Optimizing Booster Stations

Jonas B. Weber
Chair of Technology Management
University of Siegen
Unteres Schloss 3
57072 Siegen, Germany
jonas.weber@uni-siegen.de

Ulf Lorenz
Chair of Technology Management
University of Siegen
Unteres Schloss 3
57072 Siegen, Germany
ulf.lorenz@uni-siegen.de

ABSTRACT

Booster stations are fluid systems consisting of interconnected components such as pumps, pipes, valves and fittings. One of their main applications is to supply whole buildings or higher floors with drinking water if the supply pressure of the water company is not high enough to guarantee a continuous supply for all consumers. This means that a booster station must increase the pressure of supplied drinking water at a given time-variant flow rate. The consumer's demands must be matched at any time and the system operation is restricted by the general laws of fluid mechanics.

A common approach to handle the corresponding optimization problem is to model it as a mixed integer linear program (MILP) and to solve this program using a standard MILP solver. This approach is not suitable for large problem instances with practical relevance as it is not possible to obtain good solutions in reasonable time. Hence, the main obstacle is the optimization speed.

In this work, we present an approach to obtain good solutions in reasonable time even for large practical relevant instances. We do this by addressing the problem with heuristics from both the primal and dual side combined with the use of problem specific and technical knowledge. This approach is based on modeling the problem as a MILP as well as a mathematical graph and using both views simultaneously.

CCS CONCEPTS

•**Mathematics of computing → Combinatorial optimization;** Network flows; •**Applied computing → Computer-aided design;** *Operations research;*

KEYWORDS

Technical Operations Research, Booster Station, Pressure Booster System, System Synthesis, Application of Discrete Mathematics

ACM Reference format:
Jonas B. Weber and Ulf Lorenz. 2017. Optimizing Booster Stations. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3067695.3082482

978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3082482

1 INTRODUCTION

In our research, we investigate an algorithmic system design process for technical systems. The field of research we operate in is called 'Technical Operations Research' (TOR). TOR is originated from engineering and developed in cooperation with mathematicians as a part of the German Research Foundation (DFG) founded Collaborative Research Center SBF 805 'Control of Uncertainties in Load-Carrying Structures in Mechanical Engineering'. It combines technical and mathematical know-how, known from Operations Research, to generate an optimal design of technical systems regarding specific goals like energy consumption or investment costs.

The TOR approach divides the problem development process into seven steps which belong to two phases. The deciding phase and the acting phase:

DECIDING

(1) What is the function?
(2) What is my goal?
(3) How large is the playing field?

ACTING

(4) Find the optimal system!
(5) Verify!
(6) Validate!
(7) Lay Out!

TOR was applied to a various range of technical problems such as booster stations [14, 15], ventilation systems [7, 18], hydrostatic power transmission systems [11], heating circuits [17], water-conveying systems [4, 5], pump systems with multiple objectives [6], hydro power stations [16] and the design of gearboxes [3].

In the context of TOR we mostly investigate the design of fluid systems consisting of components such as pumps, pipes, valves and fittings. A special member of this class are the so-called booster stations.

Booster stations are used to supply whole buildings or higher floors with drinking water if the supply pressure of the water company is not high enough to guarantee a continuous supply for all consumers. A typical area of application are skyscrapers.

A possibility to handle problems like this is to model them as a mixed integer linear program (MILP) which is solved using a MILP solver. But usually pratically relevant problems, especially for booster stations, are too large to be solved in reasonable time even by a high quality commercial MILP solver like CPLEX [10].

In this paper, we present an approach to find 'good' system designs for large realistic instances of booster stations regarding a given objective. The goal was to develop a new methodology to solve those realistic instances in reasonable time. In this case 'solving' means either to find a provable optimal solution or to find a good solution whose quality can be evaluated by providing a strong quality criteria. We do this by addressing the problem with heuristics from both the primal and dual side using a special modeling approach. In this approach we abstract the problem as a MILP as well as a mathematical graph and use both views simultaneously. In doing so, we use both views combined with problem specific and technical knowledge to enhance the time needed to solve the optimization problem.

2 BOOSTER STATIONS

A booster station, also referred to as pressure booster system, is a network of either one type or different types of single rotary pumps. 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 each pump. In this paper, we concentrate on the third concept, booster stations with continuously variable speed control of each pump. 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 each pump 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 supplied by 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.

