenschuh et al. (2014) examine the optimal layout for the application example of sticky separation in waste paper processing. In their paper, a Mixed-Integer Non-Linear Program (MINLP) for the simultaneous selection of the network topology as well as the optimal settings of each separator for the steady state is proposed. A comparable approach based on a MINLP formulation to design decentralized water supply systems for skyscrapers is used in Leise et al. (2018).

Recently, there have also been increased efforts to include resilience considerations for the design of water distributions networks. The resilience of water distribution networks from a topological perspective based on an implementation of the  $K$ -shortest paths algorithm is examined in Herrera et al. (2016). Furthermore, in Meng et al. (2018), an analysis framework for studying the correlations between resilience and topological features, exemplified for water distribution networks, is proposed. However, these are simulative approaches. In the context of (non-)linear programming, Altherr et al. (2019) investigate decentralized water distribution networks in high-rise buildings, using Branch-and-Bound to exploit the special tree-structure of the considered networks in order to obtain  $K$ -resilient systems, i.e. the operation can be ensured if at most  $K$  components break down.

Besides global optimization methods, a wide range of heuristics, especially metaheuristics, have been applied in literature. In the context of the design of water distribution networks, this includes (but is not limited to) Genetic Algorithms (Savić and Walters 1997), Simulated Annealing (Cunha and Sousa 1999), Tabu Search (Cunha and Ribeiro 2004) and Ant Colony Optimization (Maier et al. 2003). For further insight, we refer to De Corte and Sørensen (2013) and Mala-Jetmarova et al. (2018). Additionally, it should be noted that many approaches do not use an explicit mathematical formulation. Instead external solvers such as EPANET<sup>4</sup> are often applied to check for hydraulic feasibility (Altherr et al. 2019). An example for the combination of metaheuristics and (non-)linear programming is shown in Cai et al. (2001). The authors use a Genetic Algorithm to fix variables in their non-linear optimization model for water management resulting in a linear formulation.

<sup>4</sup> <https://www.epa.gov/water-research/epanet> (accessed April 28, 2020).

However, it should be noted that even if heuristic approaches may often times yield good solutions, optimality cannot be guaranteed. In order to be able to overcome this disadvantage one possibility is the application of dual methods to provide a reference measure for the solutions found by (meta-)heuristics. In this regard, Altherr (2016) uses Simulated Annealing and Dynamic Programming to obtain solutions for the design of hydrostatic transmission systems and, importantly, also provides dual bounds obtained via Lagrangean Relaxation to assess the primal solutions.

With regard to thermofluid systems, contributions exist which examine the optimization of heating, ventilation and air conditioning (HVAC) systems. Note that besides other methods being also commonly applied, we focus on the application of (non-)linear programming techniques. For example, in Pöttgen et al. (2016) the generation-side of an already existing heating circuit at a conference center in Darmstadt (Germany) is examined and the authors propose alternative system designs. In Gustafsson (1998), another approach is taken. Instead of the HVAC system, the corresponding building is retrofitted based on life-cycle analysis.

Furthermore, there is the research area of Model Predictive Control (MPC) for HVAC systems. In MPC, a system model is combined with forecasts of external parameters and the resulting optimization problem of finding control decisions is typically solved online and in real time (Risbeck 2018). In this context, Risbeck et al. (2015) examine the optimized equipment usage of a central heating and cooling plant including thermal energy storage systems. In Risbeck et al. (2017), the authors further propose a framework for optimizing the operational planning of HVAC systems in commercial buildings considering both a central plant as well as the building subsystem. Similar applications have also been studied in Deng et al. (2013) and Kashima and Boyd (2013). While most of these works aim to provide an optimal control focusing on the online and real-time aspect, the emphasis in this paper is on the design aspect and the integration of estimated load data is rather used in order to evaluate favorable system designs for the intended use. However, for a more detailed overview of MPC and its application to HVAC systems, we refer to Afram and Janabi-Sharifi (2014).

Another adjacent topic is the synthesis of energy systems, typically operating as cogeneration systems for the simultaneous production of heat and power or tri-generation systems with coupled cooling (Andiappan 2017). For instance, a MILP approach for the selection and sizing of a smart building system is presented in Ashouri et al. (2013). Apart from just selecting and sizing, the authors also determine operating strategies in parallel to compare different configurations. Another contribution with regard to the synthesis of energy systems is given in Voll et al. (2013). Here, a framework for the automated superstructure generation and optimization of distributed energy supply systems based on a MILP formulation is proposed. While contributions in this field also focus on heating and cooling (combined with power generation), the approaches intentionally consider a higher level of aggregation for the synthesis task than it is the scope of this paper. For further insight into the optimization approaches for energy systems, we therefore refer to Andiappan (2017).

Beyond typical flow networks, system design approaches have also been applied to the optimization of other technical systems such as gearboxes (Altherr et al. 2018b) and lattice structures (Reintjes et al. 2018).

|                |                       | ← Model Features →   |  |                                 |                                    |
|----------------|-----------------------|----------------------|--|---------------------------------|------------------------------------|
|                |                       | Basic Fluid Systems  | <i>Extension: Uncertainty (Resilience)</i> | <i>Extension: Heat Transfer</i> | <i>Extension: Dynamic Behavior</i> |
| Implementation | Development of Models | Tile 1.1 (Section 4) | Tile 2.1 (Section 6)                       | Tile 3.1 (Section 7)            | Tile 4.1 (Section 8)               |
|                | Scalability           | Tile 1.2 (Section 5) | Tile 2.2                                   | Tile 3.2                        | Tile 4.2                           |
|                | Simulative Validation | Tile 1.3             | Tile 2.3                                   | Tile 3.3                        | Tile 4.3                           |

**Fig. 1** Overview of the sub-challenges and contributions in this paper

In contrast to most approaches, our major challenge is to be able to model the synthesis of fluid-based systems in a general and consistent way similar to the widespread simulation environments such as Modelica or Matlab/Simulink and at the same time to be able to perform algorithmic optimization. The focus is that due to a modular principle, system designers should have the possibility to pick out relevant elements for their application and extend or modify them if necessary. All elements should be based on the same foundation, as it is common for the above mentioned simulation tools. With this in mind, however, the development of suitable models and methods for the design of general thermofluid systems to a practical extent is a visionary challenge. Therefore, the decomposition into sub-challenges, as shown in Fig. 1, is necessary. Starting in the upper left corner with the basic fluid system model, we can unfold our investigation in two different dimensions. The first dimension is the extension of the fluid system model in order to include additional features. This comprises the consideration of uncertainty, in particular resilience, heat transfer as well as dealing with dynamic system behavior. The second dimension is the degree of implementation, from the formulation of suitable models and model extensions for which instances can be solved on a laboratory scale with the help of standard solvers, to the development of sophisticated heuristics and algorithms exploiting the system-specific features for handling larger instances, to the validation of proposed solutions by means of detailed simulation.

The sub-challenges examined in this paper are indicated by tiles with the specification of a section number. Resulting from these selected sub-challenges, the following research questions arise:

1. How can engineering knowledge be integrated into the solution process through domain-specific primal as well as dual methods and thus improve it compared to the use of standard solvers?

2. How can the approach be extended to include, if necessary, the consideration of resilience as an additional technical requirement for individual synthesis tasks?
3. How can heat transfer and especially the resulting need to consider dynamic effects, which are highly relevant for many engineering applications, be integrated effectively into the existing model framework?

In order to address these questions and to provide substantial progress for the overall vision, our contributions in this paper are:

- We develop heuristics for the system synthesis of fluid systems and combine them in a Branch-and-Bound framework which yields promising results and allows the synthesis of larger systems compared to the use of standard solvers. Also, we highlight the need for dual bounds in order to evaluate the primal solutions. Another special aspect of the heuristics is the integration of implicit engineering knowledge about the specific properties of the considered technical systems for the algorithmic optimization, thus pointing out the potential of the interdisciplinary approach.
- We propose an approach which enables the consideration of resilience in the considered setting as a subsequent design decision, hence it is also possible to increase the resilience of existing systems.
- We present extensions that allow the integration of heat transfer and dynamic effects for the synthesis of thermofluid systems. In order to handle the dynamic effects a novel approach which considers a variable length of time steps is presented. The extent to which the associated restrictions are reasonable can only be decided from an engineering perspective which again underlines the necessity of the interdisciplinary approach.

The paper contains research which has been partly presented in conference proceedings, see Hartisch et al. (2018), Weber and Lorenz (2017), Weber and Lorenz (2019a), Weber and Lorenz (2019b) and Weber et al. (2020), and is organized as follows: Sect. 2 provides a deeper insight into a possible software tool for engineers to design technical systems as well as the associated workflow. After the general technical and physical background as well as selected system components are discussed in Sect. 3, the basic MILP formulation for the design of fluid systems, as described in literature, is presented in Sect. 4. Based on this formulation, we propose a Branch-and-Bound framework containing a relaxation based on technical problem-specific characteristics in Sect. 5. The framework is then applied to the example of booster stations for high-rise buildings to solve practical examples and the performance of the framework is discussed using these instances. The application case of booster stations is used again in Sect. 6 to demonstrate an approach based on Quantified Programming in order to meet resilience requirements with respect to given system designs as they can be generated by the presented Branch-and-Bound framework. In Sect. 7, we extend the existing fluid system model by including heat transfer in order to describe basic thermofluid systems. A further important extension, the consideration of dynamic system behavior, is discussed in Sect. 8. Whereas the models and extensions previous to Sect. 8 only consider a sequence of system states without temporal couplings, we

present an approach for the description of the dynamic system behavior, e.g. caused by storage components, with a focus on its practical applicability. Finally, to conclude the paper, we discuss our results and directions for future research in Sect. 9.

## 2 Supporting the system design process by optimization

We aim to establish quantitative, modern mathematical methods during the system design process by developing specialized tools for engineers. Apart from transferring these methods and projecting them onto the application of designing technical systems, a systematic workflow with a strong focus on providing engineers with the methodical procedures to exploit the corresponding quantitative methods is required. Here, we therefore discuss a suitable systematical design approach for this project and present a possible software implementation.

### 2.1 Systematical design approach

In an attempt to automatically find optimal pump system designs, Pelz et al. (2012) propose a systematical design approach in order to combine planning and engineering approaches with mathematical optimization. By guiding the designer through specific steps, the approach prepares the generation and solution of an optimization program and structures the application of the optimization results to reality. This approach divides the problem development process into seven steps which can be split into two phases—a deciding phase and an acting phase:

#### *DECIDING*

1. What is the system's function?
2. What is my goal?
3. How large is the playing field?

#### *ACTING*

4. Find the optimal system!
5. Verify!
6. Validate!
7. Realize!

The degree of detail is continuously refined from step to step. Steps 1, 6 and 7 describe the common planning process of an engineer or system designer and are further supplemented by additional intermediate steps relevant to the system design in order to streamline the planning process, facilitate the communication between the interest groups involved and therefore catalyze the generation of optimal solutions.

The first step is to determine the system's function. For the function all relevant components of the system which are involved in the fulfillment of its



purpose as well as the load history are of importance. Typical functions are, as in this paper, the transport of material, heating or cooling.

Subsequently, in step 2, the intended primary goal has to be concretized. This step is of great importance since the goal massively influences the final solution of the problem. The goal can vary depending on the interest groups involved. For example, the goal of an investor can be a low net expense. The operator, on the other hand, could consider a high availability, while a state institution could focus on a low energy consumption or low pollutant emissions as a priority. Since these goals can be conflicting, a completely different system may emerge depending on the goal but one that is optimal in its area. For this reason, the definition of the goal must also be seen as a subjective influence on the optimal system and must therefore be formulated in agreement with all relevant interest groups. This step is often neglected in practice.

The third and last step of the deciding phase is to determine the size of the playing field. This is the framework in which a system is to be optimally designed. In the case of a technical system, the components must be preselected. An algorithm takes over the task of selecting components from a pool of different components and making optimal use of them for the overall system. The delimitation of the playing field represents an important restriction for the possible solutions and must be carefully defined in mutual consultation with all interest groups. It must be clear that the approach will only find technical solutions that are part of the playing field and therefore will not replace human imagination or find creative solutions beyond the possible solutions. This underlines the intention of this approach which has to be seen as a decision tool and should not replace the decision maker.

At the end of the deciding phase, all decisions by the users which influence the optimal system have been made. Thus, the formulation of the system's requirements is completed.

After the requirements for the system have been defined, the next step is the computation of a system proposal. This is done by setting up mathematical models and applying algorithms to solve them. The consideration is not limited to the components themselves but also to their design and it depends heavily on the defined playing field. Thus, a system proposal can be found with regard to topology and control parameters and is then converted into a physical model. With this approach, a global optimum with respect to the initial decisions made in the deciding phase cannot always be found within reasonable time. In general, however, it is not necessarily a question of finding the optimal system but of generating the best possible system and being able to estimate its quality. In practice, systems that are proven to be among the best percentage of possible solutions are often more than sufficient.

Following the algorithmic search for a system, the suggested solution needs to be reviewed and verified by the system designer. This is done by models with concentrated parameters, so-called 0D-models (Betz 2017), and stands in close interaction with the previous step.

Finally, the two last steps of the second phase are the validation of the system with experiments or higher-dimensional computational models and the subsequent realization.

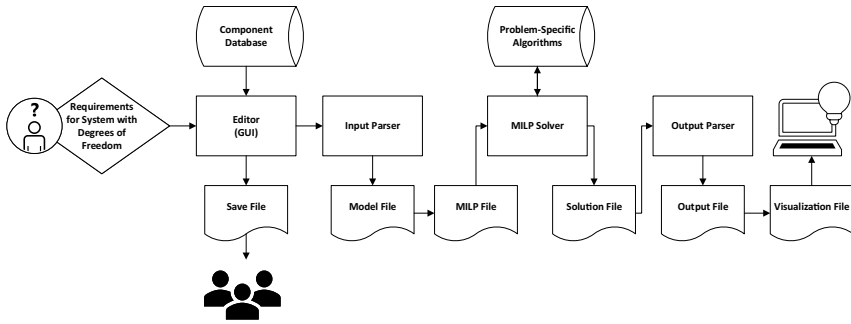


Fig. 2 Schematic workflow of the software tool

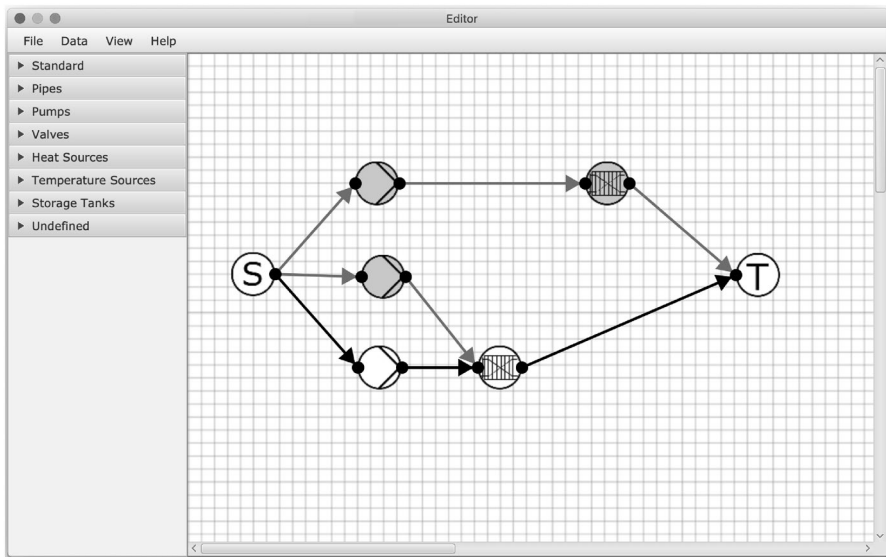


Fig. 3 Graphical user interface for the implicit system description

## 2.2 Software framework

In Pelz et al. (2012) and Saul et al. (2016), a realization of the approach described above is presented. In this context, we focus on how the desired workflow, illustrated in Fig. 2, can be implemented by the use of software.

The process starts with a user, who is typically an engineer or system designer, defining the system requirements as well as the possible degrees of freedom for the system according to the design approach presented above. For this purpose, we propose a customized graphical user interface (GUI), as sketched in Fig. 3. On the left sidebar the user can select components from a component catalog and drag-and-drop them onto the drawing board. The component catalog is connected to a database in

which the components are stored with their corresponding technical descriptions and characteristics. Degrees of freedom can be expressed by using ‘optional’ components and connections as indicated by the light grey highlighted symbols in Fig. 3. After the implicit system structure has been defined, the load related requirements to be met by the system, i.e. certain load scenarios as well as the intended objective, cost or energy minimization, can be specified in the corresponding submenus of the ‘Data’ tab. In this context, each load scenario describes a set of measurement points of the system that must reach certain values at defined points in time.

After the necessary requirements and degrees of freedom have been defined, the next step of the intended workflow is the creation of a MILP instance from the graphical representation. This also involves the intermediate step of a higher-level model file that aims for easier human readability. The mathematical program can then be solved using a combination of standard MILP techniques and state-of-the-art solvers as well as problem-specific algorithms and procedures.

Following this, the optimization result can be saved and reinterpreted graphically. Finally, the user is able to customize and examine the proposed system according to the last three steps of the design approach presented above.

### 3 Physical and technical background

While fluid systems only deal with the distribution of fluids, thermofluid systems fulfill two subtasks, the distribution as well as heating and cooling of fluids. The relevant physical quantities to describe such systems are the volume flow  $\dot{V}$  [ $\text{m}^3/\text{s}$ ], the pressure  $H$  [bar] (or [m]), the heat flow  $\dot{Q}$  [J/s] and the temperature  $T$  [ $^\circ\text{C}$ ]. While the pressure is related to the distribution, the heat flow and the temperature are necessary to describe heating and cooling and the volume flow couples both subtasks. Besides that, there are different groups of technical components involved to fulfill the individual subtasks. The relevant physical relationships necessary to describe the system behavior as well as selected technical components used in such systems are briefly explained below.

