Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Essays of applied mathematical optimization in logistics

Cumulative thesis to obtain the academic degree of
doctor rerum oeconomicarum (Dr. rer. oec.)
according to the doctoral degree regulations 2014 at the
Faculty of Business Administration (Hamburg Business School)

submitted by Justus Bonz
born on 31.08.1989 in Hamburg

Hamburg, February 16, 2021

Chairperson:	Prof. Dr. Wolfgang Drobetz
First Examiner:	Prof. Dr. Knut Haase
Second Examiner:	Prof. Dr. Guido Voigt
Date of Disputation:	2021-01-22

Contents

1 Synopsis

1.1 Introduction

To keep the research as close as possible to the industry my co-author and I partnered up with two companies and a governmental organization. All decision problems addressed in the articles emerged from their logistical operations and the results have been or are currently being implemented. Every organization provided detailed information about their processes relevant for the research question alongside rich data sets.

The *HSL Logistik GmbH* is family-operated railroad company for cargo trains based in Hamburg. Their energy consumption is one of their biggest cost factors. The implementation of our approach optimizing their energy management is currently in development for a use in production. Combining their experience of the cargo business with the state of the art in the academic literature helped us formulating a novel problem in the energy sector.

The *Feldschlösschen Getränke AG* is our second industrial partner with new questions for a well-studied decision problem. From their daily business managing their warehouse questions with unknown answers have been formulated. With the help of a rich data set containing all order picking requests over two years I analyzed in greater depth the problem of storage location assignment. Based on the results of my answers they already rearranged their pick storage warehouse.

Lastly, the cooperation with the *Ministry of Municipal and Rural Affairs (MOMRA)* already lead to research articles like *A Pilgrim Scheduling Approach to Increase Safety During the Hajj* which optimized the schedules for the pilgrims during the pilgrimage called Hajj. The implementation of those schedules was accompanied by schedule contradictions. Identifying schedule compliant and non-compliant pilgrims helped increasing the safety of all pilgrims. For the operation a small workforce manually observed the pilgrims and noted the schedule compliance in a smartphone application. The custom developed smartphone application also displayed the route for the members of the workforce through the campsite of the pilgrims.

1.2 Research Contribution

All research articles for this dissertation are listed in Table 1. In the first article we formulate a non-linear model to minimize the energy cost when operating a fleet of electrical locomotives. The cost function reflects the current price model of the *DB Netze* responsible for the railway infrastructure. Beside the cost for the total work done *DB Netze* prices the peak demand which is the maximum power demand that has occurred over 15-minute intervals during the billing period. The development of a novel model which coordinates the locomotives to minimize the electrical cost is the main research contribution of this article. A field study evaluated the mathematical framework and compared the calculated and the billed energy demand. Furthermore, a tailored heuristic and a genetic algorithm ensures short computational times necessary for an online application. Both heuristic approach utilizes artificial velocity limits which elegantly simplify finding feasible solutions. We showed that coordinating the fleet of electric cargo trains reduces the energy cost. The findings of this article can be applied to other use cases when the calculation of the electrical work done is adjusted accordingly.

The second article extends the current approach of formulating the storage location assignment problem in a warehouse. Picking customer orders take less effort with a sophisticated assignment of stock-keeping units to storage locations. *Feldschlösschen Getränke AG* already had an existing storage location assignment and pondered about changing it due to changes in the assortment and changing customer behaviors. Even though rearranging the warehouse takes effort itself, it may benefit the order picking process. The currently known approach in the literature has been extended and replenishments of the stock-keeping units are considered. The quadratic multiple knapsack problem has been applied for the first time to solve the linear mixed integer problem. Another approach is formulated which reduces the instance size to keep the computational time for the optimal formulation in check. The idea is to generate a small set of artificial orders with similar properties like the original input data set. I implement a variant of the 2-opt algorithm to improve the solutions of all approaches. Furthermore, I formulate a recommendation on the frequency of rearrangement based on the results of the real picking orders from two years. I conducted another study to gain insights when rearranging only a small subset of the assortment.

The third article evaluates four known solution approaches for multi objective problems on the application of observing the schedule compliance of pilgrims during the Hajj. The four approaches are based on the data set from the Hajj in 2015 and their advantages and disadvantages are analyzed. Additionally, performance indicators measuring multiple properties of the three approaches are calculated and compared for all approaches.

Table 1: List of my three research articles.