Booster stations are part of the so-called fluid systems. In the case of a booster station a fluid system primarily contains four component groups: 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 and the pipes of booster stations and consider the pressure reducers and valves implicitly. Hence, the presentation is simplified to an interconnection of pumps and pipes which form a connected network. The relevant physical variables are: The volume flow Q through the pumps and pipes which is comparable to the flow in the classic flow problem of theoretical computer science, the pressure increase H generated by the pumps, their power consumption P and their rotational speed n. Two of them always define an operation point in the pump characteristics. For fluid systems in general as well as for booster stations specifically certain physical laws and equations apply which are presented in the following subsections.

2.1 Continuity Equation

All fluid systems must satisfy the continuity equation: 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 says 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, ρ representing the density of the fluid, c representing the flow rate and A representing the cross-sectional area of the flow tube:

$$\dot{m} = \rho_1 \cdot c_1 \cdot A_1 = \rho_2 \cdot c_2 \cdot A_2 \tag{1}$$

If the term $c_1 \cdot A_1$ is replaced by the volume flow Q, the equation can be stated as:

$$\dot{m} = \rho_1 \cdot Q_1 = \rho_2 \cdot Q_2 \tag{2}$$

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

$$Q = Q_1 = Q_2 \tag{3}$$

This relation holds for ideal fluid systems without losses and is applicable for the system as well as for single components. It is more or less similar to the flow conservation of the classical flow problem in computer science.

2.2 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 which are not effected by external forces except for gravity a constant mathematical term, the Bernoulli energy equation, holds:

$$\frac{c_1^2}{2g} + \frac{p_1}{\rho g} + z_1 = \frac{c_2^2}{2g} + \frac{p_2}{\rho g} + z_2 = const. \tag{4}$$

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

$$p_1 + \rho \cdot g \cdot z_1 + \frac{\rho}{2} \cdot c_1^2 = p_2 + \rho \cdot g \cdot z_2 + \frac{\rho}{2} \cdot c_2^2 = const. \tag{5}$$

If a pump is used between the points 1 and 2, the pressure increase Δp_P must be considered additionally:

$$p_1 + \rho \cdot g \cdot z_1 + \frac{\rho}{2} \cdot c_1^2 + \Delta p_P = p_2 + \rho \cdot g \cdot z_2 + \frac{\rho}{2} \cdot c_2^2 \tag{6}$$

2.3 Affinity Laws

All pumps used in fluid systems have an opposite relation between their volume flow and pressure increase (H). With increasing volume flow the possible pressure increase decreases. Additionally, the power consumption (P) of pumps increases with increasing volume flow. If pumps with variable speed control are used, there is another relation. For those pumps the possible pressure increase as well as their power consumption rises with increasing speed (n) if the volume flow is held constant. For pumps with continuously variable speed control, these relations between the physical variables can be described by the so-called affinity laws: $Q \sim n$, $H \sim n^2$ and $P \sim n^3$. For a specific pump, these relations can be easily understood looking at the pump characteristics shown in figure 1.

2.4 Interconnection of modules

Modules (single pumps or whole subsystems) can be connected pairwise either in series or in parallel. The interconnection has an influence on the physical variables similar to the connection of electrical resistors. If, on one hand, modules are connected in series, the total pressure increase results as the sum of the single pressure

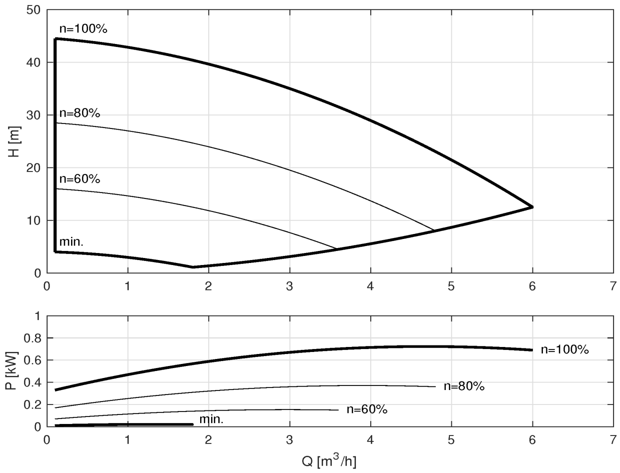

Figure 1: Typical characteristic curve of a speed controlled pump

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 OPTIMIZATION

In this section, we present the optimization problem and the corresponding mathematical model as well as our approach to solve this problem.

3.1 Problem Statement and Approach

Our goal is to implement an algorithmic system design process for instances of booster stations with a realistic character which can generate 'good' systems in reasonable time. The focus is on the time aspect as the runtime is essential for the practical usability. At first, we must define the system properties and the problem itself as well as the aimed approach to generate those systems in reasonable time.