#### 3.1 Continuity equation

All fluid distribution systems must satisfy the continuity equation. It states that the transported mass through a flow tube remains constant in the case of steady state flows. This criterion meets the general principal of mass conservation which claims that the inlet mass flow must be equal to the outlet mass flow. In fluid mechanics, this can be expressed as follows with  $\dot{m}$  representing the mass flow, i.e. the time derivative of mass,  $\rho$  representing the density of the fluid,  $v$  representing the flow velocity and  $A$  representing the cross-sectional area of the flow tube (Munson et al. 2009):

$$\dot{m} = \rho_1 \cdot v_1 \cdot A_1 = \rho_2 \cdot v_2 \cdot A_2 \quad (1)$$

If the term  $v \cdot A$  is replaced by the volume flow  $\dot{V}$ , the equation can be stated as:

$$\dot{m} = \rho_1 \cdot \dot{V}_1 = \rho_2 \cdot \dot{V}_2 \quad (2)$$

In the case of incompressible fluids—like water—the relation can be simplified because of the pressure-independent density:

$$\dot{V} = \dot{V}_1 = \dot{V}_2 \quad (3)$$

This relation holds for ideal systems without losses and is applicable for a system itself as well as for single components.

### 3.2 Bernoulli's equation

Furthermore, Bernoulli's equation, which is derived from the general conservation of momentum, applies. For steady state motions of frictionless (ideal), incompressible fluids that are not effected by external forces except for gravity, the Bernoulli energy equation holds (Munson et al. 2009):

$$\frac{v_1^2}{2g} + \frac{H_1}{\rho g} + z_1 = \frac{v_2^2}{2g} + \frac{H_2}{\rho g} + z_2 = \text{const.} \quad (4)$$

Here,  $v$  is the fluid flow velocity at a point on a streamline,  $g$  is the acceleration due to gravity,  $z$  is the elevation of the point above a reference plane,  $H$  is the pressure at the chosen point and  $\rho$  is the density of the fluid. If this equation is multiplied by  $\rho$  and  $g$ , this results in the Bernoulli pressure equation:

$$H_1 + \rho \cdot g \cdot z_1 + \frac{\rho}{2} \cdot v_1^2 = H_2 + \rho \cdot g \cdot z_2 + \frac{\rho}{2} \cdot v_2^2 = \text{const.} \quad (5)$$

Furthermore, the pressure increase (or decrease)  $\Delta H_C$  must be considered separately if a pressure-modifying component is used between the points 1 and 2:

$$H_1 + \rho \cdot g \cdot z_1 + \frac{\rho}{2} \cdot v_1^2 + \Delta H_C = H_2 + \rho \cdot g \cdot z_2 + \frac{\rho}{2} \cdot v_2^2 \quad (6)$$

### 3.3 Specific heat formula

The physical quantities related to heating and cooling are coupled by the specific heat formula. In this regard, the specific heat is the amount of heat per unit mass required to raise the temperature by one Kelvin (or degree Celsius) (Incropera et al. 2007). The relationship is typically expressed as:

$$\Delta Q = m \cdot c \cdot \Delta T \quad (7)$$

Here,  $c$  is the specific heat. As an example, the specific heat of water—the common substance with the highest specific heat—is about 4182 joule per kilogram and Kelvin at a temperature of 20°C. However, the relationship does not hold if phase changes occur due to the fact that heat added or removed during a phase change does not change the temperature. By assuming a constant density  $\rho$  of about one kilogram per liter, the mass can be replaced in terms of volume with  $V = m/\rho$ . Furthermore, the above equation can be stated on a flow rate basis with  $T$  being the temperature

difference with respect to a predefined reference temperature. Taking these considerations into account, Eq. (7) can be rewritten as:

$$\dot{Q} = \dot{V} \cdot c \cdot T \quad (8)$$

### 3.4 Mixing of fluids

If (possibly different) fluids with different temperatures are mixed, the mixing temperature  $T_M$  of  $|N|$  fluids being mixed can be calculated as:

$$T_M = \frac{\sum_{i \in N} m_i \cdot c_i \cdot T_i}{\sum_{i \in N} m_i \cdot c_i} \quad (9)$$

As a simplification for water, the mass can be estimated by  $m = V \cdot \rho$  with the assumption of  $\rho \approx 1$ . Additionally, since only one kind of fluid is mixed and the specific heat terms  $c_i$  in each summand of both sums are assumed to be uniquely constant, they cancel each other out. Furthermore, the equation can be formulated on a flow rate basis and the numerator can be rewritten using Eq. (8):

$$T_M = \frac{\sum_{i \in N} \dot{V}_i \cdot T_i}{\sum_{i \in N} \dot{V}_i} = \frac{\sum_{i \in N} \dot{Q}_i}{\sum_{i \in N} \dot{V}_i} \cdot \frac{1}{c} \quad (10)$$

### 3.5 Components

There is a wide variety of different components used in thermofluid systems depending on the respective field of application. Again, a general distinction can be made between components related to the distribution, such as pumps or valves, and those related to heating and cooling. For the latter, two ideal sources of thermal energy can be distinguished: ideal heat sources and ideal temperature sources. An ideal heat source is able to deliver a constant, predefined heat flow independent of the inlet or outlet temperature as well as the volume flow. An example for components which can be seen as ideal heat sources are simple heaters. An ideal temperature source, in contrast, can maintain a predefined temperature at its outlet independent of the heat flow required as well as the inlet temperature and the volume flow. It therefore produces a constant absolute temperature. Components which can be modeled as an ideal temperature source are for instance heat exchangers for district heat as examined in Pöttgen et al. (2016). In this paper, all heating and cooling components are associated with one of the two thermal energy sources.

In the following, the operation principles and the procedure for modeling two exemplarily selected component types, pumps as representatives of the distribution side and chillers as examples for the heating and cooling side, are described.

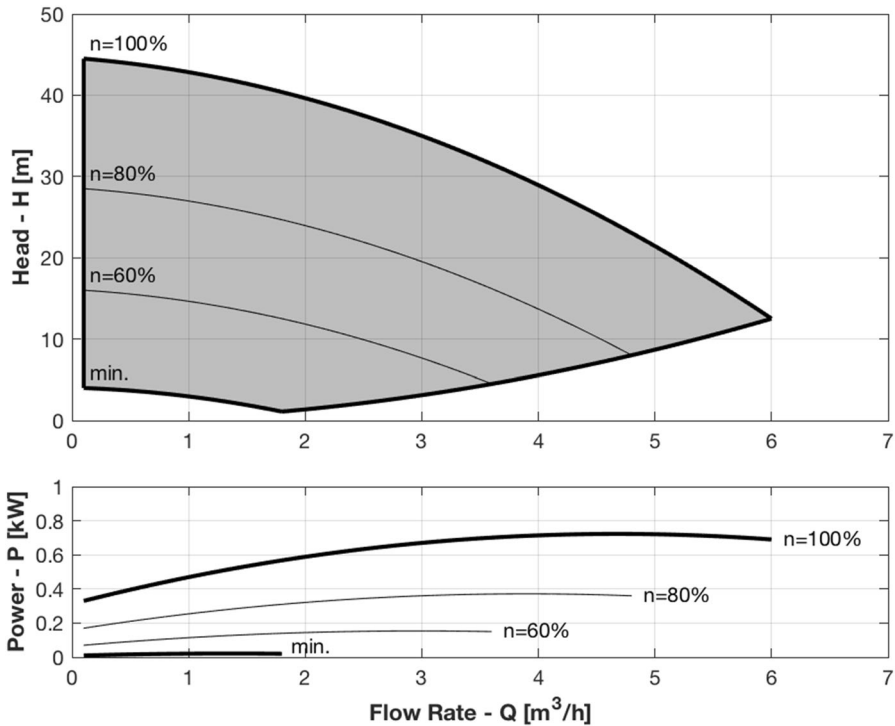


Fig. 4 Exemplary characteristic map of a speed controlled pump

### 3.5.1 Pumps

In general, pumps have an opposite relation between their volume flow and pressure increase. With increasing volume flow the possible pressure increase decreases. Additionally, the power consumption  $P$  of pumps increases with increasing volume flow. There are basically three different classes of pumps, resulting from their speed control. These are pumps with constant speed, with stepped speed control and pumps with continuously variable speed control. The operation of a constant speed pump is fairly straightforward. For speed controlled pumps the possible pressure increase as well as their power consumption rises with increasing rotational speed  $n$  if the volume flow is held constant. This can be described by the so-called affinity laws:

$$\dot{V} \sim n, \Delta H \sim n^2 \text{ and } P \sim n^3 \quad (11)$$

Fixing any two of those variables determines the remaining ones. For variable speed controlled pumps this relation is manifested in their respective characteristic curves, see Fig. 4. The operation can be described by quadratic and cubic approximations with regression coefficients  $a_i$ ,  $b_i$ ,  $c_i$  and  $d_i$  to determine the pressure increase and power consumption for a given flow-speed-tuple (Ulanicki et al. 2008)

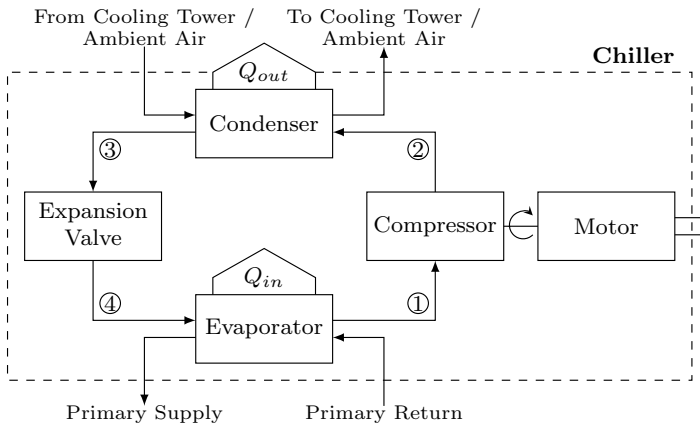


Fig. 5 Working principle of a compression chiller

$$\Delta H = a_1 \cdot \dot{V}^2 + b_1 \cdot \dot{V} \cdot n + c_1 \cdot n^2 \quad (12)$$

$$P = a_2 \cdot \dot{V}^3 + b_2 \cdot \dot{v}^2 \cdot n + c_2 \cdot \dot{V} \cdot n^2 + d_2 \cdot n \quad (13)$$

Single pumps or whole subsystems can be connected pairwise either in series or in parallel. If, on one hand, modules are connected in series, the total pressure increase results as the sum of the single pressure increases, while the flow through them remains constant. If, on the other hand, modules are connected in parallel, the pressure increase remains constant and the total volume flow through both modules is the sum of the single volume flows.

### 3.5.2 Chillers

Many different chiller types exist. However, a rough classification between two types, vapor absorption and vapor compression chillers, can be made. In the following, we concentrate on the latter. This type can again be subdivided into centrifugal, reciprocating, scroll and screw chillers by the compressor technologies used. Finally, those can be further classified into water-cooled and air-cooled chillers, depending on a chiller's heat sink. All those types have in common that the cooling is realized by a circular process consisting of four subprocesses, as shown in Fig. 5. In the first step, the internal refrigerant enters the evaporator as a liquid-vapor mixture and absorbs the heat of the cooling medium returning from the heat source (1). The vaporous refrigerant is then sucked in and compressed while the resulting heat is absorbed by the refrigerant (2). During the subsequent liquefaction process, the superheated refrigerant enters the condenser, is cooled by the ambient air or water of a cooling tower and liquefies again (3). Finally, in

the expansion process, the pressure of the refrigerant is reduced from condensing to evaporating pressure and the refrigerant expands again (4).

To model the specific operation of a chiller, the ‘DOE2’ electric chiller simulation model, as examined in Hydeman et al. (2002), can be used. This model is based on the following performance curves:

$$CAP_{FT} = a_1 + b_1 \cdot t_{chws} + c_1 \cdot t_{chws}^2 + d_1 \cdot t_{cws/oat} + e_1 \cdot t_{cws/oat}^2 + f_1 \cdot t_{chws} \cdot t_{cws/oat} \quad (14)$$

$$EIR_{FT} = a_2 + b_2 \cdot t_{chws} + c_2 \cdot t_{chws}^2 + d_2 \cdot t_{cws/oat} + e_2 \cdot t_{cws/oat}^2 + f_2 \cdot t_{chws} \cdot t_{cws/oat} \quad (15)$$

$$EIR_{FPLR} = a_3 + b_3 \cdot PLR + c_3 \cdot PLR^2 \quad (16)$$

$$PLR = Q / (Q_{ref} \cdot CAP_{FT}) \quad (17)$$

$$P = P_{ref} \cdot EIR_{FPLR} \cdot EIR_{FT} \cdot CAP_{FT} \quad (18)$$

The  $CAP_{FT}$  curve, see Eq. (14), represents the available (cooling) capacity  $Q$  as a function of evaporator and condenser temperatures. The  $EIR_{FT}$  curve, see Eq. (15), which is also a function of evaporator and condenser temperatures describes the full-load efficiency of a chiller. Finally, the  $EIR_{FPLR}$  curve, see Eq. (16), represents a chiller’s efficiency as a function of the part-load ratio  $PLR$ , see Eq. (17). For the  $CAP_{FT}$  and  $EIR_{FT}$  curve, the chilled water supply temperature  $t_{chws}$  is used as an estimate for the evaporator temperature and the condenser water supply  $t_{cws}$  and outdoor dry-bulb temperature  $t_{oat}$  are used for the condenser temperature of water-cooled and air-cooled chillers, respectively. With Eqs. (14)–(17) it is possible to determine the power consumption  $P$  of a chiller for any load and temperature condition by applying Eq. (18). The operation of a given chiller is therefore defined by the regression coefficients  $a_i$ ,  $b_i$ ,  $c_i$ ,  $d_i$ ,  $e_i$  and  $f_i$ , the reference capacity  $Q_{ref}$  and the reference power consumption  $P_{ref}$  (Hydeman et al. 2002).

Within the scope of our research, chillers can be assigned to the group of temperature sources. The chilled water supply temperature is therefore assumed to be independent of the inlet temperature and the volume flow. Depending on the application the condenser water supply or outdoor dry-bulb temperature may be assumed to be constant.

#### 4 Fluid systems (Tile 1.1)

All fluid systems have two things in common: First, each system contains a fluid that moves through a system of connected pipes and other components. Second, a pressure difference in the system causes fluids to move. Hence, pressure is the driving



force in fluid systems. Nevertheless, it should be noted that the focus of this paper is on water-based systems. The density of water is assumed to be constant which is a common simplification. However, for gas networks this simplification is typically not applicable (Geißler et al. 2011). Therefore, different models and methods have to be used to tackle these optimization problems.

The general system synthesis task considered in this paper can be stated as follows: Given a construction kit of technical components as well as a technical specification of load collectives, compare all valid systems and choose the one for which the lifespan costs—the sum of purchase costs and the expected energy costs—are minimal. In this context, a system is called a valid system if it is able to satisfy every prospected load. We assume that the transition times and therefore also the transition costs between the load changes are negligible compared to the total costs. Hence, corresponding models can be stated as quasi stationary. Each load out of the load collective is called a load scenario. A load scenario consists of two components: the time interval of the system’s operational life for this scenario as well as the demanded values for the respective physical quantities at certain points in the system.

The decision making can be abstracted in two ways. On one hand, it can be stated using linear (and non-linear) constraints as a MI(N)LP. Hence, the decisions of the optimization problem can be described by variables: first and second stage variables. In the first stage the optimization program must decide whether a component is needed and thus bought. In the second stage, a bought component can be turned on/off and possibly speed controlled to cover all load scenarios during the system’s operation. On the other hand, the problem can be abstracted as a source-target-network  $(G, S_G, T_G)$  with a complete graph  $G = (V, E)$ , vertices  $V$  and edges  $E$ , whereas  $S_G, T_G \in V$  are distinguished sets of vertices, namely the sources and the sinks of the network. An edge represents a component from the construction kit and a vertex represents a possible connection between components. The complete graph of the construction kit, consisting of all components that may be installed, plus the sources and sinks contains every possible system. Therefore, each system can be modeled by a subgraph of the complete graph representing the decisions made for the system.

The optimization model for fluid systems presented here is based on the models available in literature (see e.g. Betz 2017; Geißler et al. 2011; Pelz et al. 2012; Pöttgen and Pelz 2016). It serves as a starting point for the step-by-step extension according to Fig. 1. All variables and parameters used are shown in Table 1.

$$\min \sum_{(i,j) \in E} (C_{i,j}^{buy} \cdot b_{i,j}) + C^{kWh} \cdot OLS \cdot \sum_{s \in S} \left( F^s \cdot \left( \sum_{(i,j) \in E} p_{i,j}^s \right) \right) \tag{19}$$

$$a_{i,j}^s \leq b_{i,j} \quad \forall s \in S, (i,j) \in E \tag{20}$$

$$\dot{v}_{i,j}^s \leq \dot{V}^{max} \cdot a_{i,j}^s \quad \forall s \in S, (i,j) \in E \tag{21}$$

**Table 1** Variables and parameters of the optimization model for fluid systems

| Symbol                           | Range            | Description  |
|----------------------------------|------------------|--|
| $b_{i,j}$                        | $\{0, 1\}$       | Purchase decision of component $(i, j)$  |
| $a_{i,j}^s$                      | $\{0, 1\}$       | Activation decision of component $(i, j)$ in scenario $s$                      |
| $\dot{v}_{i,j}^s$                | $\mathbb{R}_0^+$ | Volume flow through component $(i, j)$ in scenario $s$                         |
| $h_k^s$                          | $\mathbb{R}_0^+$ | Pressure at port $k$ in scenario $s$   |
| $p_{i,j}^s$                      | $\mathbb{R}_0^+$ | Power consumption of component $(i, j)$ in scenario $s$                        |
| $n_{i,j}$                        | $[0, 1]$         | Relative speed (or level) of component $(i, j)$ in scenario $s$                |
| $\Delta h_{i,j}^s$               | $\mathbb{R}$     | Pressure increase/decrease by component $(i, j)$ in scenario $s$               |
| $S$                              | –                | Set of scenarios   |
| $E$                              | –                | Set of components  |
| $V$                              | –                | Set of ports   |
| $S_G$                            | –                | Set of sources   |
| $T_G$                            | –                | Set of sinks   |
| $C^{kWh}$                        | –                | Energy costs per kilowatt hour   |
| $C_{i,j}^{buy}$                  | –                | Purchase costs of component $(i, j)$   |
| $OLS$                            | –                | Operational lifespan of the system   |
| $F^s$                            | –                | Fraction of the operational lifespan $OLS$ of scenario $s$                     |
| $\dot{V}^{max}$                  | –                | Upper bound on the volume flow   |
| $H^{max}$                        | –                | Upper bound on the pressure  |
| $\dot{V}_{in/out k}^{min/max s}$ | –                | Lower/upper bound on the volume flow entering/leaving port $k$ in scenario $s$ |
| $H_k^{min/max s}$                | –                | Lower/upper bound on the pressure at port $k$ in scenario $s$                  |

$$h_k^s \leq H^{max} \quad \forall s \in S, k \in V \tag{22}$$

$$\sum_{(i,k) \in E} \dot{v}_{i,k}^s - \sum_{(k,j) \in E} \dot{v}_{k,j}^s = 0 \quad \forall s \in S, k \in V \setminus \{S_G, T_G\} \tag{23}$$

$$h_j^s - h_i^s \leq \Delta h_{i,j}^s + (1 - a_{i,j}^s) \cdot H^{max} \quad \forall s \in S, (i,j) \in E \tag{24}$$

$$h_j^s - h_i^s \geq \Delta h_{i,j}^s - (1 - a_{i,j}^s) \cdot H^{max} \quad \forall s \in S, (i,j) \in E \tag{25}$$

$$\dot{V}_{out k}^{min s} \leq \sum_{(k,j) \in E} \dot{v}_{k,j}^s \leq \dot{V}_{out k}^{max s} \quad \forall s \in S, k \in V \tag{26}$$

$$\dot{V}_{in k}^{min s} \leq \sum_{(i,k) \in E} \dot{v}_{i,k}^s \leq \dot{V}_{in k}^{max s} \quad \forall s \in S, k \in V \tag{27}$$

$$H_k^{min s} \leq h_k^s \leq H_k^{max s} \quad \forall s \in S, k \in V \tag{28}$$

$$\Delta h_{ij}^s = \Delta H(a_{ij}^s, n_{ij}^s, v_{ij}^s) \quad \forall s \in S, (i, j) \in E \quad (29)$$

$$p_{ij}^s = P(a_{ij}^s, n_{ij}^s, v_{ij}^s, \Delta h_{ij}^s) \quad \forall s \in S, (i, j) \in E \quad (30)$$

The objective of the optimization model is to minimize the sum of investment costs and expected energy costs over a system's lifespan, see Objective (19). A component can only be used to satisfy a load scenario if it is installed, see Constraint (20). If a component is operational, its volume flow is reasonable or vanishes otherwise, see Constraint (21). Similarly, the pressure head must be reasonable at each port, see Constraint (22). Due to the law of flow conservation, the volume flow has to be preserved at all vertices, except for the sources and sinks, see Constraint (23). If a component is operational, the pressure propagation has to be ensured. In case of pumping components the pressure increase caused by the component increases the pressure at its outlet and therefore the adjacent system pressure, see Constraints (24) and (25). For non-pumping components the pressure increase  $\Delta h_{ij}^s$  is typically 0. Constraints (26)–(28) enable the setting of target values for the volume flow and pressure at certain points in the system. The generally non-linear operating behavior of components and the determination of their respective operating points is represented by Constraints (29) and (30). For the example of pumps, the associated relationships are shown in Sect. 3.5.