Article:	Reducing the combined peak energy demand for a fleet of electrical locomotives
Status:	Recommended resubmission in OR Spectrum
Authors:	Justus Bonz, Knut Haase
Article:	The quadratic multiple knapsack problem assigning storage locations in a warehouse
Status:	Working paper
Authors:	Justus Bonz, Knut Haase
Article:	Application of a multi-objective multi traveling salesperson problem with time windows
Status:	Minor revision in Public Transport
Authors:	Justus Bonz

1.3 Declaration on Co-Authorships

The third article listed in Table 1 is a single-authored article without any co-authorship to declare. The following list is used to declare my personal responsibility for multiple aspects of the remaining two articles:

1. Minor contribution
2. Major contribution
3. Exclusive contribution

Table 2 lists my contributions for the first and Table 3 for the second article.

Table 2: My contribution to the first article.

Aspect	Contribution
Concept	major
Physical framework	exclusive
Field test	minor
Non-linear model	major
Genetic algorithm	exclusive
Tailored heuristic	exclusive
Implementation of the algorithms	exclusive
Computational study	exclusive
Manuscript	major

Table 3: My contribution to the second article.

Aspect	Contribution
Concept	exclusive
Knapsack approach	exclusive
2-opt algorithm	exclusive
Reduce data set size approach	major
Computational study	exclusive
Manuscript	major

2 Research Articles

2.1 Reducing the combined peak energy demand for a fleet of electrical locomotives

Reducing the combined peak energy demand for a fleet of electrical locomotives

Justus Bonz, Knut Haase

October 1, 2020

Abstract

We consider a German railroad company with a high energy demand. Energy providers incentivize a low peak demand of their customers by pricing the peak demand. In the electricity networks for railroads in Germany the peak demand is priced with a constant factor. Coordinating the fleet of locomotives of the railroad company can decrease the peak demand and therefore their energy costs. A real-time approach is presented and an algebraic non-linear model for generating train movements, which minimizes the energy cost. The input of the model are train schedules including train properties, track topologies and velocity limits along the track. The output are train speed profiles for all trains in operation. To the best of our knowledge such a model determining the velocity of multiple trains while calculating the physical work done to minimize the electrical peak demand does not exist yet. To evaluate the mathematical framework and therefore the fundamental equations of a model a field study has been conducted to compare the measured and calculated amount of work done. A tailored heuristic and a genetic algorithm are developed to find good solutions in a reasonable time. Both heuristic approaches are based on time-based velocity limits to simplify the control over the movement of each train. This innovative approach solves the difficulty of computing feasible velocity-time graphs for each train while ensuring the arrival after a certain distance at the scheduled destination. On small instances, the non-linear model finds on average the best solutions with a computational time limit of 5 minutes. The non-linear model struggles to find a feasible solution on bigger instances where the tailored heuristic finds the best solutions. Combining the solution obtained with the tailored heuristic with the genetic algorithm improves the solution. The cost savings of coordinating all trains instead of optimizing each train individually is on average 3.8% when operating two trains, over 20% when operation five trains and around 30% for seven or more trains. These results advice to take the peak demand into account when optimizing the energy consumption.

Keywords: OR in energy; Non-linear programming; Electrical cargo trains; Train controlling; Train dispatching; Genetic algorithms

1 Introduction

Energy-intensive companies are billed for their total consumption of electric work as well as for their electric peak demand. The fee for the electric work is

1

2.1 Reducing the combined peak energy demand for a fleet of electrical locomotives

based on the total amount of consumed energy during the billing period. To incentivize a low variance of the energy consumption of their customers, energy providers typically charge a separate fee. Time-dependent energy prices are one approach for energy providers to shift the energy demand to preferable time periods. Another strategy prices the peak demand of the billing period. DB Energie GmbH is a subsidiary of Deutsche Bahn AG, whose field of activity is the generation, procurement and provision of energy and they operate the traction current. They divide the year into 15 minute periods and calculate the average load profile of their customers.

Railway companies operating electrical locomotives in Germany can decrease their electricity cost in two ways. They can either reduce the amount of consumed energy W or their peak demand $P^{\max}$ in the billing period. The billing period is usually a year. The cost for the electric energy for all companies operating trains in Germany in 2019 is (DB Netze, 2019):

$$\text{cost} = \begin{cases} 96.18\frac{€}{\text{kW}} \cdot P^{\max} + 0.00184\frac{€}{\text{kWh}} \cdot W & \text{if} \quad \frac{W}{P^{\max}} > 2500\text{h} \\ 0.0569\frac{€}{\text{kWh}} \cdot W & \text{otherwise} \end{cases} \tag{1}$$

See Figure 1 for the visualization of the electricity cost depending on the peak demand $P^{\max}$ containing three examples of the total work W.