A system is called a reasonable system if fluid (water) can enter at any component and can exit somewhere else without violating any of the physical laws of fluid mechanics.

We assume that all components and their characteristics are known in advance. This results in a given pump construction kit which only consists of speed controlled rotary pumps. For these pumps, all characteristic curves are known. The properties of the pipes and valves like pressure losses and their costs are neglected to simplify the model.

Furthermore, so-called expected load collectives are considered. We assume that the transition times and therefore also the transition costs between the load changes are negligible compared to the total costs. Hence, the model can be stated as quasi stationary. The system must be able to satisfy all loads. Each load out of the load collective is called a load scenario. A load scenario consists of three components: An assigned time slice which indicates the excepted fraction of the system's operational life this scenario occurs as well as a demanded volume flow and a demanded pressure increase.

In this paper, one system is better than another if it has lower life cycle costs. These costs are defined as the sum of the purchase costs of each component and their excepted energy costs in all load scenarios over a given period of time. Therefore, 'good' systems distinguish them from low life cycle costs compared to all other possible systems.

Generally, the problem can be abstracted in two ways. On one hand, it can be stated using linear constraints as a mixed integer linear program (MILP) as shown in section 3.2. 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 speed controlled to cover all load scenarios during system operation. On the other hand, the problem can be abstracted as a complete (mathematical) graph $G = (V, E)$, with vertices V and edges E. An edge represents a component from the construction kit and a vertex represents the possible connections between components. Furthermore, two additional vertices, the source and the sink, with corresponding pipes exist. The complete graph of the construction kit and two water plugs contains every possible system. Therefore, each system can be modeled by a sub graph of the complete graph representing the purchase decisions made for the pumps and pipes.

The common approach to generate a MILP and to solve it using a standard MILP solver like CPLEX is not suitable for the investigated problem. As the instances reach a practically relevant size, large MILPs are created which cannot be solved in reasonable time by this approach. Therefore, the special approach we used can be stated 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 algorithm to close the optimality gap between the primal and dual solution. Thus, we can obtain provable optimal solutions for the system design.

3.2 Mathematical Model

As mentioned before the problem can be modeled as a MILP. The variables and parameters used are shown in table 1 and 2. The type abbreviations stand for 'binary' ($x \in \{0, 1\}$) and 'semi-continuous' ($x \in [l, r] \cup \{0\}$).

$$min \sum_{p \in P} C_p^{pump} \cdot y_p + \sum_{s \in S} (C^{kWh} \cdot A_s \cdot \sum_{p \in P} (p_p^s \cdot T)) \tag{7}$$

Table 1: Variables of the MILP

Variable	Type	Description
y_p	b	Buy pump p
$y_{i,j}$	b	Buy pipe from pump i to pump j
x_p^s	b	Use pump p in scenario s
$x_{i,j}^s$	b	Use pipe from pump i to pump j in scenario s
n_p^s	sc	Speed of pump p in scenario s
q_p^s	sc	Volume flow through pump p in scenario s
h_p^s	sc	Pressure increase by pump p in scenario s
p_p^s	sc	Power consumption of pump p in scenario s
$q_{i,j}^s$	sc	Volume flow through pipe from pump i to pump j in scenario s
$h_{*,p}^s$	sc	Pressure before pump p in scenario s
$h_{p,*}^s$	sc	Pressure after pump p in scenario s

Table 2: Parameters of the MILP

Parameter	Description
C_p^{pump}	Purchase costs for pump p
C^{kWh}	Costs per kilowatt hour of electricity
A_s	Share of load scenario s
T	Operating life
Q_s	Volume flow to be pumped in scenario s
H_{source}^s	Inlet pressure in scenario s
H_{sink}^s	Outlet pressure in scenario s
Q^{min}, Q^{max}	Minimal and maximal volume flow
H^{min}, H^{max}	Minimal and maximal pressure
$H_p(Q,n)$	Function for pressure increase H of pump p
$P_p(Q,n)$	Function for power consumption P of pump p