In principle, the model presented above is a MINLP due to the non-linear relationship resulting from the non-linear constraints for describing the component behavior. Unfortunately, MINLPs are in general hard to solve or even intractable (Geißler 2011). The corresponding constraints are therefore piecewise linearly approximated to make them accessible for MILP techniques. The implementation is straightforward and the linearization techniques used follow those presented in Vielma et al. (2010).

## 5 Algorithmic synthesis of fluid systems (*Tile 1.2*)

In the following, we present our contribution to the algorithmic synthesis of fluid systems on a larger scale, i.e. an algorithmic system design process for instances of practical interest. The goal is to generate 'good' systems in reasonable time. In this context, 'good' refers to solutions with a desirable objective function value and the runtimes should allow for practical applicability. However, the usual procedure to simply generate the corresponding MILP and to solve it using a standard MILP solver fails to solve such instances in reasonable time because of the inability to provide strong dual bounds. Therefore, we develop a problem specific approach exploiting the special system characteristics by primal and dual heuristics. In order to maintain a certain practical relevance, we examine the application case of so-called booster stations. According to the principles of Algorithm Engineering, as explained in Sanders (2009), this is an important feature since applications play an important role for the development of algorithms and serve as realistic inputs for meaningful experiments. In addition, as in this case, not all future applications for

the algorithms to be included to a library are known in advance, therefore providing algorithms validated on related applications with realistic inputs is an important factor (Sanders 2009).

The basic idea is as follows: Use both the MILP and the graph view simultaneously and benefit from both. On the primal side, we use heuristics, especially local search algorithms, to obtain good primal solutions. In this paper, we focus on Simulated Annealing but other local search algorithms, e.g. Genetic Algorithms or Tabu Search, are possible, too. In this step, the graph representation is used to define neighborhoods and the MILP representation is used to evaluate the quality of the generated systems. On the dual side, we use a heuristic which is based on problem specific and technical knowledge to relax the generated MILP. Doing so, we obtain lower bounds. Finally, both heuristics are combined in a Branch-and-Bound framework to close the optimality gap between the primal and dual solutions. Thus, we can obtain provable optimal solutions for the system design.

### 5.1 Primal heuristic: Simulated Annealing

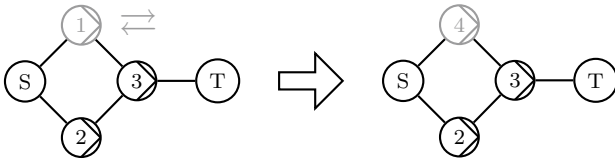
The implemented Simulated Annealing algorithm follows Boussaïd et al. (2013) with some modifications: Previous calculations are saved and a penalty term for non-valid system topologies is implemented. The algorithm is used to find good topologies for the first stage of the two-staged optimization problem (the topology problem) as described in Sect. 4. After generating a topology, the binary first stage variables are fixed in the MILP. Afterwards, the second stage (the operation problem) is solved optimally for the chosen topology regarding the different load scenarios using a standard solver. For the topology decision only series-parallel networks, as defined in MacMahon (1890), are considered to ensure that only technically sound topologies are generated, e.g. each technical component has at least one successor and one predecessor in the network.

The problem specific neighborhood function necessary for Simulated Annealing consists of four single neighborhoods, similar to the operators used in Altherr (2016). These are the replace ( $N_{Replace}$ ), the swap ( $N_{Swap}$ ), the add ( $N_{Add}$ ) and the delete neighborhood ( $N_{Delete}$ ):

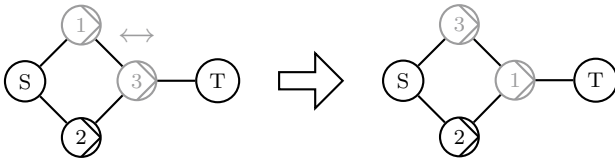
$$N = N_{Replace} \cup N_{Swap} \cup N_{Add} \cup N_{Delete}$$

Illustrative examples for the respective neighborhoods described below are shown in Fig. 6.

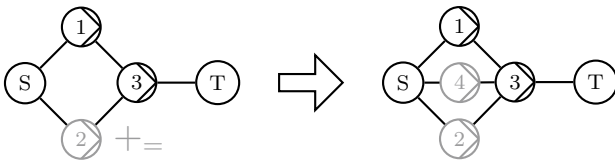
- $N_{Replace}$ : A component  $p_i$ , in the case of booster stations a pump, of the set of bought components—a subset of the available construction kit—is selected randomly and replaced by a component  $p_j$  from the set of unbought components. The previous predecessors and successors of  $p_i$  are the new predecessors and successors of  $p_j$ . This neighborhood can only be created if the network consists of at least one component and there is at least one unbought component.
- $N_{Swap}$ : Two different components  $p_i$  and  $p_j$  of the set of bought components are selected randomly.  $p_i$  and  $p_j$  swap positions in the network. The previous prede-



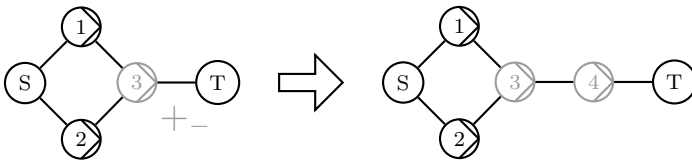
(a)  $N_{Replace}$ : Pump 1 is replaced by pump 4



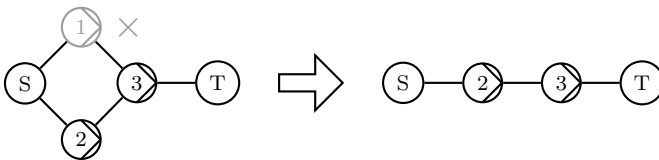
(b)  $N_{Swap}$ : Pump 1 and pump 3 swap positions



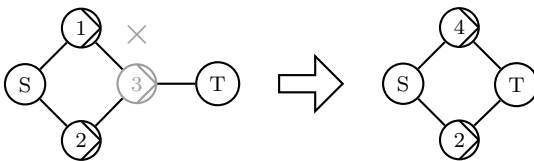
(c)  $N_{Add}$ : Pump 4 is added in parallel to pump 2



(d)  $N_{Add}$ : Pump 4 is added in series behind pump 3



(e)  $N_{Delete}$ : Pump 1 is deleted without modified connections of predecessors and successors



(f)  $N_{Delete}$ : Pump 3 is deleted with modified connections of predecessors and successors

**Fig. 6** Exemplary illustration of **a**  $N_{Replace}$ , **b**  $N_{Swap}$ , **c**  $N_{Add}$ —parallel case, **d**  $N_{Add}$ —serial case, **e**  $N_{Delete}$ —without modification and **f**  $N_{Delete}$ —with modification

cessors and successors of  $p_i$  are the new predecessors and successors of  $p_j$  and vice versa. This neighborhood can only be created if the network of bought components consists of at least two components.

- $N_{Add}$ : A component  $p_i$  of the set of unbought components is selected randomly and it is decided whether  $p_i$  is connected in series or in parallel. If  $p_i$  is connected in series, a component out of the set of bought components, a source or a sink is selected. If a source or a sink is selected,  $p_i$  is connected in series behind the source or before the sink. If a component  $p_j$  is selected,  $p_i$  is connected before or behind  $p_j$ . The source, the sink or  $p_j$  becomes the new predecessor or the new successor of  $p_i$ . Furthermore,  $p_i$  adopts their previous successors or predecessors. If  $p_i$  is connected in parallel, a component  $p_j$  of the set of bought components is selected. All predecessors and successors of  $p_j$  become the predecessors and successors of  $p_i$  as well. This neighborhood can only be created if the set of unbought components consists of at least one component and in the case of a parallel connection if the set of bought components consists of at least one component.
- $N_{Delete}$ : A component  $p_i$  of the set of bought components is selected randomly and is deleted from the network. If a predecessor  $p_{i,p}$  or a successor  $p_{i,s}$  of  $p_i$  only has  $p_i$  as its successor or predecessor, a successor or predecessor of  $p_i$  is selected randomly. It then becomes the new successor or predecessor of  $p_{i,p}$  or  $p_{i,s}$ . This is necessary to ensure the flow conservation. Otherwise the connection is deleted without substitution. This neighborhood can only be created if there is at least one component in the set of bought components.

To generate a starting solution, a simple heuristic is used which is based on  $N_{Add}$  to obtain valid solutions. First, a minimal network including only the sources and sinks is considered. If this network is already a valid solution, it is accepted as the starting solution. Otherwise, components are added until a valid topology is generated. If the set of unbought components is empty and the solution is still not valid, the whole network is deleted and the procedure starts again with a minimal network until a valid solution is found.

For the considered problem, non-valid solutions have no associated costs. If the costs were set to  $+\infty$ , the algorithm would never accept them as the current solution. In this case, it would not be possible to reach every solution in the solution space with the defined neighborhood function. To avoid this, a penalty term is introduced assigning costs to non-valid solutions. If a solution is non-valid, double the costs of the starting solution are used instead. This approach has two advantages: First, the costs are low enough that non-valid solutions can be used as current solution in the algorithm and second, high enough that they should be greater than the costs of all valid solutions.

The critical steps for the runtime of the algorithm are the calculations for the optimal operation mode for the found topologies performed by the MILP solver. To enhance the runtime of the algorithm a list is created which holds the last solutions. Every time a calculation is needed, the list is checked first whether this topology has already been calculated. If not, the system is added to the list. If the list reaches the defined maximum size, the oldest entry is deleted such that new solutions can be stored.

In addition, a cooling schedule has to be determined. We use an exponential cooling function  $T(t) = T_0 \cdot \alpha^t$  which is widely used in literature (Boussaïd et al. 2013). Here,  $T_0$  is the starting temperature and  $t$  indicates the number of temperature reductions performed. The parameter  $\alpha$  is a value between 0 and 1. It influences the slope of the cooling function. A threshold value  $T_{stop}$  acts as a termination criterion. As soon as the temperature falls below this threshold value, the algorithm terminates. Furthermore, the number of iterations per temperature level has to be chosen in such a way that the search space is explored sufficiently. These parameters have to be determined experimentally depending on the specific problem. For our experiments a value of  $\alpha = 0.9$  showed good results ensuring a balance between runtime and exploration of the search space. The start temperature  $T_0$  was set to 10,000. For the considered instances, especially with regard to the dimensions of the occurring costs, this proved particularly suitable to ensure both sufficient diversification and intensification. With regard to the dimension of expected costs,  $T_{stop}$  was set to 10. Hence, at the end of the algorithm almost exclusively cost improvements are accepted in order to ensure intensification. To establish a balance at each temperature level, 100 iterations were carried out per temperature level. This proved to be favorable to explore the search space. At lower values the search space is reduced too much and at higher values the algorithm starts to cycle.

## 5.2 Dual heuristic: problem-specific relaxation

A simple LP-relaxation, i.e. dropping the integrality constraints, is not suitable to obtain strong lower bounds. For that reason, an approach is presented which uses problem specific knowledge to meet this requirement, see Algorithm 1.

In the first step the original problem is relaxed by disabling the coupling constraints which connect the buy ( $b_{i,j}$ ) and the operation variables ( $a_{i,j}^s$ ) of the components for all load scenarios, i.e. only bought components can be used to satisfy the load scenarios:

$$a_{i,j}^s \leq b_{i,j} \quad (31)$$

Note that in the case of the booster stations considered in this paper the term ‘components’ corresponds to pumps. Afterwards, the problem is split into  $|S|$ -many subproblems, one for each load scenario. The remaining buy variables in all subproblems are substituted by the suitable operation variables. Afterwards, each of the  $|S|$  subproblems is split again into two sub-subproblems. The respective problems represent the optimization tasks for minimizing the energy costs and the investment costs for one single load scenario  $s$ . The new objective functions for the sub-subproblems are:

$$\min \left( C^{kWh} \cdot \sum_{(i,j) \in E} F^s \cdot p_{i,j}^s \cdot OLS \right) \quad (32)$$

$$\min \sum_{(i,j) \in E} C_{ij}^{buy} \cdot a_{ij}^s \quad (33)$$

For each of these  $2 \cdot |S|$  problems the optimal solution is determined by a MILP solver. A lower bound is composed of the sum of the energy costs and the maximum of all investment costs for each load scenario:

$$\underline{z} = \sum_{s \in S} \left( C^{kWh} \cdot \sum_{(i,j) \in E} F^s \cdot p_{ij}^s \cdot OLS \right) + \max_{s \in S} \left( \sum_{(i,j) \in E} C_{ij}^{buy} \cdot a_{ij}^s \right) \quad (34)$$

This is obviously a valid way to obtain lower bounds: The energy costs for one load cannot be lower than those which arise for the decoupled case because this is also the configuration with minimal costs for the original problem in the given load scenario. Therefore, the sum of these energy costs cannot be higher than in the original problem. Given the fact that the optimal system for the original problem must be able to operate in each load scenario, the investment costs cannot be lower than the maximum of the individually computed investment costs for each decoupled load scenario because this is the configuration with minimal costs to serve the ‘most challenging’ load scenario.

---

#### Algorithm 1 Dual heuristic

---

- 1: Let  $P_0$  be the original problem
  - 2: Let  $f$  be the objective function
  - 3: Disable coupling constraints for  $P_0$
  - 4: Split  $P_0$  into  $P_1, \dots, P_{|S|}$  (one for each load scenario  $s$ )
  - 5: Lower Bound  $LB \leftarrow 0$
  - 6: Energy-Costs  $EC \leftarrow 0$
  - 7: Invest-Costs  $IC \leftarrow 0$
  - 8: **for**  $k \in \{1 \dots |S|\}$  **do**
  - 9: Replace buy- with operation-variables for  $P_k$
  - 10: Generate topology problem  $T_k$  for  $P_k$
  - 11: Generate operation problem  $C_k$  for  $P_k$
  - 12:  $EC \leftarrow EC + f(T_k)$
  - 13: **if**  $IC < f(C_k)$  **then**
  - 14:  $IC \leftarrow f(C_k)$
  - 15: **end if**
  - 16: **end for**
  - 17:  $LB \leftarrow EC + IC$
- 

### 5.3 Closing the gap: Branch-and-Bound

Based on the basic Branch-and-Bound algorithm, as described in Clausen (1999), a framework using problem specific knowledge to obtain optimal solutions for the considered minimization problem is presented, see Algorithm 2. Branch-and-Bound belongs to the class of exact solution methods. It is a widespread method for solving large, combinatorial optimization problems. The complete enumeration of such problems is impractical because the number of possible solutions grows



exponentially with the problem size. However, the advantage of the Branch-and-Bound is that parts of the solution space can be pruned. For this a dynamically generated search tree is used.

Initially, this search tree only consists of one node, the root node which represents the whole search space of the original problem. Typically, a feasible solution for the root problem is calculated beforehand and becomes the initial best known solution. Otherwise, the best known solution value is set to  $+\infty$  if a minimization problem is considered. In this paper, we use the solution of Simulated Annealing as described in Sect. 5.1. Note that the best known solution value is used as a synonym for the global upper bound.

In each iteration of the algorithm an unexplored (active) node, representing a specific subproblem, is processed. An iteration contains three steps: selecting a node, dividing the solution space of this node into two smaller subspaces (branching) and calculating the bounds for the arising subproblems. The selection of a node follows a certain selection strategy. Here, we use the best-first-search selection strategy, for which always the node out of the set of active nodes with the lowest bound is selected. For these nodes one or more so-called conflicting components exist. These are components which are used for operation in the relaxation but their costs are not part of the investment costs of the relaxation. After the selection, branching is performed and two child nodes are generated by introducing additional constraints in order to divide the solution space. The branching rule for the active nodes is defined as follows: A component out of the set of conflicting components of this node is selected randomly. For one of the subproblems, an additional constraint is added which sets the binary buy-variable of the selected conflicting component to 0, i.e. the component is not part of the system. For the other subproblem, an additional constraint which sets the buy-variable to 1, i.e. the component is part of the system, is added instead. Hence, the search space is split into two smaller disjoint search spaces. Note that if a buy-variable is set to 0, the selected conflicting component is not bought and therefore cannot be used for operation. As a result, any solution with an operation-variable associated with this component not equal to 0 would be inherently infeasible for the original problem due to Constraint (20) of the MILP for fluid systems. Therefore, the operation-variables associated with these components in the respective subproblems are fixed to 0. In the opposite case, the operation-variables are not effected by such a restriction.

Afterwards, the bounds of the newly generated nodes are calculated immediately. This is called the eager evaluation strategy, whereas for the so-called lazy strategy, the bounds of the child nodes are not calculated until the respective node is selected and the nodes are selected according to the bound of their parent node. The bounds of the nodes are determined by solving the relaxation defined in Sect. 5.2 for the given subproblem. If the solution of the relaxation of a node is a valid solution for the original problem, its value is compared to the currently best known solution and the better solution is kept. In this implementation, a solution of the relaxation is valid for the original problem if only those components are used for operation which are also bought. This means that their purchase costs are part of the investment costs of the system according to the explanations given in Sect. 5.2. If the bound is worse than the best known solution, no further

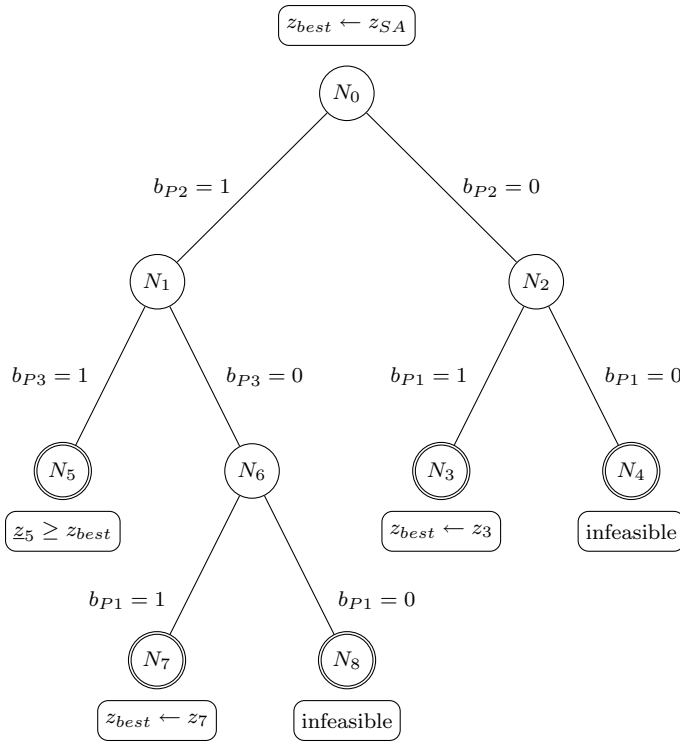


Fig. 7 Exemplary illustration of the implemented Branch-and-Bound method

exploration for this subtree is needed because the subproblem contains no better solutions for the original problem than the currently best known solution. The same applies if there are no feasible solutions for the subproblem. Otherwise, if none of these three cases occur, the node becomes part of the set of active nodes since the corresponding subproblem may still contain better solutions than the currently best known solution.

The search terminates if there are no active nodes left. The currently best known solution at this point is the provable optimal solution to the original problem since there are no subproblems left which could contain a better solution and the union of their disjoint search spaces equals the search space of the original problem.

An exemplary illustration for branching in the case of the application to booster stations is given in Fig. 7. The procedure starts from the root node  $N_0$  with initial best known solution  $z_{best}$  resulting from using the objective value of the solution produced by Simulated Annealing  $z_{SA}$ . Branching is performed on the buy-variables of the conflicting pumps as described above, here represented by  $b_{P_x}$ . The node indices indicate the sequence of the node creation. Furthermore, the example includes all three termination criteria: the solution of the relaxation is also a valid solution for

the original problem  $(N_3, N_7)$ , the subproblem is infeasible  $(N_4, N_8)$  or the bound obtained by the relaxation  $\underline{z}_x$  is worse than the currently best found solution  $(N_5)$ .

---

**Algorithm 2** Branch-and-Bound framework
 

---

```

1: Calculate global Upper Bound  $UB$ 
2: Let  $n_0$  be the original problem
3: Calculate Lower Bound of  $n_0$   $LB(n_0)$ 
4: Let  $RELAX(n_0)$  be the optimal solution for relaxed  $n_0$ 
5: ActiveNodes  $AN \leftarrow AN \cup \{n_0\}$ 
6: while  $AN \neq \emptyset$  do
7:   Take  $n_i \in AN$  with  $LB(n_i) \leq LB(n_j) \forall n_j \in AN$ 
8:   Take a component  $p_p$  from  $n_i$  which is used but not bought
9:   Split  $n_i$  into two child nodes  $cn_0, cn_1$ 
10:  Fix buy-variable of  $p_p$  to 0 for  $cn_0$  and to 1 for  $cn_1$ 
11:  for  $cn_k \in \{cn_0, cn_1\}$  do
12:     $AN \leftarrow AN \cup \{cn_k\}$ 
13:    Calculate  $LB(cn_k)$ 
14:    if  $LB(cn_k) \geq UB$  then
15:       $AN \leftarrow AN \setminus \{cn_k\}$ 
16:    else if  $RELAX(cn_k)$  is a valid solution for  $n_0$  then
17:       $AN \leftarrow AN \setminus \{cn_k\}$ 
18:       $UB \leftarrow LB(cn_k)$ 
19:    end if
20:  end for
21:   $AN \leftarrow AN \setminus \{n_i\}$ 
22: end while
23: Optimal Solution  $OS \leftarrow UB$ 

```

---

## 5.4 Application to booster stations

To validate the developed approach, test instances with a realistic character were designed. For this, the application example of so-called booster stations is used.

A booster station, also referred to as pressure booster system, is a network of either one type or different types of typically two to six single rotary pumps. A main field of application is the supply of whole buildings or higher floors, especially in high-rise buildings, with drinking water if the supply pressure provided by the water company is not high enough to satisfy the demand at all times. Typically, a distinction between three different system concepts is made. These concepts are booster stations with cascade control, with continuously variable speed control of one pump and with continuously variable speed control of all pumps. In this paper, we concentrate on the third concept, booster stations with continuously variable speed control of all pumps. For this concept, the number of active pumps as well as their speed depends on the required volume flow. Because of the continuously variable speed control of all pumps, a very constant inlet pressure occurs and it is possible

to compensate high supply pressure fluctuations even if a malfunction occurs or a pump is failing. There is no sudden pressure increase because the other pumps can step in. Furthermore, we focus on a connection concept in which the booster station is connected to the water supply directly and no discharge sided pressure vessels are used. If necessary, so-called normal zones are implemented. These can be served by the supply pressure itself and are therefore not connected to the booster station. This can be used to avoid overpressure for lower floors. For all other floors overpressure is avoided by installing reducing valves if necessary.

A booster station system primarily consists of four types of components: pumps, pipes, pressure reducers and valves. Furthermore, each system has at least one source and one sink. In this paper, we focus on the pumps of booster stations and consider the other components implicitly. Hence, the presentation is simplified to a switchable interconnection of pumps which form a connected network. The relevant physical variables are: the volume flow  $\dot{Q}$  through the pumps, the pressure head  $\Delta H$  generated by the pumps, their power consumption  $P$  and their rotational speed  $n$ .

All calculations are based on DIN 1988-300 (2012) and DIN 1988-500 (2011). Furthermore, the planning horizon was set to 10 years with assumed mean energy costs of 0.30 Euro per kWh.

To generate the test instances, different characteristics were varied and combined:

- the height and usable area of the buildings
- the intended use of the building with the corresponding load profile
- the conditioning of hot water
- the available pump kit

This results in 24 different instances. The names of the instances are derived from the abbreviations for the respective characteristics. In the following, these characteristics are specified.

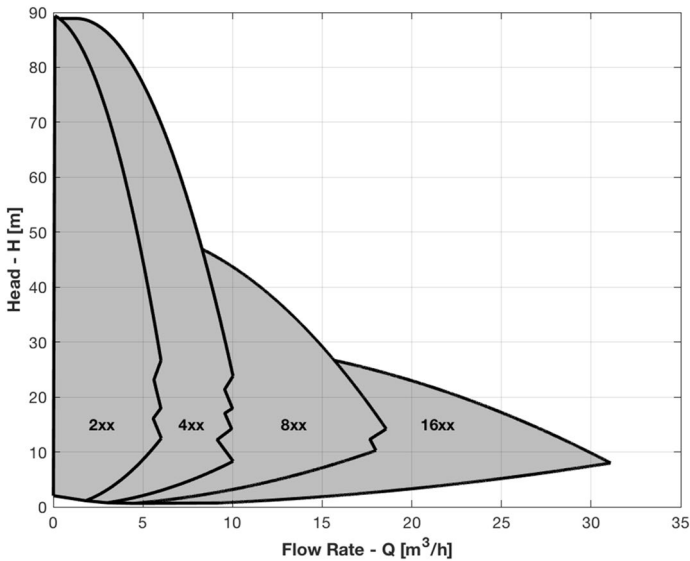
*Buildings:* Two different fictional buildings are used. Both are high-rise buildings but vary in two characteristics: The first building (*B15*) is 15 floors high and each floor has a usable area of 350 sq. m. The second building (*B10*) is 10 floors high and has a usable area of 700 sq. m for each floor. This means that different pressure increases and maximum volume flows are required as the building's height and usable area effect the pressure losses and demanded volume flows.

*Intended use:* The buildings are either used as a hypothetical hospital (*H*), a residential (*R*) or an office building (*O*). All usage types differ regarding their furnishing and consumption behavior. Hence, different maximum volume flows, pressure losses and load profiles occur. Depending on the usage four or five load scenarios are distinguished.

*Hot water conditioning:* The conditioning of hot water either occurs in so-called centralized storage water heaters (*C*) or decentralized group water heaters (*D*). These concepts result in different pressure losses along the piping.

*Available pump kit:* For each test instance one of two disjoint pump kits with five pumps each is available. All of them are speed-controlled single rotary pumps and taken from the Wilo-Economy MHIE<sup>5</sup> model series, see Fig. 8. The first kit includes

<sup>5</sup> [https://wilo.com/ie/en/Products-and-expertise/Series-Finder/Wilo-Economy-MHIE\\_110.html](https://wilo.com/ie/en/Products-and-expertise/Series-Finder/Wilo-Economy-MHIE_110.html) (accessed September 20, 2019).



**Fig. 8** Schematic representation of the characteristic maps of the used pumps

**Table 2** Load data for the test instances

| Test instance | $\dot{V}^{max}$ [m <sup>3</sup> /h] | $\Delta H$ [m] |
|---------------|-------------------------------------|----------------|
| B10_O_D_1/2   | 6.48                                | 27.5           |
| B10_O_C_1/2   | 4.86                                | 19.5           |
| B15_O_D_1/2   | 6.45                                | 41.4           |
| B15_O_C_1/2   | 5.37                                | 33.4           |
| B10_R_D_1/2   | 9.85                                | 32.1           |
| B10_R_C_1/2   | 6.90                                | 24.1           |
| B15_R_D_1/2   | 7.92                                | 45.0           |
| B15_R_C_1/2   | 6.70                                | 37.0           |
| B10_H_D_1/2   | 14.53                               | 31.8           |
| B10_H_C_1/2   | 9.68                                | 23.8           |
| B15_H_D_1/2   | 10.49                               | 48.4           |
| B15_H_C_1/2   | 8.87                                | 40.4           |

the types from 203 to 403 of the model series (1) and the second kit the types from 404 to 1602 (2) with different prices and characteristics.

As a summary, Table 2 shows the peak loads for the different test instances in terms of the maximum volume flow  $\dot{V}^{max}$  and necessary pressure head  $\Delta H$ . Note that there are always two test instances for each of the 12 entries since they are used with two different pump kits. For the partial loads, which depend on the considered building type, Table 3 shows the different scenarios with the relative

**Table 3** Time shares and relative volume flows of the load scenarios

| Building    | Scenario | $F$  | $\dot{V}/\dot{V}^{max}$ |
|-------------|----------|------|-------------------------|
| Office      | 1        | 0.03 | 1.00                    |
|             | 2        | 0.12 | 0.40                    |
|             | 3        | 0.30 | 0.30                    |
|             | 4        | 0.55 | 0.00                    |
| Residential | 1        | 0.17 | 1.00                    |
|             | 2        | 0.17 | 0.70                    |
|             | 3        | 0.25 | 0.40                    |
|             | 4        | 0.21 | 0.35                    |
|             | 5        | 0.20 | 0.15                    |
| Hospital    | 1        | 0.08 | 1.00                    |
|             | 2        | 0.25 | 0.50                    |
|             | 3        | 0.25 | 0.25                    |
|             | 4        | 0.17 | 0.10                    |
|             | 5        | 0.25 | 0.05                    |

time shares  $F$  of the operational lifespan for which these scenarios are expected to occur and the associated relative volumes flows  $\dot{V}/\dot{V}^{max}$ .

## 5.5 Computational study

In order to validate the developed approach, we conducted a computational study using the 24 test instances introduced in Sect. 5.4. All calculations were performed on a MacBook Pro Early 2015 with a 2.7 GHz Intel Core i5 and 8 GB 1867 MHz DDR3 memory, using CPLEX Optimization Studio 12.6 as MILP solver.

### 5.5.1 Solutions

In this section, the quality of the solutions found by the primal and dual heuristics is presented.

*Simulated Annealing:* Table 4 shows a summary of the performance for the presented implementation of Simulated Annealing in all test instances. The best solution found by Simulated Annealing is represented by  $z_{SA}$ . The lower bound  $\underline{z}$  is calculated using the dual heuristic and the optimal solution  $z^*$  is obtained via Branch-and-Bound. The relative gap between the solution of Simulated Annealing and the lower bound  $gap_{\underline{z}}$  is defined as  $(z_{SA} - \underline{z})/\underline{z}$ . The relative gap between the best solution obtained by Simulated Annealing and the actual optimal solution  $gap_{z^*}$  is defined as  $(z_{SA} - z^*)/z^*$ .

The mean value of  $gap_{\underline{z}}$  for all test instances was 9.27% with a standard deviation of 6.37%. In 14 out of 24 cases the optimal solution was found by the implemented Simulated Annealing algorithm. The mean value for  $gap_{z^*}$  was 0.69% with

**Table 4** Solution statistics for Simulated Annealing with regard to the lower bounds and optimal solutions

| Test instance | $z_{SA}$  | $\underline{z}$ | $gap_{\underline{z}} [\%]$ | $z^*$     | $gap_{z^*} [\%]$ |
|---------------|-----------|-----------------|----------------------------|-----------|------------------|
| B10_O_D_1     | 6007.54   | 5962.32         | 0.76                       | 6007.54   | 0.00             |
| B10_O_D_2     | 6492.46   | 6026.12         | 7.74                       | 6492.46   | 0.00             |
| B10_O_C_1     | 4370.36   | 4024.15         | 8.60                       | 4370.36   | 0.00             |
| B10_O_C_2     | 4712.02   | 4224.70         | 11.54                      | 4712.02   | 0.00             |
| B15_O_D_1     | 10,069.90 | 10,015.90       | 0.54                       | 10,069.90 | 0.00             |
| B15_O_D_2     | 9115.01   | 8116.26         | 12.31                      | 9115.01   | 0.00             |
| B15_O_C_1     | 6571.15   | 6162.81         | 6.63                       | 6571.15   | 0.00             |
| B15_O_C_2     | 7002.53   | 6288.44         | 11.36                      | 7002.53   | 0.00             |
| B10_R_D_1     | 24,601.00 | 24,004.30       | 2.49                       | 24,518.10 | 0.34             |
| B10_R_D_2     | 23,516.20 | 22,215.40       | 5.86                       | 23,516.20 | 0.00             |
| B10_R_C_1     | 12,711.90 | 12,334.20       | 3.06                       | 12,711.90 | 0.00             |
| B10_R_C_2     | 13,968.60 | 12,157.00       | 14.90                      | 13,968.60 | 0.00             |
| B15_R_D_1     | 29,360.00 | 27,570.20       | 6.49                       | 29,319.40 | 0.14             |
| B15_R_D_2     | 28,407.00 | 24,457.10       | 16.15                      | 28,407.00 | 0.00             |
| B15_R_C_1     | 20,505.40 | 19,750.90       | 3.82                       | 20,486.40 | 0.09             |
| B15_R_C_2     | 19,909.10 | 17,315.50       | 14.98                      | 19,909.10 | 0.00             |
| B10_H_D_1     | 25,068.10 | 23,912.40       | 4.83                       | 24,607.60 | 1.87             |
| B10_H_D_2     | 23,704.70 | 22,127.50       | 7.13                       | 23,287.70 | 1.79             |
| B10_H_C_1     | 13,315.10 | 12,659.00       | 5.18                       | 13,070.80 | 1.87             |
| B10_H_C_2     | 13,946.80 | 11,168.80       | 24.87                      | 13,946.80 | 0.00             |
| B15_H_D_1     | 27,936.80 | 26,651.30       | 4.82                       | 27,210.70 | 2.67             |
| B15_H_D_2     | 28,186.40 | 25,001.00       | 12.74                      | 27,377.30 | 2.96             |
| B15_H_C_1     | 21,380.80 | 18,942.90       | 12.87                      | 20,974.60 | 1.94             |
| B15_H_C_2     | 21,649.00 | 17,637.20       | 22.75                      | 21,041.10 | 2.89             |

a standard deviation of 1.08%. However, if the optimal solution was not found, the mean value of  $gap_{z^*}$  was 1.65% with a standard deviation of 1.1%.

*Lower bounds:* Furthermore, the lower bounds were compared to the optimal solution. Table 5 summarizes the results for all test instances. Again,  $\underline{z}$  is the dual bound and  $z^*$  is the optimal solution obtained by Branch-and-Bound. The relative gap between the initial dual bound and the optimal solution, denoted by  $gap$ , is defined as  $(z^* - \underline{z})/z^*$ . The mean value of  $gap$  was 7.45% with a standard deviation of 5.23%. The maximum of  $gap$  was 19.92%, while the minimum was only 0.54%.