$$y_{(i,j)} + y_{(j,i)} \leq 1 \tag{8}$$

$$x_p^s \leq y_p \tag{9}$$

$$x_{(i,j)}^s \leq y_{(i,j)} \tag{10}$$

$$\sum_{(i,j)\in E} x_{(i,j)}^s \geq 1 \tag{11}$$

$$\begin{aligned} q_p^s &\geq Q^{min} \cdot x_p^s & q_p^s &\leq Q^{max} \cdot x_p^s \\ h_p^s &\geq H^{min} \cdot x_p^s & h_p^s &\leq H^{max} \cdot x_p^s \\ h_{*,p}^s &\geq H^{min} \cdot x_p^s & h_{*,p}^s &\leq H^{max} \cdot x_p^s \\ h_{p,*}^s &\geq H^{min} \cdot x_p^s & h_{p,*}^s &\leq H^{max} \cdot x_p^s \end{aligned} \tag{12}$$

$$q_{(i,j)}^s \geq Q^{min} \cdot x_{(i,j)}^s \qquad q_{(i,j)}^s \leq Q^{max} \cdot x_{(i,j)}^s \tag{13}$$

$$Q_s = \sum_{(source,j)\in E} q_{(source,j)}^s \qquad Q_s = \sum_{(i,sink)\in E} q_{(i,sink)}^s \tag{14}$$

$$q_p^s = \sum_{(p,j)\in E} q_{(p,j)}^s \qquad q_p^s = \sum_{(i,p)\in E} q_{(i,p)}^s \tag{15}$$

$$h_{*,p}^s + h_p^s = h_{p,*}^s \tag{16}$$

$$\pm(H_{source}^s - H_{sink}^s) \leq H^{max} \cdot (1 - x_{source,sink}^s) \tag{17}$$

$$\begin{aligned} \pm(H_{source}^s - h_{*,j}^s) &\leq H^{max} \cdot (1 - x_{source,j}^s) \\ \pm(h_{i,*}^s - H_{sink}^s) &\leq H^{max} \cdot (1 - x_{i,sink}^s) \end{aligned} \tag{18}$$

$$\pm(h_{i,*}^s - h_{*,j}^s) \leq H^{max} \cdot (1 - x_{i,j}^s) \tag{19}$$

The objective function (7) contains two parts, the investment costs for the system components and their operating costs. The investment costs include the purchase costs of the installed pumps. Pipes have no related costs in this model. The energy costs result from the power consumption of each active pump in the corresponding load scenario multiplied by the costs per kilowatt hour electricity.

Furthermore, constraints exist which ensure that the physical laws and the general properties of fluid systems are met:

- The purchase decisions are considered as follows: (8) For two components (including the source and the sink) there can be at most one pipe connecting them. (9) A pump can only be used to satisfy a load scenario if it is installed. (10) The same applies for pipes. (11) In each load scenario at least one pipe must be used to ensure the flow conservation.
- The operational bounds must be observed: (12) If a pump is used, its volume flow, pressure increase and adjacent pressures must be reasonable. Otherwise they vanish. (13) If a pipe is used, its volume flow must be reasonable. Otherwise it vanishes.
- All valid systems must satisfy the continuity equation: (14) The flow rates at the source and sink must be equal. (15) The flow rate through a pipe must be conserved.
- Bernoulli's equation must apply: (16) The pressure increase of a pump increases the adjacent system pressure behind its outlet. (17) The source and the sink can only be connected if no pressure increase is needed. (18) If a pump is connected to the source or the sink, the pressure propagates through the pipe. (19) If two pumps are interconnected, the pressure propagates through the pipe.
- Additionally, the operation point of each pump must lie on its characteristic curve. This can be achieved by generating a suitable number of points from the empirically known $H_p(Q,n)$ and $P_p(Q,n)$ functions as base points and forcing the respective variables on the linearized curves defined by these points. The used linearization techniques follow [19].

4 FINDING GOOD PRIMAL SOLUTIONS USING SIMULATED ANNEALING

The implemented Simulated Annealing algorithm follows [8] with some modifications: Previous calculations are saved and a penalty term for invalid system topologies is implemented. The algorithm is used to find good topologies for the first stage of the two-staged optimization problem (topology problem) as described in section 3.1.

After generating a topology, the binary first stage variables are fixed in the MILP. Afterwards the second stage (operation problem) is solved optimally for the chosen topology regarding the different load scenarios using CPLEX. For the topology decision only series-parallel networks as defined in [13] are considered to ensure that only technically applicable systems are generated.

4.1 Neighborhood Function

The problem specific neighborhood function needed for Simulated Annealing consists of four single neighborhoods: The replace ($N_{Replace}$), the swap (N_{Swap}), the add (N_{Add}) and the delete neighborhood (N_{Delete}):

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

$N_{Replace}$: A pump p_i of the set of bought pumps is selected randomly and replaced by a pump p_j from the set of unbought pumps. The previous predecessors and successors of p_i are the new predecessors and successors of pj. This neighborhood can only be created if the network consists of at least one pump and there is at least one unbought pump.

N_{Swap}: Two different pumps p_i and p_j of the set of bought pumps are selected randomly. p_i and p_j swap positions in the network. The previous predecessor and successors of p_i are the new predecessors and successor of p_j and vice versa. This neighborhood can only be created if the network of bought pumps consists at least of two pumps.

N_{Add} : A pump p_i of the set of unbought pumps 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 pump out of the set of bought pumps, the source or the sink is selected. If the source or the sink is selected, p_i is connected in series behind the source and before the sink, respectively. If a pump p_j is selected, p_i is connected before or behind p_j. The source, the sink or p_j becomes the new predecessor and the new successor of p_i, respectively and p_i adapts their previous successors and predecessors, respectively.

If p_i is connected in parallel, a pump p_j of the set of bought pumps 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 pumps consists of at least one pump and in the case of a parallel connection if the set of bought pumps consists at least of one pump.

N_{Delete}: A pump p_i of the set of bought pumps 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 and successor of p_i, respectively is selected randomly and 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 pump in the set of bought pumps.

4.2 Generating a starting solution

To generate a starting solution a simple heuristic is used which is based on the add neighborhood to obtain valid solutions. First, a minimal network including only a source and a sink is considered. If this network is already a valid solution, it is accepted as the starting solution. Otherwise a pump is added to the network. If the set of unbought pumps is empty and the solution is still not valid, the whole network will be deleted and the procedure starts again with a minimal network until a valid topology is found.

4.3 Penalty term

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, it is valued with the double costs of the starting solution. This factor 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 higher than the costs of all valid solutions.

4.4 Saving previous solutions

The critical step for the runtime of the algorithm are the calculations for the optimal operation mode for the found topologies performed by CPLEX. 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 calculated by CPLEX and added to the list. If the list reaches the defined maximum size, the last entry will be deleted so that new solutions can be stored.

5 GENERATING TIGHT LOWER BOUNDS

A simple LP-relaxation, i.e. dropping the integrality constraints, is not suitable to deliver tight lower bounds for realistic instances. For that reason, an approach is presented which uses problem specific knowledge to obtain tight lower bounds. The pseudo code is presented at the end of the section.

In the first step the original problem is relaxed by disabling the coupling constraints which connect the buy- (y_p, $y_{i,j}$) and the operation-variables (x^s_p, $x^s_{i,j}$), cf. (9) and (10), of the pumps and pipes for all load scenarios, i. e. only bought components can be used to satisfy the load scenarios:

$$x^s_p \leq y_p \tag{20}$$

$$x^s_{i,j} \leq y_{i,j} \tag{21}$$

Afterwards the problem is split into $|S|$-many sub problems, one for every load scenario. The remaining buy-variables in all sub problems are substituted by the suitable operation-variables. Afterwards, each of the $|S|$ sub problems is split again into two sub sub problems. These problems represent the optimization task for minimizing the energy costs and the investment costs, respectively for one single load scenario. The new objective functions for the sub sub problems are:

$$\min(C^{kwh} \cdot \sum_{p \in P} A_s \cdot p^s_p \cdot T) \tag{22}$$

$$\min \sum_{p \in P} C^{pump}_p \cdot x^s_p \tag{23}$$

For each of this $2|S|$ problems the optimal solution is calculated by CPLEX. A lower bound arises out of the sum of the energy costs and the maximum of all investment costs for every load scenario:

$$\underline{z} = \sum_{s \in S} (C^{kwh} \cdot \sum_{p \in P} A_s \cdot p_p^s \cdot T) + \max_{s \in S} (\sum_{p \in P} C_p^{pump} \cdot x_p^s) \quad (24)$$

This is obviously a valid way to achieve 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 every load scenario, the investment costs cannot be lower than the maximum of the investment costs for every decoupled load scenario because this is the configuration with minimal costs to serve the 'most challenging' load scenario.

```
Let P_0 be the original problem
Let f be the objective function
Disable coupling constraints for P_0
Split P_0 into P_1, ..., P_|S| one for each load scenario s
Lower Bound LB ← 0
Energy-Costs EC ← 0
Invest-Costs IC ← 0
for each k ∈ {1...|S|} do
  Replace buy- with operation-variables for P_k
  Generate topology problem T_k for P_k
  Generate operation problem C_k for P_k
  EC ← EC + f(T_k)
  if IC < f(C_k) then
    IC ← f(C_k)
  end if
end for
LB ← EC + IC
```

6 CLOSING THE GAP USING BRANCH-AND-BOUND

Based on the basic Branch-and-Bound algorithm, as described in [9], a method using problem specific knowledge to obtain optimal solutions is presented. The Simulated Annealing algorithm is used to obtain a good starting solution and by relaxing the problem we generate lower bounds (bounding function). During the procedure, the disabled coupling constraints were restored successively.