### 5.5.2 Runtime

In this section, the runtimes of all three procedures are presented. It should be noted that the runtime of the Branch-and-Bound framework includes the runtime of Simulated Annealing as it generates the starting solution for the procedure. An overview of the runtimes for all test instances for Simulated Annealing (SA), the dual

**Table 5** Solution statistics for the dual heuristic with regard to the optimal solutions

| Test instance | $\underline{z}$ | $z^*$     | gap [%] |
|---------------|-----------------|-----------|---------|
| B10_O_D_1     | 5962.32         | 6007.54   | 0.75    |
| B10_O_D_2     | 6026.12         | 6492.46   | 7.18    |
| B10_O_C_1     | 4024.15         | 4370.36   | 7.92    |
| B10_O_C_2     | 4224.70         | 4712.02   | 10.34   |
| B15_O_D_1     | 10,015.90       | 10,069.90 | 0.54    |
| B15_O_D_2     | 8116.26         | 9115.01   | 10.96   |
| B15_O_C_1     | 6162.81         | 6571.15   | 6.21    |
| B15_O_C_2     | 6288.44         | 7002.53   | 10.20   |
| B10_R_D_1     | 24,004.30       | 24,518.10 | 2.10    |
| B10_R_D_2     | 22,215.40       | 23,516.20 | 5.53    |
| B10_R_C_1     | 12,334.20       | 12,711.90 | 2.97    |
| B10_R_C_2     | 12,157.00       | 13,968.60 | 12.97   |
| B15_R_D_1     | 27,570.20       | 29,319.40 | 5.97    |
| B15_R_D_2     | 24,457.10       | 28,407.00 | 13.90   |
| B15_R_C_1     | 19,750.90       | 20,486.40 | 3.59    |
| B15_R_C_2     | 17,315.50       | 19,909.10 | 13.03   |
| B10_H_D_1     | 23,912.40       | 24,607.60 | 2.83    |
| B10_H_D_2     | 22,127.50       | 23,287.70 | 4.98    |
| B10_H_C_1     | 12,659.00       | 13,070.80 | 3.15    |
| B10_H_C_2     | 11,168.80       | 13,946.80 | 19.92   |
| B15_H_D_1     | 26,651.30       | 27,210.70 | 2.06    |
| B15_H_D_2     | 25,001.00       | 27,377.30 | 8.68    |
| B15_H_C_1     | 18,942.90       | 20,974.60 | 9.69    |
| B15_H_C_2     | 17,637.20       | 21,041.10 | 16.18   |

heuristic to generate initial lower bounds (LB) and Branch-and-Bound (B&B) is given in Table 6.

*Simulated Annealing:* The Simulated Annealing algorithm took on average 475 s to terminate. High deviations occurred. The maximum runtime was 2412 s, while the minimum runtime was only 85 s. This results from the fact that the MILP solver needs much more time to solve the operation problem if the created neighborhood is large in terms of many bought components.

*Lower bounds:* Generating lower bounds took on average 661 s. The longest runtime was 1582 s, while the shortest runtime was only 208 s. Note that in most cases this was comparable to the time the Simulated Annealing algorithm took to terminate. Hence, this circumstance allows a timely examination of a solution found by Simulated Annealing in practice.

*Branch-and-Bound:* The average runtime for generating optimal solutions was 9969 s. The maximum runtime was 21,473 s and the minimum runtime only 4148 s. If the initial upper bound found by Simulated Annealing was already the optimal solution, the



**Table 6** Runtime statistics (in seconds)

| Test instance | SA   | LB   | B&B    |
|---------------|------|------|--------|
| B10_O_D_1     | 140  | 398  | 4260   |
| B10_O_D_2     | 143  | 260  | 4996   |
| B10_O_C_1     | 102  | 372  | 7313   |
| B10_O_C_2     | 85   | 208  | 4148   |
| B15_O_D_1     | 437  | 811  | 7542   |
| B15_O_D_2     | 224  | 310  | 4848   |
| B15_O_C_1     | 353  | 612  | 6484   |
| B15_O_C_2     | 188  | 269  | 5249   |
| B10_R_D_1     | 551  | 1095 | 10,518 |
| B10_R_D_2     | 652  | 1582 | 12,522 |
| B10_R_C_1     | 329  | 828  | 8382   |
| B10_R_C_2     | 333  | 687  | 21,473 |
| B15_R_D_1     | 1054 | 1010 | 11,827 |
| B15_R_D_2     | 290  | 501  | 13,721 |
| B15_R_C_1     | 399  | 767  | 10,177 |
| B15_R_C_2     | 231  | 426  | 12,361 |
| B10_H_D_1     | 439  | 801  | 5948   |
| B10_H_D_2     | 417  | 742  | 13,615 |
| B10_H_C_1     | 275  | 653  | 13,557 |
| B10_H_C_2     | 109  | 574  | 9956   |
| B15_H_D_1     | 2412 | 709  | 6273   |
| B15_H_D_2     | 660  | 707  | 13,243 |
| B15_H_C_1     | 1162 | 957  | 18,796 |
| B15_H_C_2     | 424  | 583  | 12,039 |

average runtime was 8804 s and therefore 31.75% faster than in the opposite case where the average runtime was 11,599 s.

## 6 Resilient system design (*Tile 2.1*)

As an extension to the approach presented above one can enhance the resilience of technical systems by adding possible breakdown scenarios. The concept of resilience is of great interest since it cannot only be applied to control uncertainty during the design phase, but it is also applicable for the system's operation. Instead of designing systems that are robust with respect to specific single 'what-if' assumptions made beforehand during the design phase, resilient system design aims at building systems that perform 'no matter what' (Altherr et al. 2018a). In this context, resilience of a technical system is the ability to overcome minor failures and thus to avoid a complete breakdown of its vital functions. A possible failure of the system's components is one critical case the system designer should keep in mind.

In this context, optimization under uncertainty can be used in order to describe and increase resilience of technical systems (Altherr et al. 2018a). Prominent solution paradigms for optimization under uncertainty are, inter alia, Stochastic Programming (Birge and Louveaux 2011), Robust Optimization (Ben-Tal et al. 2009), Dynamic Programming (Bellman 2003) and Sampling (Gupta et al. 2004). In the early 2000s the idea of universally quantified variables, as they are used in Quantified Constraint Satisfaction Problems (Gerber et al. 1995), was picked up again (Subramani 2004)—coining the term Quantified Integer Program (QIP)—and further examined (Ederer et al. 2011; Lorenz et al. 2010). QIP gives the opportunity to combine traditional Linear Programming formulations with some uncertainty bits. Hence, a solution of a QIP is a strategy for assigning existentially quantified variables such that some linear constraint system is fulfilled. By adding a minmax objective function one must further find the best strategy (Ederer et al. 2011).

In this spirit, for our contribution to the design of more resilient technical systems, we consider the following special case: Starting from a valid network configuration  $(G, S_G, T_G)$  that is able to satisfy the desired loads of any scenario  $i \in S$ , we are allowed to add some additional components to make the system more resilient against breakdowns.

More concrete, we define  $I := E$  as the set of initial components,  $A$  as the set of additional components and try to find a subset  $A' \subseteq A$  such that  $G' := ((V, I \cup A'), S_G, T_G)$  fulfills resilience in the following sense: For each scenario  $i \in S$  it has to be ensured that if a single component  $e \in I$  is affected by a breakdown, a valid combination in  $G'' := ((V, (I \cup A') \setminus \{e\}), S_G, T_G)$  must exist such that the demanded load in scenario  $i$  can always be satisfied.

The set of additionally bought components  $A'$  must be selected such that the lifetime costs of the resulting system, i.e. investment costs and operational costs, are minimal. Hence, a multistage optimization problem arises: Design or adapt the system (stage 1) such that for each anticipated load scenario (stage 2) we can find the optimal operation point (stage 3) and ensure for each breakdown case (stage 4) that the functionality of the system is ensured (stage 5).

Since the system design process can be conducted in several consecutive steps the arising problem is a multistage optimization problem. Hence, we make use of a Quantified Mixed-Integer Linear Program (QMIP) to find optimal system configurations with increased resilience.

It should be noted that although applied to booster stations in this paper, the approach can be abstracted for a variety of technical system using the general representation of so-called process networks as shown in Hartisch et al. (2018). Furthermore, similar to the concept of  $K$ -resilience examined in Altherr et al. (2019), the simultaneous breakdown of multiple pumps can be considered if necessary.

## 6.1 Quantified Programming

Quantified Mixed-Integer Linear Programming is a direct and formal extension to Mixed-Integer Linear Programming utilizing uncertainty bits. In QMIPs the variables

are ordered explicitly and they are quantified either existentially or universally resulting in a multistage optimization problem under uncertainty:

Let there be a vector of  $n$  variables  $x = (x_1, \dots, x_n)^\top \in \mathbb{Q}^n$ , lower and upper bounds  $l \in \mathbb{Q}^n$  and  $u \in \mathbb{Q}^n$  with  $l_i \leq x_i \leq u_i$ , a coefficient matrix  $A \in \mathbb{Q}^{m \times n}$ , a right-hand side vector  $b \in \mathbb{Q}^m$  and a vector of quantifiers  $Q = (Q_1, \dots, Q_n)^\top \in \{\forall, \exists\}^n$ . Let  $I \subset \{1, \dots, n\}$  be the set of integer variables and  $\mathcal{L}_i = \{x \in \mathbb{Q} \mid (l_i \leq x \leq u_i) \wedge (i \in I \Rightarrow x \in \mathbb{Z})\}$  the domain of variable  $x_i$  and let  $\mathcal{L} = \{x \in \mathbb{Q}^n \mid x_i \in \mathcal{L}_i\}$  be the domain of the entire variable vector. Let the term  $Q \circ x \in \mathcal{L}$  with the component wise binding operator  $\circ$  denote the quantification vector  $(Q_1 x_1 \in \mathcal{L}_1, \dots, Q_n x_n \in \mathcal{L}_n)^\top$  such that every quantifier  $Q_i$  binds the variable  $x_i$  to its domain  $\mathcal{L}_i$ . We call  $(Q, l, u, A, b)$  with

$$\begin{aligned} z &= \min c^\top x \\ \text{s.t. } Q \circ x &\in \mathcal{L} : Ax \leq b \end{aligned}$$

a Quantified Mixed-Integer Linear Program (QMIP).

Note that the objective function is actually a minmax function alternating according to the quantifier sequence: Existential variables are set with the goal of minimizing the objective value while obeying the constraint system, whereas universal variables are aiming at a maximized objective value. For more details, we refer to Wolf (2015). QMIPs allow a straightforward modeling of multistage optimization problems and the domain of universal variables might be modeled explicitly using a second linear constraint system (Hartisch et al. 2016).

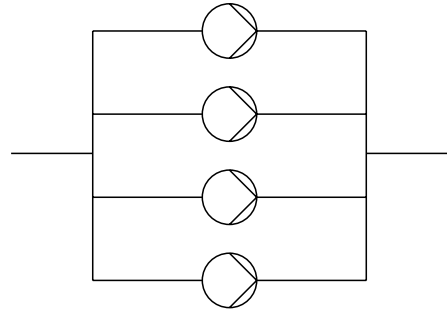
Solutions of QMIPs are strategies for assigning existentially quantified variables such that the linear constraint system  $Ax \leq b$  is fulfilled. One way to deal with quantified programs is to build the corresponding deterministic equivalent program (DEP) (Wolf 2015; Wets 1974) and to solve the resulting MILP using standard MILP solvers. Further, a novel open-source solver for QMIPs is available performing an enhanced game tree search (Ederer et al. 2017).

## 6.2 Application to booster stations

In order to build on the results of Sect. 5, the application example of generating cost-efficient resilient booster stations out of non-resilient ones is examined here. The requirements for the considered case of resilient booster stations are manifested in DIN 1988-500 (2011). It states that booster stations must have at least one stand-by pump. If one pump breaks down, the system must be able to satisfy the peak flow and thus all demanded loads at any time. In contrast to related contributions (cf. Altherr et al. 2019), a further requirement mentioned in DIN 1988-500 is considered. This requirement states that in order to avoid stagnation water, an automatic, cyclic interchange between all pumps including the stand-by pumps is necessary. Therefore, all pumps have to operate at least once in 24 h. This additional requirement is strongly connected to the cost-efficiency goal.

In this example the relevant costs for a booster station are the investment costs for the stand-by pumps as well as the operational costs of the overall system over

**Fig. 9** Booster station with exclusively parallel pumps



a predefined lifespan. As the breakdown cases are expected to only take place in a small amount of time compared to the lifespan, due to short repair times, they do not significantly affect the operational costs of the system and are therefore neglected. However, the requirement for all pumps to operate once in 24 h, i.e. in at least one of the daily repeating load scenarios, massively affects the operational costs. Given this circumstance, it is not a trivial task to determine by which stand-by pumps the system should be extended in order to obtain a cost-optimal system.

Theoretically, a set of pumps or entire subsystems can be connected either in parallel or in series. However, according to today's practice only parallel connections are favorable from a technical point of view (Betz 2017). As mentioned in Pöttgen and Pelz (2016) two major reasons exist for considering parallel arrangements: Firstly, heavy part loads caused by the deactivation of single pumps are avoided. Secondly, in case of failure of a single pump the remaining system components are not directly affected and retain their full functionality. Although serial arrangements are generally conceivable, the resulting control strategies between two operating points are very difficult to realize in practice. We make use of this circumstance to obtain significantly smaller pump networks by using only parallel connections to demonstrate the approach. Figure 9 shows such a network with four parallel pumps.

### 6.3 Optimization model

The quantified optimization model consists of five stages corresponding to variable blocks in the QMIP. The first existential block primarily represents the investment decision concerning the additional pumps. In the universal second variable block the load scenario is selected. The existential third variable block is used to determine the cost-optimal operating point of the available pumps for the given scenario. In the subsequent universal variable block one of the initial pumps is chosen for breakdown. The final existential block is used to check whether the remaining pumps (excluding the broken one) are able to fulfill the selected load scenario.

As the handling of the breakdown- and standard-control is independent—and only depends on the first stage investment decision—we could also have built a three-stage model: investment decision (first stage), selection of a load and a breakdown scenario (second stage), and finally computing the standard- and breakdown-control (third stage). However, using five stages has severe

**Table 7** Variables and parameters of the QMIP

| Symbol                        | Range            | Stage          | Description   |
|-------------------------------|------------------|----------------|---|
| $b_p$                         | $\{0, 1\}$       | 1( $\exists$ ) | Purchase decision for pump $p \in A$                            |
| $a_p^i$                       | $\{0, 1\}$       | 1( $\exists$ ) | Usage decision for pump $p \in P$ in scenario $i \in S$         |
| $c^i$                         | $\mathbb{R}_0^+$ | 1( $\exists$ ) | Operational costs in scenario $i \in S$                         |
| $s$                           | $S$              | 2( $\forall$ ) | Scenario selection  |
| $\sigma^i$                    | $\{0, 1\}$       | 3( $\exists$ ) | Indicator whether scenario $i \in S$ was selected               |
| $x_p$                         | $\{0, 1\}$       | 3( $\exists$ ) | Indicator whether pump $p \in P$ is used                        |
| $\dot{v}_p$                   | $\mathbb{R}_0^+$ | 3( $\exists$ ) | Volume flow through pump $p \in P$                              |
| $\Delta h_p$                  | $\mathbb{R}_0^+$ | 3( $\exists$ ) | Pressure increase by pump $p \in P$                             |
| $\rho_p$                      | $\mathbb{R}_0^+$ | 3( $\exists$ ) | Power consumption of pump $p \in P$                             |
| $n_p$                         | $[0, 1]$         | 3( $\exists$ ) | Rotational speed of pump $p \in P$                              |
| $d$                           | $I$              | 4( $\forall$ ) | Selection of damaged pump                                       |
| $\delta_p$                    | $\{0, 1\}$       | 5( $\exists$ ) | Indicator whether initial pump $p \in I$ is damaged             |
| $x_p^B$                       | $\{0, 1\}$       | 5( $\exists$ ) | Indicator whether pump $p \in P$ is used in case of disturbance |
| $\dot{v}_p^B$                 | $\mathbb{R}_0^+$ | 5( $\exists$ ) | Volume flow through pump $p \in P$ in case of disturbance       |
| $\Delta h_p^B$                | $\mathbb{R}_0^+$ | 5( $\exists$ ) | Pressure increase by pump $p \in P$ in case of disturbance      |
| $I = \{1, \dots, n\}$         | –                | –              | Set of initial pumps  |
| $A = \{n + 1, \dots, n + m\}$ | –                | –              | Set of additional pumps   |
| $P = A \cup I$                | –                | –              | Set of all available pumps                                      |
| $S = \{1, \dots, \bar{S}\}$   | –                | –              | Set of scenarios  |
| $C_p^{buy}$                   | –                | –              | Investment costs for pump $p \in A$                             |
| $\dot{V}^i$                   | –                | –              | Demanded volume flow in scenario $i \in S$                      |
| $\Delta H^i$                  | –                | –              | Demanded pressure increase in scenario $i \in S$                |
| $F^i$                         | –                | –              | Relative frequency of scenario $i \in S$                        |
| $OLS$                         | –                | –              | Operational lifespan of the system                              |
| $C^{kWh}$                     | –                | –              | Costs per kilowatt hour of electricity                          |
| $\Delta H^{\max}$             | –                | –              | General upper bound for pressure increase                       |
| $\dot{V}^{\max}$              | –                | –              | General upper bound for volume flow                             |
| $M$                           | –                | –              | Sufficiently large constant                                     |

advantages. Firstly, the chosen variable sequence indicates the processing order more accurately: For any scenario, we must provide a standard-control first and subsequently valid breakdown-controls must be ensured for the particular scenario. Secondly, the DEP contains significantly less variables since the standard-control decision variables do not have to be duplicated for each breakdown scenario (Wolf 2015). A similar argument is valid for game tree search methods: If modeled as a three-stage QMIP, the standard-control found for one breakdown scenario must be rediscovered for another breakdown scenario, even though it could simply stay the same.

Table 7 displays the parameters and variables used for the QMIP. For the sake of compact presentation, we do not explicitly state the quantification vector  $\mathcal{Q} \circ x$ .