The calculation and selection of the nodes are based on the eager strategy combined with the best-first-search (BeFS) strategy for the Branch-and-Bound algorithm.

If a new valid solution is found, it is checked whether it is a new upper bound. In a valid solution only those pumps are used for operation which are also bought and therefore their purchase costs are part of the investment costs of the system.

Unexplored (active) nodes are branched based on the following branching rule. For these nodes so-called conflicting pumps exist. These are pumps which are used for operation but are not bought and their costs are not part of the investment costs. The branching rule for the active nodes is defined as follows: A pump out of the set of conflicting pumps is selected randomly. For the first sub problem, the selected conflicting pump is not bought and therefore not used for operation. Hence, the values of its binary buy-variable and operation-variables are fixed to 0. For the second sub problem, the selected conflicting pump is bought and can be used for operation. Hence, its buy-variable is fixed to 1. The operation-variables are not effected in this case because the pump might or might not be used for operation.

After exploring all active nodes, the current upper bound is the provable optimal solution of the original problem.

```
Calculate global Upper Bound UB
Let n_0 be the original problem
Calculate Lower Bound of n_0 LB(n_0)
Let RELAX(n_0) be the optimal solution for relaxed n_0
ActiveNodes AN ← AN ∪ {n_0}
while AN ≠ ∅ do
  Take n_i ∈ AN with LB(n_i) ≤ LB(n_j) ∀ n_j ∈ AN
  Take a pump p_p from n_i which is used but not bought
  Split n_i into two child nodes cn_0, cn_1
  Fix buy-variable of p_p to 0 for cn_0 and to 1 for cn_1
  for each cn_k ∈ {cn_0, cn_1} do
    AN ← AN ∪ {cn_k}
    Calculate LB(cn_k)
    if LB(cn_k) ≥ UB then
      AN ← AN\{cn_k}
    else if RELAX(cn_k) is a valid solution for n_0
      AN ← AN\{cn_k}
      UB ← LB(cn_k)
    end if
  end for
  AN ← AN\{n_i}
end while
Optimal Solution OS ← UB
```

7 TEST CASES

To test the developed approach, test cases with a realistic character were designed. In all test cases a booster station was used which was directly connected to the water supply. Hence, the required pressure increase supplied by the booster station was the total pressure increase needed minus the supply pressure. If necessary, normal zones and reducing valves were used to avoid overpressure. All calculations are based on the DIN standards 1988-3 and 1988-5 described in [1, 2, 12]. Furthermore, the usage period was set to 10 years with assumed mean energy costs of 0.3 Euro per kWh.

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

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

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

7.1 Buildings

Two different fictional buildings are used. Both are skyscrapers 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.

7.2 Usage

The buildings are either used as a so-called hospital (H), residential (R) or 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 were distinguished.

7.3 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.

7.4 Available pump kit

For each test case one of two disjoint pump kits (including five pumps each) is available. All of them are speed controlled single rotary pumps and taken from the *Wilo Economy MHIE* model series. The first kit includes 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.

8 RESULTS

All calculations were performed on a MacBook Early 2015 with a 2.7 GHz Intel Core $i5$ and 8 GB 1867 MHz DDR3 memory.

For the Simulated Annealing a cooling schedule with an exponential cooling function, $T(t) = T_0 \cdot \alpha^t$, was used. For the parameter α we chose a value of 0.9. The initial temperature T_0 was set to 10, 000 and 100 iterations were performed at each temperature. The algorithm stopped when T reached the threshold T_{stop} of 1.

8.1 Solutions

In this section, the quality of the solutions found by the Simulated Annealing algorithm and the lower bounds is presented.

8.1.1 Simulated Annealing. Table 3 shows the objective values of the best solutions found by Simulated Annealing (z_{SA}). Also, the optimality gap ($gap_{\underline{z}}$) between the objective value and the lower bound ($\underline{z}$) is presented as well as the real gap (gap_{z^*}) between the found primal solution and the optimal solution (z^*) obtained by Branch-and-Bound. The mean optimality gap for all test cases 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 actual mean deviation was 0.69% with a standard deviation of 1.08%. However, if the optimal solution was not found, the mean deviation was 1.65% with a standard deviation of 1.1%.