However, in Table 7 both the stage and thus the variable order as well as the variable quantification is given.

$$\min \sum_{i \in S} F^i \cdot c^i + \sum_{p \in A} C_p^{buy} \cdot b_p \quad (35)$$

$$x_p \leq b_p, x_p^B \leq b_p \quad \forall p \in A \quad (36)$$

$$a_p^i - x_p + \sigma^i \leq 1 \quad \forall p \in P, i \in S \quad (37)$$

$$\sum_{i \in S} a_p^i \geq 1 \quad \forall p \in I \quad (38)$$

$$\sum_{i \in S} a_p^i \geq b_p \quad \forall p \in A \quad (39)$$

$$\sum_{i \in S} \sigma^i = 1 \quad (40)$$

$$\sum_{i \in S} i \cdot \sigma^i = s \quad (41)$$

$$\sum_{p \in I} \delta_p = 1 \quad (42)$$

$$\sum_{p \in I} p \cdot \delta_p = d \quad (43)$$

$$x_p^B + \delta_p \leq 1 \quad \forall p \in I \quad (44)$$

$$\rho_p = \mathcal{P}_p(\dot{v}_p, n_p) \quad \forall p \in P \quad (45)$$

$$\Delta h_p = \Delta H_p(\dot{v}_p, n_p) \quad \forall p \in P \quad (46)$$

$$\Delta h_p^B = \Delta H_p^B(\dot{v}_p^B) \quad \forall p \in P \quad (47)$$

$$\Delta h_p = x_p \cdot \sum_{i \in S} \Delta H^i \cdot \sigma^i \quad \forall p \in P \quad (48)$$

$$\Delta h_p^B = x_p^B \cdot \sum_{i \in S} \Delta H^i \cdot \sigma^i \quad \forall p \in P \quad (49)$$

$$\sum_{p \in P} \dot{v}_p = \sum_{i \in S} \dot{V}^i \cdot \sigma^i \quad (50)$$

$$\sum_{p \in P} \dot{v}_p^B = \sum_{i \in S} \dot{V}^i \cdot \sigma^i \quad (51)$$

$$\dot{v}_p \leq \dot{V}^{\max} \cdot x_p \quad \forall p \in P \quad (52)$$

$$\Delta h_p \leq \Delta H^{\max} \cdot x_p \quad \forall p \in P \quad (53)$$

$$\dot{v}_p^B \leq \dot{V}^{\max} \cdot x_p^B \quad \forall p \in P \quad (54)$$

$$\Delta h_p^B \leq \Delta H^{\max} \cdot x_p^B \quad \forall p \in P \quad (55)$$

$$M \cdot (1 - \sigma^i) + c^i \geq C^{kWh} \cdot OLS \cdot \sum_{p \in P} \rho_p \quad \forall i \in S \quad (56)$$

The objective function (35) aims at minimizing the weighted operational costs in the scenarios as well as the costs resulting from buying additional pumps. Constraint (36) links the first and third variable block as well as the first and fifth variable block by demanding that only purchased pumps can be used. The feature that each pump must be used in at least one of the considered load scenarios is guaranteed by Constraints (37), (38) and (39). Constraints (40)–(43) link the universal variable decision of the selected scenario and the selected broken pump with the corresponding existential variables, while Constraint (44) prohibits the use of a broken pump. The operating point of a used pump must lie on its characteristic curve which describes the non-linear relation between  $\Delta h_p$ ,  $\dot{v}_p$ ,  $n_p$  and  $\rho_p$ . This coherence is outlined in Constraints (45) and (46) and is again modeled using the linearization technique presented in Vielma et al. (2010). As the power consumption of the booster station in the case of a breakdown is not subject of the optimization the non-linear relation between  $\Delta h_p^B$  and  $\dot{v}_p^B$  can be modeled more easily: Constraint (47) ensures that the selected operating point  $(\Delta h_p^B, \dot{v}_p^B)$  lies somewhere within the characteristic map without specifying the speed of the pump. Hence, linearizing the boundaries of the map and checking their fulfillment suffices. Constraints (48)–(51) ensure that the demanded volume flow and pressure increase of the selected load scenario are fulfilled for both the standard-control and the breakdown-control. Note that resolving the non-linearity in Constraints (48) and (49) is a trivial task by using a big-M

**Table 8** Load scenarios for the test instances

| Scenario | Example 1 |                               |                | Example 2 |                               |                |
|----------|-----------|-------------------------------|----------------|-----------|-------------------------------|----------------|
|          | $F$       | $\dot{V}$ [m <sup>3</sup> /h] | $\Delta H$ [m] | $F$       | $\dot{V}$ [m <sup>3</sup> /h] | $\Delta H$ [m] |
| 1        | 0.1       | 25                            | 20             | 0.25      | 20                            | 30             |
| 2        | 0.3       | 20                            | 30             | 0.25      | 12                            | 50             |
| 3        | 0.4       | 15                            | 40             | 0.25      | 10                            | 20             |
| 4        | 0.2       | 10                            | 60             | 0.25      | 5                             | 15             |

formulation. Constraints (52)–(55) set bounds on the volume flow and the pressure increase of a used pump and deal with unused pumps in particular. In Constraint (56) the power consumption resulting from the selected standard-control is transformed into energy costs. Note that the universal integer variables  $s$  and  $d$  and the existential binary variables  $\sigma$  and  $\delta$  are very similar and closely linked through Constraints (40)–(43). One might suggest that the binary variables  $\sigma$  and  $\delta$  could just as well be universal variables and thus replacing  $s$  and  $d$ . However, exactly one load and one breakdown scenario each must be selected. This would lead to a restriction of these variables as it is done in Constraints (40) and (42) but restricting universal variables using linear constraints (instead of simple variable bounds) requires further actions and a certain overhead (Hartisch et al. 2016; Hartisch and Lorenz 2019).

## 6.4 Computational study

In order to demonstrate the impact of this approach, we investigate two artificial examples. As for the pumps, the Wilo-Economy MHIE model series as already introduced in Sect. 5 is used. However, the single pump in group 16xx is neglected hereinafter due to its superiority compared to the other pumps in the considered examples. A suitable number of data points was extracted from the pumps' data-sheets in order to approximate the characteristic maps.

The two created QMIP instances are solved using the framework provided by the QMIP solver Yasol<sup>6</sup>. As the game tree search itself can only deal with continuous variables in the final variable block, we use the option of creating and solving the corresponding DEP. Since the runtimes of the inspected instances were in the range of seconds, we will not deepen this subject any further.

*Test instance 1:* As a first example, we investigate a system which is already optimized regarding the sum of investment and operational costs over a predefined set of load scenarios for the non-resilient case, shown on the left-hand side of Table 8. This system consists of one pump each of the types 206, 403, 406 and 803 connected in parallel and has initial operational costs of 75,288.88 Euro assuming a lifetime of ten years. In order to transform this given (functional) booster station into a more resilient one, we apply the presented optimization model. The set of selectable pumps  $A$  contains each pump of the Wilo-Economy MHIE series once. According to the solution of the QMIP, it is optimal to add the additional pump 205 with investment costs of 1805 Euro to the system. This might seem somewhat surprising at the

<sup>6</sup> <http://q-mip.org> (accessed September 20, 2019).



first glance given that even though the system was optimized for the non-resilient case none of the already installed pumps is doubled and instead a new type is added to the network in order to compensate for the breakdown of one of the initial pumps. This shows that even for an optimized system finding a more resilient configuration is a non-trivial task. Compared to the original system the selected additional pump is operational in the first scenario which results in an increase in the lifetime operational costs of only 3.52 Euro compared to the non-resilient case. Summing up, the minimal additional costs to make the initial booster station resilient are 1808.52 Euro.

*Test instance 2:* As a second example, we consider the case of an initial system with multiple identical pumps connected in parallel following the conventional design approach. The obvious way to achieve the addressed sense of resilience for such a system is to add another pump of the same type to the network. However, cheaper configurations might exist. For this example, we investigate such a system with three pumps of the 406-type. The corresponding load scenarios can be found on the right-hand side of Table 8 and the system is projected to be operational for five years. Again, as in the previous example we want to transform the not yet resilient system into a more resilient one by adding pumps of the Wilo-Economy MHIE series at most once. After solving the arising QMIP, it is suggested to buy the not yet present pump type 403 as an expansion of the network. Following this suggestion, the operational costs decrease in scenarios 2, 3 and 4 in comparison to the initial system. This is due to the fact that the initial system was not optimal itself for the given load scenarios—a circumstance occurring frequently as systems are often designed to cover a broad range of conditions for various applications. Regarding the financial effects of this investment decision, 2243.30, Euro can be saved over the five years compared to adding a fourth pump of the 406-type in order to increase resilience. These savings result from two different reasons: Firstly, selecting the 404 pump with lower investment costs and secondly, being able to operate more efficiently in the individual load scenarios as a better system operating point can be reached with the addition of a 404-type pump.

## 7 Thermofluid systems (Tile 3.1)

From a technical point of view, thermofluid systems can be regarded as fluid systems with superimposed heat transfer. For the modeling of these systems several possibilities with different focal points are presented in literature. However, in the following, we focus on two essential aspects: maintaining a simple representation and the compatibility to the fluid model from Sect. 4. We therefore extend the optimization model for fluid systems by the introduction of additional constraints dealing with heating and cooling. This involves introducing the physical quantities heat flow and temperature and their interactions in the model as well as taking care of the additional component groups necessary for heating and cooling as presented in Sect. 3.5. Since this is an extension of fluid systems, only the additional constraints are shown here and the previously presented constraints of Sect. 4 still apply. Hence,

**Table 9** Additional variables and parameters of the optimization model extension for thermofluid systems

| Symbol                  | Range            | Description   |
|-------------------------|------------------|---|
| $\dot{q}_{ij}^{in\ s}$  | $\mathbb{R}_0^+$ | Heat flow at the inlet of component $(i, j)$ in scenario $s$      |
| $\dot{q}_{ij}^{out\ s}$ | $\mathbb{R}_0^+$ | Heat flow at the outlet of component $(i, j)$ in scenario $s$     |
| $t_k^s$                 | $\mathbb{R}_0^+$ | Temperature at port $k$ in scenario $s$                           |
| $\Delta\dot{q}_{ij}^s$  | $\mathbb{R}$     | Heat increase/decrease by component $(i, j)$ in scenario $s$      |
| $t_{ij}^s$              | $\mathbb{R}_0^+$ | Outlet temperature of temperature source $(i, j)$ in scenario $s$ |
| $TS(E)$                 | –                | Set of (ideal) temperature sources                                |
| $\dot{Q}^{max}$         | –                | Upper bound on the heat flow                                      |
| $T^{max}$               | –                | Upper bound on the temperature                                    |
| $T_k^{min/max\ s}$      | –                | Lower/upper bound on the temperature at port $k$ in scenario $s$  |

the full MILP results from joining both parts. All new variables and parameters used are shown in Table 9.

$$\dot{q}_{ij}^{in\ s} \leq \dot{Q}^{max} \cdot a_{ij}^s \quad \forall s \in S, (i, j) \in E \tag{57}$$

$$\dot{q}_{ij}^{out\ s} \leq \dot{Q}^{max} \cdot a_{ij}^s \quad \forall s \in S, (i, j) \in E \tag{58}$$

$$\sum_{(i,k) \in E} \dot{q}_{i,k}^{out\ s} - \sum_{(k,j) \in E} \dot{q}_{k,j}^{in\ s} = 0 \quad \forall s \in S, k \in V \setminus \{S_G, T_G\} \tag{59}$$

$$t_k^s = T \left( \sum_{(i,k) \in E} \dot{v}_{i,k}^s, \sum_{(i,k) \in E} \dot{q}_{i,k}^{out\ s} \right) \quad \forall s \in S, k \in V \setminus \{S_G, T_G\} \tag{60}$$

$$T(\dot{v}_{ij}^s, \dot{q}_{ij}^{in\ s}) \leq t_i^s + (1 - a_{ij}^s) \cdot T^{max} \quad \forall s \in S, (i, j) \in E \tag{61}$$

$$T(\dot{v}_{ij}^s, \dot{q}_{ij}^{in\ s}) \geq t_i^s - (1 - a_{ij}^s) \cdot T^{max} \quad \forall s \in S, (i, j) \in E \tag{62}$$

$$\dot{q}_{ij}^{out\ s} \leq \dot{q}_{ij}^{in\ s} + \Delta\dot{q}_{ij}^s + (1 - a_{ij}^s) \cdot \dot{Q}^{max} \quad \forall s \in S, (i, j) \in E \setminus TS(E) \tag{63}$$

$$\dot{q}_{ij}^{out\ s} \geq \dot{q}_{ij}^{in\ s} + \Delta\dot{q}_{ij}^s - (1 - a_{ij}^s) \cdot \dot{Q}^{max} \quad \forall s \in S, (i, j) \in E \setminus TS(E) \tag{64}$$

$$t_j^s \leq t_{ij}^s + (1 - a_{ij}^s) \cdot T^{max} \quad \forall s \in S, (i, j) \in TS(E) \tag{65}$$

$$t_j^s \geq t_{ij}^s - (1 - a_{ij}^s) \cdot T^{max} \quad \forall s \in S, (i, j) \in TS(E) \tag{66}$$

$$T_k^{min\ s} \leq t_k^s \leq T_k^{max\ s} \quad \forall s \in S, k \in V \quad (67)$$

$$\Delta \dot{q}_{ij}^s = \Delta \dot{Q}(a_{ij}^s, n_{ij}^s, v_{ij}^s, t_i^s, t_j^s) \quad \forall s \in S, (i, j) \in E \setminus TS(E) \quad (68)$$

$$t_i^s = T(a_{ij}^s, n_{ij}^s, v_{ij}^s, \dot{q}_{ij}^{in\ s}, \dot{q}_{ij}^{out\ s}) \quad \forall s \in S, (i, j) \in TS(E) \quad (69)$$

$$p_{ij}^s = P(a_{ij}^s, n_{ij}^s, v_{ij}^s, \Delta \dot{q}_{ij}^s, t_{ij}^s, \Delta h_{ij}^s) \quad \forall s \in S, (i, j) \in E \quad (70)$$

If a component is operational, its heat flow is reasonable or vanishes otherwise, see Constraints (57) and (58). Note that since the heat flow, unlike the volume flow, can change along edges, two variables rather than one are required to model it. The variable  $\dot{q}_{ij}^{in\ s}$  represents the heat flow directly behind a vertex corresponding to a component's inlet. Following this,  $\dot{q}_{ij}^{out\ s}$  represents the heat flow directly before a vertex corresponding to a component's outlet. Due to the law of flow conservation, the heat flow has to be preserved at all vertices, except for the sources and sinks, see Constraint (59). For the mixing of incoming flows at the vertices, except for the sources and the sinks, the resulting temperature depends on the sums of the incoming volume flows and heat flows, see Constraint (60). Note that  $T(\dot{v}, \dot{q})$  describes the non-linear relationship according to the specific heat formula with the specific heat  $c$  being held constant. Furthermore, all flows exiting an operational component must have the same temperature, see Constraints (61) and (62). If a heating (or cooling) component is operational, the transferred heat increases (or decreases) the heat flow, see Constraints (63) and (64). For non-heating (or -cooling) components this increase (or decrease) is typically 0. However, there is an exception as we differentiate between two ideal sources of thermal energy: ideal heat sources and ideal temperature sources. While an ideal heat source provides a constant heat flow, ideal temperature sources maintain a constant outlet temperature. In the case of ideal temperature sources denoted by  $TS(E)$ , Constraints (63) and (64) do not apply. Rather a constant temperature is assigned to an operational component's outlet, see Constraints (65) and (66). Constraint (67) enables the setting of target values for the temperature at certain points in the system. The generally non-linear operating behavior of components and the determination of their respective operating points is represented by Constraints (68)–(70). This is an extension to Constraint (30) since the behavior of the component groups dealing with heating and cooling may depend on the adjacent temperature or heat flow. For the application example of a compression chiller, the corresponding relationships are shown in Sect. 3.5.

The non-linear constraints for describing the component behavior are again piecewise linearly approximated using the linearization techniques presented in Vielma et al. (2010). Furthermore, the heat transfer leads to an additional non-linearity. However, even with this, the model is still manageable since the non-linearity is only of bilinear nature. A suitable approach to linearize this relationship is presented in Geißler (2011).

## 8 Handling dynamic system behavior (*Tile 4.1*)

In the previous considerations it was assumed that similar loads can be aggregated to so-called scenarios since the state of a system at a certain point in time does not depend on the temporal sequence. However, this is not applicable whenever the actual state of the system depends on its load history and thus a path dependency occurs. Regarding thermofluid systems this is the case for systems with storage, components with extensive start-up and run-down phases, general delayed system responses or the like. While similar work, e.g. in the field of optimizing transient gas networks (cf. Mahlke et al. 2010), conducts a very detailed and comprehensive modeling of dynamic effects, partly due to the more complex characteristics of gas in contrast to water, in our contribution to handle dynamic system behavior, we focus on a deliberately simpler implementation to ensure practical applicability with reasonable technical simplifications. Therefore, an appropriate time representation which meets this requirement must be developed. It should be noted that the approach presented here focuses on the application of storage components, although it may be applicable for the other purposes mentioned above, too.

According to Floudas and Lin (2004), two different types for the representation of time exist—discrete and continuous representations. The first one, which is the most widely used approach in related literature, divides the observation period into uniform time intervals. All system events—the internal and external actions that cause the system to leave the current state—are associated with the start or the end of an interval. While the benefits of this representation—including a reference grid for all operations, an easy implementation and typically well-structured mathematical problems—seem attractive for some cases, it also has major disadvantages. Because of the a-priori fixed intervals and interval lengths, events are limited to these points in time. For this reason, the discrete representation is only an approximation, with its resolution depending on the number of intervals. However, more intervals lead to higher computational effort. Therefore, a trade-off between accuracy and the computational effort must be made. Additionally, the discrete representation leads to larger instances than necessary since the intervals are typically uniform and therefore the length of an interval is the smallest common divider of the duration of each considered (constant) load that occurs. This is especially the case for real-world applications (Floudas and Lin 2004).

Due to the discussed disadvantages, we focus on a continuous-time representation. For this, a global event-based approach is used. This means that the event points (or actions) define a joint, unified reference grid for all components of the system, while a unit-specific event-based approach would introduce its own reference grid for each component (Floudas and Lin 2004). The basic idea is that (additional) variables are used to determine the timings of the intervals. However, there are also challenges for this approach. Non-linearities arise due to the fact that the interval lengths are no longer constant but variable. Furthermore, the estimation and adjustment of the number of time intervals is a challenge. If the number of intervals is underestimated, inaccurate solutions or even infeasibility may occur. If, on the

other hand, the number of intervals is overestimated, unnecessarily large instances arise.