8.1.2 Lower Bounds. Furthermore, the lower bounds were compared to the optimal solution. The mean deviation between the lower bounds and the optimal solution was 7.45% with a standard deviation of 5.23%. The maximum deviation was 19.92% while the minimum was just 0.54%.

Table 3: Simulated Annealing, lower bounds, optimal solution - solutions and gaps

Test case	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	10069.90	10015.90	0.54%	10069.90	0.00%
B15_O_D_2	9115.01	8 116.26	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	24601.00	24004.30	2.49%	24518.10	0.34%
B10_R_D_2	23516.20	22215.40	5.86%	23516.20	0.00%
B10_R_C_1	12711.90	12334.20	3.06%	12711.90	0.00%
B10_R_C_2	13968.60	12157.00	14.90%	13968.60	0.00%
B15_R_D_1	29360.00	27570.20	6.49%	29319.40	0.14%
B15_R_D_2	28407.00	24457.10	16.15%	28407.00	0.00%
B15_R_C_1	20505.40	19750.90	3.82%	20486.40	0.09%
B15_R_C_2	19909.10	17315.50	14.98%	19909.10	0.00%
B10_H_D_1	25068.10	23912.40	4.83%	24607.60	1.87%
B10_H_D_2	23704.70	22127.50	7.13%	23287.70	1.79%
B10_H_C_1	13315.10	12659.00	5.18%	13070.80	1.87%
B10_H_C_2	13946.80	11168.80	24.87%	13946.80	0.00%
B15_H_D_1	27936.80	26651.30	4.82%	27210.70	2.67%
B15_H_D_2	28186.40	25001.00	12.74%	27377.30	2.96%
B15_H_C_1	21380.80	18942.90	12.87%	20974.60	1.94%
B15_H_C_2	21649.00	17637.20	22.75%	21041.10	2.89%

8.2 Runtime

In this section, the runtime of all three procedures is presented. It should be noted that the runtime of the Branch-and-Bound algorithm includes the runtime of Simulated Annealing as it generates the starting solution for the Branch-and-Bound.

8.2.1 Simulated Annealing. The Simulated Annealing algorithm needed a mean of 475.37 seconds to terminate. High deviations occurred. The maximum runtime was 2 411.58 seconds, while the minimum runtime was only 85.34 seconds. This results from the fact that *CPLEX* needs much more time to solve the operation problem if the created neighborhood is large in terms of many bought components.

8.2.2 Lower Bounds. Generating lower bounds took 660.97 seconds in mean. In most cases this was slightly more time than the Simulated Annealing algorithm needed to find good solutions. The highest runtime was 1 582.3 seconds while the lowest runtime was only 207.86 seconds.

8.2.3 Branch-and-Bound. The mean runtime for generating optimal solutions was 9 968.8 seconds. The maximum runtime was 21 472.9 seconds and the minimum runtime only 4 148.12 seconds. If the initial upper bound found by the Simulated Annealing was already the optimal solution, the mean runtime was 8 804.11 seconds and therefore 31.75% faster than in the opposite case where the mean runtime was 11 599.36 seconds.

9 CONCLUSIONS

In this paper, we presented an approach to optimize booster stations in reasonable time. Using primal and dual heuristics, we are able to find good primal and dual solutions for the system design in an appropriate amount of time even for relatively large instances. Furthermore, we presented a Branch-and-Bound algorithm which combines both heuristics to obtain provable optimal system designs regarding a given objective. The presented approach was validated for test cases with practically relevant size and different demands for the pressure increase at time-variant flow rates with four to five different load scenarios. The runtime in the range of minutes to one hour for the primal and dual heuristics as well as the range of one to six hours for obtaining optimal solutions was quite reasonable in both cases and of practical relevance. In further research, we plan to test our approach with other primal heuristics such as Genetic Algorithms or Tabu Search. This will allow us to investigate their influence on the runtime. Additionally, we plan to expand our approach to other technical applications.

REFERENCES

[1] DIN 1988-300:2012-05. 2012. *Technische Regeln für Trinkwasser-Installationen - Teil 3: Ermittlung der Rohrdurchmesser.* Technische Regel des DVGW, Berlin.

[2] DIN 1988-500:2010-10. 2010. *Technische Regeln für Trinkwasser-Installationen - Teil 5: Druckerhöhungsanlagen mit drehzahlgeregelten Pumpen.* Technische Regel des DVGW, Berlin.