In order to describe the approach, a short introduction of the properties of storage components is given at this point. Generally, the filling level of a storage component at a point in time  $t$  can be determined using the flow balance equation:

$$V_t = V_{t-1} + \int_{t-1}^t (\dot{V}_{in} - \dot{V}_{out}) \quad (71)$$

For constant flows between  $t - 1$  and  $t$  the equation simplifies to:

$$V_t = V_{t-1} + (\dot{V}_{in} - \dot{V}_{out}) \cdot (\tau_t - \tau_{t-1}) = V_{t-1} + \Delta\dot{V} \cdot \Delta\tau \quad (72)$$

It can be seen that a non-linear term still exists since the two variables  $\Delta\dot{V}$  and  $\Delta\tau$  are multiplied. Nevertheless, the relation becomes easier to handle compared to Eq. (71) if only constant flows occur. The resulting challenge is how to choose the (maximum) number of intervals to ensure that only constant flows occur.

In the following, a system with only one source and one sink is used for illustration purposes. In this case, flows are constant as long as the demand at the source  $\dot{V}_{source}$ , which corresponds to the system's demand, is constant:

$$\dot{V}_{sink} + \sum_{i \in SC} (\dot{V}_i^{in} - \dot{V}_i^{out}) = \dot{V}_{source} \quad (73)$$

The demand of the system changes every time an activity at the sink  $\dot{V}_{sink}$ , i.e. on the consumer side, takes place or the demand changes indirectly due to the filling or emptying of a storage component out of the set  $SC$ . The change in demand due to the first is called a main event. Therefore, the number of main events is known in advance because of the a-priori determined projected demands by the consumer—comparable to the load scenarios in the aggregated representation but with respect to their chronological order—whereas the number of intervals between the main events still needs to be determined.

*Key observation:* If there is a constant demand at the sink, a storage component should strive to empty as early as possible and to fill as late as possible during this period to avoid energy losses. Even if energy losses are not explicitly considered, in many cases, it is reasonable to assume that the filling or emptying takes place in only one continuous process instead of multiple, interrupted processes right before or after a main event. As an appropriate technical simplification, we define that at most one filling and one emptying process per storage component takes place between two main events. On this basis, the upper bound on the number of intervals ( $n_{Intervals}$ ) between two main events usually decreases significantly and is connected to the number of sources ( $n_{Sources}$ ) and storage components ( $n_{SC}$ ) as follows:

$$n_{Intervals} = n_{Sources} + 2 \cdot n_{SC} \quad (74)$$

In the case of one source and one storage component, as shown in Fig. 10, we assume that there are at most three intervals  $i$  between two main events  $me$ : one for the emptying of the storage component  $i_{1,1}$ , one if there is no change for the

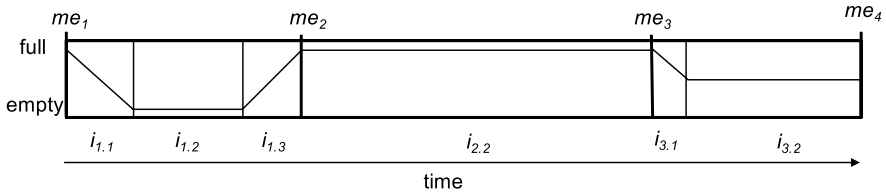


Fig. 10 Schematic representation of the system behavior

Table 10 Additional variables and parameters for the extended thermofluid system model

| Symbol  | Range            | Description  |
|---|------------------|--|
| $\Delta v_{\tau}^{e,l} / \Delta q_{\tau}^{e,l}$ | $\mathbb{R}$     | Relative volume/heat change of storage component $(i, \tau, j)$ in interval $l$ for event $e$        |
| $\Delta v_{\tau}^{e,l} / \Delta q_{\tau}^{e,l}$ | $\mathbb{R}$     | Absolute volume/heat change of storage component $(i, \tau, j)$ in interval $l$ for event $e$        |
| $d^{e,l}$                                       | $\mathbb{R}_0^+$ | Duration of interval $l$ for event $e$   |
| $v_{\tau}^{begin e,l} / q_{\tau}^{begin e,l}$   | $\mathbb{R}_0^+$ | Volume/heat level of storage component $(i, \tau, j)$ at the beginning of interval $l$ for event $e$ |
| $v_{\tau}^{end e,l} / q_{\tau}^{end e,l}$       | $\mathbb{R}_0^+$ | Volume/heat level of storage component $(i, \tau, j)$ at the end of interval $l$ for event $e$       |
| $t_{\tau}^{e,l}$                                | $\mathbb{R}_0^+$ | Temperature of storage component $(i, \tau, j)$ at the beginning of interval $l$ for event $e$       |
| $A$   | –                | Set of additional vertices for storage components representing the property to store fluid           |
| $\mathcal{E}$                                   | –                | Set of ordered (main-)events   |
| $I$   | –                | Set of intervals within an event   |
| $D^e$   | –                | Total duration between events $e$ and $e + 1$  |
| $v_{\tau}^{init} / Q_{\tau}^{init}$             | –                | Initial volume/heat level of storage component $(i, \tau, j)$  |
| $V_{\tau}^{max}$                                | –                | Maximum filling level of storage component $(i, \tau, j)$  |

component  $i_{1,2}$  and one for the filling of the component  $i_{1,3}$ . Figure 10 also illustrates that the determined number of intervals is only an upper bound. Between the main events  $me_1$  and  $me_2$  all three intervals are needed, while between  $me_2$  and  $me_3$  one interval would be sufficient as the storage component is neither filled nor emptied and the demand is satisfied by a continuous flow from the source to the sink exclusively. Therefore,  $i_{2,1}$  and  $i_{2,3}$  are assumed to have a length of 0.

In the following, the basic model extension for this approach is presented. The additional variables and parameters required are shown in Table 10. Besides that, the aggregated representation described above can be easily adapted by replacing the load scenarios  $S$  by the corresponding main events  $\mathcal{E}$  and intervals  $I$ . It should be noted that storage components are a special type of components as they can also act as (volume-restricted) sources or sinks. They are, therefore, modeled as ordinary components  $(i, j)$  with an additional vertex  $\tau \in A$  in between representing the property to store fluid  $(i, \tau, j)$ . Furthermore, the objective has to be modified because the duration of an interval is given by the value of variable

$d^{e,l}$  instead of predefined time shares of the operational lifespan ( $F^s \cdot OLS$ ) as shown in Objective (19). Hence, Objective (75) arises with the non-linear relationship  $E(d, p)$  representing the product of the interval duration and the power consumption.

$$\min \sum_{(i,j) \in E} (C_{ij}^{buy} \cdot b_{ij}) + C^{kWh} \cdot \sum_{e \in \mathcal{E}} \sum_{l \in I} E\left(d^{e,l}, \left(\sum_{(i,j) \in E} p_{ij}^{e,l}\right)\right) \quad (75)$$

$$b_{i,\tau} = b_{\tau,j} \quad \forall (i, \tau), (\tau, j) \in E : \tau \in A \quad (76)$$

$$d_{i,\tau}^{e,l} + d_{\tau,j}^{e,l} \leq 1 \quad \forall e \in \mathcal{E}, l \in I, \\ (i, \tau), (\tau, j) \in E : \tau \in A \quad (77)$$

$$\sum_{(i,\tau) \in E} \dot{v}_{i,\tau}^{e,l} - \sum_{(\tau,j) \in E} \dot{v}_{\tau,j}^{e,l} = \Delta \dot{v}_{\tau}^{e,l} \quad \forall e \in \mathcal{E}, l \in I, \tau \in A \quad (78)$$

$$\sum_{(i,\tau) \in E} \dot{q}_{i,\tau}^{out\ e,l} - \sum_{(\tau,j) \in E} \dot{q}_{\tau,j}^{in\ e,l} = \Delta \dot{q}_{\tau}^{e,l} \quad \forall e \in \mathcal{E}, l \in I, \tau \in A \quad (79)$$

$$\sum_{l \in I} d^{e,l} = D^e \quad \forall e \in \mathcal{E} \quad (80)$$

$$\Delta v_{\tau}^{e,l} = V(\Delta \dot{v}_{\tau}^{e,l}, d^{e,l}) \quad \forall e \in \mathcal{E}, l \in I, \tau \in A \quad (81)$$

$$\Delta q_{\tau}^{e,l} = Q(\Delta \dot{q}_{\tau}^{e,l}, d^{e,l}) \quad \forall e \in \mathcal{E}, l \in I, \tau \in A \quad (82)$$

$$v_{\tau}^{begin\ e,l} = v_{\tau}^{init.} + \sum_{\varepsilon \in \mathcal{E}: \varepsilon < e} \sum_{l \in I} \Delta v_{\tau}^{\varepsilon,l} \\ + \sum_{\lambda \in I: \lambda < l} \Delta v_{\tau}^{e,\lambda} \quad \forall e \in \mathcal{E}, l \in I, \tau \in A \quad (83)$$

$$v_{\tau}^{end\ e,l} = v_{\tau}^{begin\ e,l} + \Delta v_{\tau}^{e,l} \quad \forall e \in \mathcal{E}, l \in I, \tau \in A \quad (84)$$

$$q_{\tau}^{begin\ e,l} = Q_{\tau}^{init.} + \sum_{\varepsilon \in \mathcal{E}: \varepsilon < e} \sum_{l \in I} \Delta q_{\tau}^{\varepsilon,l} + \sum_{\lambda \in I: \lambda < l} \Delta q_{\tau}^{e,\lambda} \quad \forall e \in \mathcal{E}, l \in I, \tau \in A \quad (85)$$

$$q_{\tau}^{end\ e,l} = q_{\tau}^{begin\ e,l} + \Delta q_{\tau}^{e,l} \quad \forall e \in \mathcal{E}, l \in I, \tau \in A \quad (86)$$

$$v_{\tau}^{end\ e,l} \leq V_{\tau}^{max} \quad \forall e \in \mathcal{E}, l \in I, \tau \in A \quad (87)$$

$$t_{\tau}^{e,l} = T(v_{\tau}^{begin\ e,l}, q_{\tau}^{begin\ e,l}) \quad \forall e \in \mathcal{E}, l \in I, \tau \in A \quad (88)$$

For a storage component either both of its edges or neither of them can be installed, see Constraint (76). Furthermore, a storage component can either be filled or emptied but not both at a time, see Constraint (77). For a component's storage vertex, the law of flow conservation has to be modified to model the relative volume and heat changes of the component, see Constraints (78) and (79). Constraint (80) ensures that the accumulated duration of all intervals assigned to an event is equal to the period between that event and the next event. The resulting absolute volume and heat change within an interval is then determined by the relative change as well as the duration of the interval, see Constraints (81) and (82). These changes can be further used to calculate the volume and heat levels of the storage component at the beginning and end of each interval, see Constraints (83)–(86). For this, it has to be ensured that the component's storage capacity is not exceeded, see Constraint (87). In contrast to regular vertices, the temperature at a storage vertex is defined by its content, see Constraint (88).

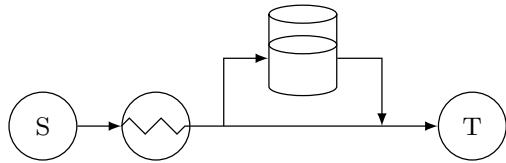
With the presented approach, the number of vertices in the space-time-graph increases only linearly with the number of storage components. In comparison to the discrete representation with a similar resolution, a smaller space-time-graph results because due to the continuous representation the resolution does not depend on the length of a predefined time step. Instead, the time intervals are only assigned to those points in time at which a possible change (as the number of intervals is an upper bound) can occur in the system. While it is not universally applicable, there are important areas of application in the field of thermofluid systems: water storage tanks, thermally stratified storage tanks and water-filled pipes. Whereas the potential applications of the first two are obvious, water-filled pipes are, inter alia, a crucial component for the temperature control of plastics processing.



As a proof of concept, we investigate a system which consists of one source ( $S$ ), one simple electrical heater ( $H$ ) with a maximum power of 300 kW, one (initially empty) water tank ( $A$ ) with a capacity of 50 l and one sink ( $T$ ), connected as illustrated in Fig. 11. The structure of the system is already fixed and only the operation, especially of the water tank, is investigated. The objective is to minimize the energy consumption of the heater, assuming its efficiency to be equal to 1, over two different consecutive main events. Table 11 shows the duration of the main events ( $D^e$ ), the demanded volume flow ( $\dot{V}_T^e$ ), the available temperature at the source ( $T_S^e$ ) and the demanded temperature at the sink ( $T_T^e$ ). Note that the heater's maximum power is not sufficient to satisfy the load for the second main event on its own.

The minimal energy consumption is approximately 11 kWh which can be confirmed easily. The solution is summarized in Table 12. Three intervals, 1.2, 1.3 and 2.1, have a length greater than 0. These intervals are associated with the three different modes for a system with one source and one storage component according to the description above. In interval 1.2 the water tank is neither filled nor emptied and the demand is satisfied by just-in-time generation. In interval 1.3 the tank is filled in order to be able to satisfy the additional demand during the second main event and in interval 2.1 the tank is finally emptied.

**Fig. 11** System structure for the proof of concept



**Table 11** Load data for the proof of concept

| $e$ | $D^e$ [s] | $\dot{V}_T^e$ [l/s] | $T_S^e$ [°C] | $T_T^e$ [°C] |
|-----|-----------|---------------------|--------------|--------------|
| 1   | 100.00    | 1.00                | 10.00        | 60.00        |
| 2   | 50.00     | 1.50                | 10.00        | 70.00        |

**Table 12** Solution for the proof of concept

| $e$ | $l$ | $d^{e,l}$ [s] | $\dot{v}_S^{e,l}$ [l/s] | $\dot{v}_T^{e,l}$ [l/s] | $t_H^{e,l}$ [°C] | $p_H^{e,l}$ [kW] | $v_A^{begin\ e,l}$ [l] | $v_A^{end\ e,l}$ [l] | $t_A^{e,l}$ [°C] |
|-----|-----|---------------|-------------------------|-------------------------|------------------|------------------|------------------------|----------------------|------------------|
| 1   | 1   | 0.00          | –                       | –                       | –                | –                | –                      | –                    | –                |
| 1   | 2   | 9.13          | 1.00                    | 1.00                    | 60.00            | 209.10           | 0.00                   | 0.00                 | –                |
| 1   | 3   | 90.87         | 1.43                    | 1.00                    | 60.00            | 300.00           | 0.00                   | 39.50                | –                |
| 2   | 1   | 50.00         | 0.71                    | 1.50                    | 81.13            | 211.17           | 39.50                  | 0.00                 | 60.00            |
| 2   | 2   | 0.00          | –                       | –                       | –                | –                | –                      | –                    | –                |
| 2   | 3   | 0.00          | –                       | –                       | –                | –                | –                      | –                    | –                |

## 9 Conclusion and outlook

The essence of this paper was to make a contribution to our vision of providing tools for engineers which supplement the human intuition during the design of technical systems. Within this context, we kept an even balance between the different perspectives of both disciplines, engineering and mathematical optimization, such that both can benefit from each other. For the consideration of thermofluid systems, the overall vision was divided into individual subgoals. Based on the simple model for fluid systems, selected subgoals were addressed in this paper to provide substantial progress towards the overall vision. In the following, we conclude our contributions and identify future research directions.

### 9.1 Algorithmic synthesis of fluid systems (*Tile 1.2*)

With regard to fluid systems, a Branch-and-Bound framework was introduced. Using primal and dual heuristics that rely on domain-specific knowledge, we were able to solve relatively large instances in reasonable time using the application example of booster stations. These instances were designed in such a way that they correspond to practice-relevant applications with varying demands for the pressure increase at time-variant volume flow rates and four to five different load scenarios. With our approach, we outperformed state-of-the-art MILP solvers which could not solve the provided instances. The runtimes of the individual heuristics and the Branch-and-Bound framework were fast enough to be of practical relevance for the considered application. In the following steps, the results have to be validated by simulation. In addition, we plan the adaption of this approach to thermofluid systems. For thermofluid systems without storage components, this adaptation is quite straightforward. In the case of the time-dependent formulation, it has to be examined to what extent the additional couplings of time steps resulting from the storage components can be reasonably integrated.

### 9.2 Resilient system design (*Tile 2.1*)

Furthermore, a QMIP formulation was used in order to design cost-efficient resilient systems, again exemplified by booster stations. In the case of booster stations, the resulting systems are in line with DIN 1988-500 (2011) and can be directly integrated into the proposed workflow. The approach proved to have the potential to support system designers in two different ways. Firstly, increasing resilience is made easy. The system designer can focus on the main functionality while the approach takes care of resilience. Also existing non-resilient systems can be transformed into more resilient ones without questioning the initial system. Secondly, the approach helps to overcome smaller design flaws. On top of increasing resilience it can help to save energy. This is also beneficial with regard to off-the-shelf systems as they can be made more resilient as well as adapted to the actual load conditions simultaneously. Thus, the presented approach combines resource-efficiency and reliability. In

the future, we plan to adopt this approach for other tasks in the domain of fluid and thermofluid systems, paying attention to the relevant problem-specific properties.

### 9.3 Handling dynamic system behavior (*Tile 4.1*)

Finally, we presented two MILP extensions for the synthesis of thermofluid systems. While the basic first extension aimed at the inclusion of general heat transfer, the second one included the introduction of a continuous time-representation for technical fluid-based systems with time-dependent behavior. This extension focuses on the dynamic effects resulting from the use of storage components. In this context, the time-dependency is considered in such a way that its essential properties can still be taken into account, while the representation is simple enough to be applicable for optimization. However, this approach is particularly advantageous if there is a manageable number of load changes since then significantly fewer time steps have to be considered in comparison to the widely used time-discrete approaches. This shows that practice-relevant dynamics do not necessarily cause instances of optimization models to explode in size. Furthermore, the model aims at providing a unified framework for current and future work on technical applications in the field of thermofluid systems. Therefore, attention was paid to a formulation that is as comprehensive and generally applicable as possible as well as to the suitability for integration into a software tool for system designers. As for future research, we work towards making the model more efficient with regard to the number of linearizations needed and alternative linearization techniques for the arising non-linearities. In addition, we plan to develop application-related algorithmic methods in order to efficiently solve larger instances.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## Appendix 1: Summary of variables and parameters for the optimization model and extensions

Table 13 shows a summary of all variables and parameters used for the basic model and the extensions for heat transfer and time-dependent behavior. Note that for the time-dependent model the index for the load scenario  $s \in S$  has to be replaced by the corresponding indices for the main event  $e \in \mathcal{E}$  and the interval  $l \in I$ .