[3] Lena C. Altherr, B. Dörig, Thorsten Ederer, P. F. Pelz, Marc E. Pfetsch, and Jan Wolf. 2017. A Mixed-Integer Nonlinear Program for the Design of Gearboxes. *Operations Research Proceedings 2016: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR)* (2017).

[4] Lena C. Altherr, Thorsten Ederer, Ulf Lorenz, P. F. Pelz, and Philipp Pöttgen. 2016. Designing a Feedback Control System via Mixed-Integer Programming. *Operations Research Proceedings 2014: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR)* XIV, 1 (2016).

[5] Lena C. Altherr, Thorsten Ederer, Ulf Lorenz, P. F. Pelz, and Philipp Pöttgen. 2016. Experimental Validation of an Enhanced System Synthesis Approach. *Operations Research Proceedings 2014: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR)* XIV, 1 (2016).

[6] Lena C. Altherr, Thorsten Ederer, Philipp Pöttgen, Ulf Lorenz, and P. F. Pelz. 2015. Multicriterial Optimization of Technical Systems Considering Multiple Load and Availability Scenarios. *Applied Mechanics and Materials* 807 (November 2015), 247–256.

[7] Lena C. Altherr, Thorsten Ederer, Christian Schänzle, Ulf Lorenz, and P. F. Pelz. 2015. Algorithmic System Design Using Scaling and Affinity Laws [accepted]. *Operations Research Proceedings 2015: Selected Papers of the International Conference on Operations Research* (November 2015). akzeptiert (unveröffentlicht).

[8] Ilhem Boussad, Julien Lepagnot, and Patrick Siarry. 2013. A survey on optimization metaheuristics. *Information Sciences* 237 (2013), 82–117. Prediction, Control and Diagnosis using Advanced Neural Computations.

[9] J. Clausen. 1999. *Branch and Bound Algorithms - Principles and Examples.* Research Report. Departement nof Computer Science, University of Copenhagen, Denmark.

[10] IBM ILOG CPLEX. *https://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/*. Accessed: 2017-03-16.

[11] B. Dörig, Thorsten Ederer, P. Hedrich, U. Lorenz, P. F. Pelz, and Philipp Pöttgen. 2014. Technical Operations Research (TOR) Exemplified by a Hydrostatic Power Transmission System. *9.IFK ? Proceedings Vol.1* (2014).

[12] KSB. 2012. *Planungshinweise Druckerhöhungsanalgen.* Number 5 in KSB Know-How. KSB Aktiengesellschaft, Frankenthal.

[13] P. A. McMahon. 1890. Yoke-Chains and Multipartite Compositions in Connexion with the Analytical Forms called "trees". In *Proceedings of the London Mathematical Society*, Vol. 1. 330–346.

[14] P. Pelz and U. Lorenz. 2013. Besser geht's nicht! TOR plant das energetisch optimale Fluidsystem. *delta p* Ausgabe 3 (2013). ERP Spezial.

[15] P. F. Pelz, U. Lorenz, T. Ederer, S. Lang, and G. Ludwig. 2012. Designing Pump Systems by Discrete Mathematical Topology Optimization: The Artificial Fluid Systems Designer (AFSD). *International Rotating Equipment Conference* (2012).

[16] P. F. Pelz, U. Lorenz, T. Ederer, M. Metzler, and P. Pöttgen. 2014. Global System Optimization and Scaling for Turbo Systems and Machines. *Proceedings of 15th International Symposium on Transport Phenomena and Dynamics of Rotating Machinery* (Februar 2014), FR203.

[17] Philipp Poettgen, Thorsten Ederer, Lena Altherr, Ulf Lorenz, and Peter F. Pelz. 2016. Examination and Optimization of a Heating Circuit for Energy-Efficient Buildings. *Energy Technology* 4, 1 (2016), 136–144.

[18] Christian Schänzle, Lena C. Altherr, Thorsten Ederer, U. Lorenz, and P. F. Pelz. 2015. As Good As It Can Be - Ventilation System Design By A Combined Scaling And Discrete Optimization Method. *Proceedings of FAN 2015, Lyon (France), 15 - 17 April 2015* (April 2015).

[19] J. P. Vielma, S. Ahmed, and G. Nemhauser. 2010. Mixed-Integer Models for Non-seperable Piecewise-Linear Optimization: Unifying Framework and Extensions. *Operations Research* 58, 2 (2010).