**Table 13** Summary of variables and parameters for the optimization model

| Symbol  | Range            | Description  |
|---|------------------|--|
| $b_{i,j}$   | $\{0, 1\}$       | Purchase decision of component $(i, j)$  |
| $a_{i,j}^s$   | $\{0, 1\}$       | Activation decision of component $(i, j)$ in scenario $s$  |
| $\dot{v}_{i,j}^s$                                     | $\mathbb{R}_0^+$ | Volume flow through component $(i, j)$ in scenario $s$   |
| $h_k^s$   | $\mathbb{R}_0^+$ | Pressure at port $k$ in scenario $s$   |
| $p_{i,j}^s$   | $\mathbb{R}_0^+$ | Power consumption of component $(i, j)$ in scenario $s$  |
| $n_{i,j}$   | $[0, 1]$         | Relative speed (or level) of component $(i, j)$ in scenario $s$                                      |
| $\Delta h_{i,j}^s$                                    | $\mathbb{R}$     | Pressure increase/decrease by component $(i, j)$ in scenario $s$                                     |
| $\dot{q}_{i,j}^{in\ s}$                               | $\mathbb{R}_0^+$ | Heat flow at the inlet of component $(i, j)$ in scenario $s$   |
| $\dot{q}_{i,j}^{out\ s}$                              | $\mathbb{R}_0^+$ | Heat flow at the outlet of component $(i, j)$ in scenario $s$  |
| $t_k^s$   | $\mathbb{R}_0^+$ | Temperature at port $k$ in scenario $s$  |
| $\Delta \dot{q}_{i,j}^s$                              | $\mathbb{R}$     | Heat increase/decrease by component $(i, j)$ in scenario $s$   |
| $t_{i,j}^s$   | $\mathbb{R}_0^+$ | Outlet temperature of temperature source $(i, j)$ in scenario $s$                                    |
| $\Delta v_{\tau}^{e,l} / \Delta \dot{q}_{\tau}^{e,l}$ | $\mathbb{R}$     | Relative volume/heat change of storage component $(i, \tau, j)$ in interval $l$ for event $e$        |
| $\Delta v_{\tau}^{e,l} / \Delta q_{\tau}^{e,l}$       | $\mathbb{R}$     | Absolute volume/heat change of storage component $(i, \tau, j)$ in interval $l$ for event $e$        |
| $d^{e,l}$   | $\mathbb{R}_0^+$ | Duration of interval $l$ for event $e$   |
| $v_{\tau}^{begin\ e,l} / q_{\tau}^{begin\ e,l}$       | $\mathbb{R}_0^+$ | Volume/heat level of storage component $(i, \tau, j)$ at the beginning of interval $l$ for event $e$ |
| $v_{\tau}^{end\ e,l} / q_{\tau}^{end\ e,l}$           | $\mathbb{R}_0^+$ | Volume/heat level of storage component $(i, \tau, j)$ at the end of interval $l$ for event $e$       |
| $t_{\tau}^{e,l}$                                      | $\mathbb{R}_0^+$ | Temperature of storage component $(i, \tau, j)$ at the beginning of interval $l$ for event $e$       |
| $S$   | –                | Set of scenarios   |
| $E$   | –                | Set of components  |
| $V$   | –                | Set of ports   |
| $S_G$   | –                | Set of sources   |
| $T_G$   | –                | Set of sinks   |
| $TS(E)$   | –                | Set of (ideal) temperature sources   |
| $A$   | –                | Set of additional vertices for storage components representing the property to store fluid           |
| $\mathcal{E}$   | –                | Set of ordered (main-)events   |
| $I$   | –                | Set of intervals within an event   |
| $C^{kWh}$   | –                | Energy costs per kilowatt hour   |
| $C_{i,j}^{buy}$                                       | –                | Purchase costs of component $(i, j)$   |
| $OLS$   | –                | Operational lifespan of the system   |
| $F^s$   | –                | Fraction of the operational lifespan $OLS$ of scenario $s$   |
| $\dot{V}^{max}$                                       | –                | Upper bound on the volume flow   |
| $H^{max}$   | –                | Upper bound on the pressure  |
| $\dot{V}_{in/out\ k}^{min/max\ s}$                    | –                | Lower/upper bound on the volume flow entering/leaving port $k$ in scenario $s$                       |
| $H_k^{min/max\ s}$                                    | –                | Lower/upper bound on the pressure at port $k$ in scenario $s$  |
| $\dot{Q}^{max}$                                       | –                | Upper bound on the heat flow   |
| $T^{max}$   | –                | Upper bound on the temperature   |
| $T_k^{min/max\ s}$                                    | –                | Lower/upper bound on the temperature at port $k$ in scenario $s$                                     |
| $D^e$   | –                | Total duration between events $e$ and $e + 1$  |

**Table 13** (continued)

| Symbol                            | Range | Description   |
|-----------------------------------|-------|---|
| $V_{\tau}^{init}/Q_{\tau}^{init}$ | –     | Initial volume/heat level of storage component ( $i, \tau, j$ ) |
| $V_{\tau}^{max}$                  | –     | Maximum filling level of storage component ( $i, \tau, j$ )     |

## References

- Afram A, Janabi-Sharifi F (2014) Theory and applications of HVAC control systems: a review of model predictive control (MPC). *Build Environ* 72:343–355
- Altherr LC (2016) Algorithmic system design under consideration of dynamic processes. Dissertation, Technische Universität Darmstadt
- Altherr LC, Brötz N, Dietrich I, Gally T, Gebner F, Kloberdanz H, Leise P, Pelz PF, Schlemmer PD, Schmitt A (2018a) Resilience in mechanical engineering—a concept for controlling uncertainty during design, production and usage phase of load-carrying structures. In: *Uncertainty in mechanical engineering III*. Trans Tech Publications Ltd, Switzerland, pp 187–198
- Altherr LC, Dörig B, Ederer T, Pelz PF, Pfetsch ME, Wolf J (2018b) A mixed-integer nonlinear program for the design of gearboxes. In: *Operations research proceedings 2016*. Springer, Cham, pp 227–233
- Altherr LC, Leise P, Pfetsch ME, Schmitt A (2019) Resilient layout, design and operation of energy-efficient water distribution networks for high-rise buildings using MINLP. *Optim Eng* 20(2):605–645
- Andiappan V (2017) State-of-the-art review of mathematical optimisation approaches for synthesis of energy systems. *Process Integr Optim Sustain* 1(3):165–188
- Ashouri A, Fux SS, Benz MJ, Guzzella L (2013) Optimal design and operation of building services using mixed-integer linear programming techniques. *Energy* 59:365–376
- Bellman RE (2003) *Dynamic programming*. Dover Publications, New York
- Ben-Tal A, El Ghaoui L, Nemirovski AS (2009) *Robust optimization*. Princeton University Press, Princeton
- Betz P (2017) Über die Anwendung exakter Optimierungsverfahren zur Planung von Druckerhöhungsanlagen. Dissertation, Technische Universität Darmstadt
- Birge JR, Louveaux F (2011) *Introduction to stochastic programming*, 2nd edn. Springer, New York
- Boussaïd I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. *Inf Sci* 237:82–117
- Bragalli C, D'Ambrosio C, Lee J, Lodi A, Toth P (2012) On the optimal design of water distribution networks: a practical MINLP approach. *Optim Eng* 13(2):219–246
- Cai X, McKinney DC, Lasdon LS (2001) Solving nonlinear water management models using a combined genetic algorithm and linear programming approach. *Adv Water Resour* 24(6):667–676
- Clausen J (1999) Branch and bound algorithms—principles and examples. Technical report, Department of Computer Science, University of Copenhagen
- Cunha MC, Ribeiro L (2004) Tabu search algorithms for water network optimization. *Eur J Oper Res* 157(3):746–758
- Cunha MC, Sousa J (1999) Water distribution network design optimization: simulated annealing approach. *J Water Resour Plan Manag* 125(4):215–221
- D'Ambrosio C, Lodi A, Wiese S, Bragalli C (2015) Mathematical programming techniques in water network optimization. *Eur J Oper Res* 243(3):774–788
- De Corte A, Sörensen K (2013) Optimisation of gravity-fed water distribution network design: a critical review. *Eur J Oper Res* 228(1):1–10
- Deng K, Sun Y, Chakraborty A, Lu Y, Brouwer J, Mehta PG (2013) Optimal scheduling of chiller plant with thermal energy storage using mixed integer linear programming. In: *Proceedings of the 2013 american control conference*, Washington, D.C., pp 2958–2963
- DIN 1988-300 (2012) Codes of practice for drinking water installations—part 300: pipe sizing; DVGW code of practice
- DIN 1988-500 (2011) Codes of practice for drinking water installations—part 500: pressure boosting stations with RPM-regulated pumps; DVGW code of practice
- Domschke P, Geißler B, Kolb O, Lang J, Martin A, Morsi A (2011) Combination of nonlinear and linear optimization of transient gas networks. *INFORMS J Comput* 23(4):605–617

- Ederer T, Lorenz U, Martin A, Wolf J (2011) Quantified linear programs: a computational study. In: Algorithms—ESA 2011. Springer, Berlin, pp 203–214
- Ederer T, Hartisch M, Lorenz U, Opfer T, Wolf J (2017) Yasol: an open source solver for quantified mixed integer programs. In: Advances in computer games. Springer, Cham, pp 224–233
- Executive Agency for Small and Medium-Sized Enterprises (2016) Overview of support activities and projects of the European Union on energy efficiency and renewable energy in the heating & cooling sector. Technical report, European Union
- Falkner H (2008) EuP Lot 11: water pumps (in commercial buildings, drinking water pumping, food industry, agriculture). Technical report, AEA Energy & Environment, Report to European Commission
- Floudas CA, Lin X (2004) Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Comput Chem Eng* 28(11):2109–2129
- Frangopoulos C, von Spakovsky M, Sciubba E (2002) A brief review of methods for the design and synthesis optimization of energy systems. *Int J Thermodyn* 5(4):151–160
- Fügenschuh A, Hayn C, Michaels D (2014) Mixed-integer linear methods for layout-optimization of screening systems in recovered paper production. *Optim Eng* 15(2):533–573
- Geißler B (2011) Towards globally optimal solutions for MINLPs by discretization techniques with applications in gas network optimization. Dissertation, Universität Erlangen-Nürnberg
- Geißler B, Kolb O, Lang J, Leugering G, Martin A, Morsi A (2011) Mixed integer linear models for the optimization of dynamical transport networks. *Math Methods Oper Res* 73(3):339–362
- Gerber R, Pugh W, Saksena M (1995) Parametric dispatching of hard real-time tasks. *IEEE Trans Comput* 44(3):471–479
- Gleixner AM, Held H, Huang W, Vigerske S (2012) Towards globally optimal operation of water supply networks. *Numer Algebra Control Optim* 2(4):695–711
- Gupta A, Pál M, Ravi R, Sinha A (2004) Boosted sampling: approximation algorithms for stochastic optimization. In: Proceedings of the thirty-sixth annual ACM symposium on theory of computing, ACM, New York, pp 417–426
- Gustafsson SI (1998) Mixed integer linear programming and building retrofits. *Energy Build* 28(2):191–196
- Hartisch M, Lorenz U (2019) Mastering uncertainty: towards robust multistage optimization with decision dependent uncertainty. In: PRICAI 2019: trends in artificial intelligence. Springer, Cham, pp 446–458
- Hartisch M, Ederer T, Lorenz U, Wolf J (2016) Quantified integer programs with polyhedral uncertainty set. In: Computers and games. Springer, Cham, pp 156–166
- Hartisch M, Herbst A, Lorenz U, Weber JB (2018) Towards resilient process networks—designing booster stations via quantified programming. In: Uncertainty in mechanical engineering III. Trans Tech Publications Ltd, Switzerland, pp 199–210
- Herrera M, Abraham E, Stoianov I (2016) A graph-theoretic framework for assessing the resilience of sectorised water distribution networks. *Water Resour Manag* 30(5):1685–1699
- Hydeman M, Gillespie KL Jr, Dexter AL (2002) Tools and techniques to calibrate electric chiller component models. *ASHRAE Trans* 108:733–741
- Incropera FP, DeWitt DP, Bergman TL, Lavine AS (2007) Fundamentals of heat and mass transfer, 6th edn. Wiley, Hoboken
- Kashima T, Boyd SP (2013) Cost optimal operation of thermal energy storage system with real-time prices. In: Proceedings of the 2013 international conference on control, automation and information sciences, Nha Trang, pp 233–237
- Leise P, Altherr LC, Pelz PF (2018) Energy-efficient design of a water supply system for skyscrapers by mixed-integer nonlinear programming. In: Operations research proceedings 2017. Springer, Cham, pp 475–481
- Lorenz U, Martin A, Wolf J (2010) Polyhedral and algorithmic properties of quantified linear programs. In: Algorithms—ESA 2010. Springer, Berlin, Heidelberg, pp 512–523
- MacMahon PA (1890) Yoke-chains and multipartite compositions in connexion with the analytical forms called “trees”. *Proc Lond Math Soc* 22:330–346
- Mahlke D, Martin A, Moritz S (2010) A mixed integer approach for time-dependent gas network optimization. *Optim Methods Softw* 25(4):625–644
- Maier HR, Simpson AR, Zecchin AC, Foong WK, Phang KY, Seah HY, Tan CL (2003) Ant colony optimization for design of water distribution systems. *J Water Resour Plan Manag* 129(3):200–209

- Mala-Jetmarova H, Sultanova N, Savić D (2018) Lost in optimisation of water distribution systems? A literature review of system design. *Water* 10(3):307
- Martin A, Klamroth K, Lang J, Leugering G, Morsi A, Oberlack M, Ostrowski M, Rosen R (2012) *Mathematical optimization of water networks*. Birkhäuser, Basel
- Meng F, Fu G, Farmani R, Sweetapple C, Butler D (2018) Topological attributes of network resilience: a study in water distribution systems. *Water Res* 143:376–386
- Morsi A, Geißler B, Martin A (2012) Mixed integer optimization of water supply networks. In: *Mathematical optimization of water networks*. Birkhäuser, Basel, pp 35–54
- Munson BR, Young DF, Okiishi TH, Huebsch WW (2009) *Fundamentals of fluid mechanics*, 6th edn. Wiley, Hoboken
- Pelz PF, Lorenz U, Ederer T, Lang S, Ludwig G (2012) Designing pump systems by discrete mathematical topology optimization: the artificial fluid systems designer (AFSD). In: *International rotating equipment conference (IREC) 2012, Düsseldorf*
- Pöttgen P, Pelz PF (2016) Evaluation of different approaches for the optimization of layout and control of booster stations. In: *ECCOMAS Congress 2016—VII European congress on computational methods in applied sciences and engineering, Crete Island*, pp 8391–8402
- Pöttgen P, Ederer T, Altherr L, Lorenz U, Pelz PF (2016) Examination and optimization of a heating circuit for energy-efficient buildings. *Energy Technol* 4(1):136–144
- Reintjes C, Hartisch M, Lorenz U (2018) Lattice structure design with linear optimization for additive manufacturing as an initial design in the field of generative design. In: *Operations research proceedings 2017*. Springer, Cham, pp 451–457
- Risbeck MJ (2018) *Mixed-integer model predictive control with applications to building energy systems*. Dissertation, University of Wisconsin-Madison
- Risbeck MJ, Maravelias CT, Rawlings JB, Turney RD (2015) Cost optimization of combined building heating/cooling equipment via mixed-integer linear programming. In: *Proceedings of the 2015 American control conference, Chicago*, pp 1689–1694
- Risbeck MJ, Maravelias CT, Rawlings JB, Turney RD (2017) A mixed-integer linear programming model for real-time cost optimization of building heating, ventilation, and air conditioning equipment. *Energy Build* 142:220–235
- Sanders P (2009) Algorithm engineering—an attempt at a definition. In: *Efficient algorithms*. Springer, Berlin, pp 321–340
- Saul B, Pöttgen P, Zimmermann W, Pelz PF (2016) SHEP: an innovative language to create and evaluate optimization programs for pump systems. In: *International rotating equipment conference (IREC) 2016, Düsseldorf*
- Savić D, Walters GA (1997) Genetic algorithms for least-cost design of water distribution networks. *J Water Resour Plan Manag* 123(2):67–77
- Subramani K (2004) Analyzing selected quantified integer programs. In: *Automated reasoning*. Springer, Berlin, pp 342–356
- Ulanicki B, Kahler J, Coulbeck B (2008) Modeling the efficiency and power characteristics of a pump group. *J Water Resour Plan Manag* 134(1):88–93
- VDI 2884 (2015) Purchase, operating and maintenance of production equipment using life cycle costing (LCC); VDI guidelines
- Vielma J, Ahmed S, Nemhauser G (2010) Mixed-integer models for nonseparable piecewise-linear optimization: unifying framework and extensions. *Oper Res* 58(2):303–315
- Voll P, Klaffke C, Hennen M, Bardow A (2013) Automated superstructure-based synthesis and optimization of distributed energy supply systems. *Energy* 50:374–388
- Weber JB, Lorenz U (2017) Optimizing booster stations. In: *Proceedings of the genetic and evolutionary computation conference companion, ACM, New York*, pp 1303–1310
- Weber JB, Lorenz U (2019a) Algorithmic system design of thermofluid systems. In: *EngOpt 2018 proceedings of the 6th international conference on engineering optimization, Springer, Cham*, pp 132–143
- Weber JB, Lorenz U (2019b) Modeling thermofluid systems: an approach customized for optimization. In: *Operations research proceedings 2018*. Springer, Cham, pp 387–393
- Weber JB, Hartisch M, Lorenz U (2020) Optimized design of thermofluid systems using the example of mold cooling in injection molding. In: *Operations research proceedings 2019, Springer, Cham* (to appear)
- Wets RJB (1974) Stochastic programs with fixed recourse: the equivalent deterministic program. *SIAM Rev* 16(3):309–339

Wolf J (2015) Quantified linear programming. Dissertation, Technische Universität Darmstadt

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Affiliations

**Jonas B. Weber<sup>1</sup> · Michael Hartisch<sup>1</sup> · Alexander D. Herbst<sup>1</sup> · Ulf Lorenz<sup>1</sup>**

Michael Hartisch  
michael.hartisch@uni-siegen.de

Alexander D. Herbst  
alexander.herbst@uni-siegen.de

Ulf Lorenz  
ulf.lorenz@uni-siegen.de

<sup>1</sup> Chair of Technology Management, University of Siegen, Unteres Schloss 3, 57072 Siegen, Germany