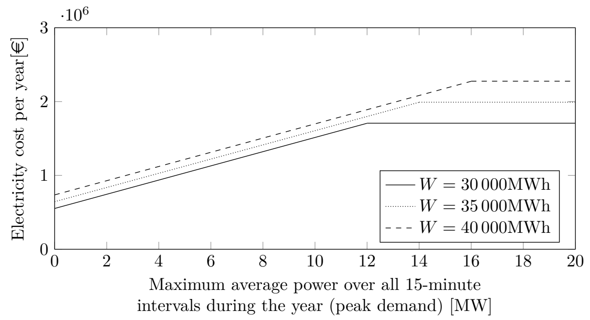


Figure 1: Electricity cost for a company depending on peak demand for different amounts of electric work.

The approach described in this paper minimize the electricity cost of a railway company. Railway companies schedule their trains to move from origins to destinations in a given time interval. This interval leaves room for movement adjustments. Especially cargo trains have some flexibilities regarding their schedules. When sharing the same rails, passenger trains have a higher priority on German railroads compared to cargo trains. Coordinating the trains to balance the electric load can lower the peak demand. Currently such a coordination of simultaneously moving trains is not in use.

Section 2 discusses approaches in the literature which address lowering the electrical peak demand. We focus on literature operating with a given time table

2

for the trains. In the following Section 3 we introduce the online approach to coordinate the trains, establish the fundamental physical formulas and formulate a non-linear mathematical optimization model. This Section also contains the field study evaluating the calculating of the work done. Afterwards, the Section 4 contains the explanation of a tailored heuristic and a genetic algorithm to reduce the computation time for finding a reasonably good solution. In Section 5 we present the results of a computational study based on generated instances. Finally, the findings are discussed in Section 6 and further research worth studying is mentioned.

2 Literature review

This paper is related to literature that addresses the electric load profile of various operations as well as literature about coordinating trains. A myriad of papers in many research areas are somehow related to our problem without addressing the same problem setting. This literature review lists relevant papers from various research topics to show the broad interest in the peak demand optimization. Firstly, literature about shifting electrical load is looked into before analyzing the literature about train operations. The objective concerning the load profile strongly depends on the context of the operation. Some research addresses the stability of circuits when balancing the load, while others try to minimize the cost for the peak demand. The price model for the peak demand is either a fixed price for the peak demand or time-dependent energy prices.

Oldewurtel et al., 2010 wrote a paper using time-dependent energy prices. Even though this price model is different from the price model relevant for this paper, it still shows that shifting electrical load can reduce the corresponding costs. They look at the climate control of buildings proposing that the air condition can get work done, while the energy price is low. This is possible, because the climate control has some degrees of freedom while keeping the temperature and CO_2 concentration in a given interval. They iteratively linearize the constraints of their non-convex problem in a form of Sequential Linear Programming. Even with an increase in the total energy consumption the electricity cost can be reduced, because of the shift in electric load.

Stabilizing the load on electrical circuit is the objective of Shao et al., 2012. This approach does not minimize the cost for electric energy but it shows another use case for minimizing the electric peak demand. Charging too many electrical cars on the same circuit simultaneously can overload the infrastructure. They propose a demand response strategy to balance the load in residential distribution circuits. They calculate a demand limit for the electric work of each household along the same circuit, while considering the customer preferences.

Manufacturing related papers also target the electric peak demand. Those approaches document the benefit of minimizing the electric peak demand, even though those models cannot be applied to the core problem of this paper. Wichmann et al., 2018 introduce an energy-oriented general lot-sizing and scheduling problem while assuming time-dependent energy prices. They formulate a linear mixed integer program, which can reduce the energy cost by 9.69% and the total cost by 1.04% compared to the classical planning approaches. Higher variations in the time-dependent energy prices lead to higher cost savings. Another idea is to fit the production to a predefined energy curve as in Nolde and Morari,

2010. Here a linear mixed integer program finds a production plan as close as possible to the predefined energy curve. The "Just-for-Peak" inventory buffers in Fernandez et al., 2013 maintain the same throughput of a manufacturing systems, while reducing the load in peak periods. Those buffers are filled while minimizing the sum of holding costs for the inventory. A non-linear integer program solves the problem and reduced the power demand in peak periods by 20%.

Minimizing the energy consumption of trains is the topic of papers like Scheepmaker et al., 2017, Tang et al., 2020, Liu and Golovitcher, 2003 or Miyatake and Ko, 2010. Those papers do address the total energy consumption and not the load profile or peak demand, even if they consider trackside traffic signals Wang et al., 2014. A. Albrecht et al., 2015 looked into coordinating two trains driving on the same track in the same direction to minimize the energy consumption without taking the peak demand into account. Similar, A. Albrecht et al., 2018 finds driving strategies for two trains in a similar setting to minimize the total energy for both trains. We address the peak demand of a fleet of trains in control of a single entity, which is not necessarily in control of two consecutive trains.

There are also technical approaches to reduce the electric peak demand. Ciccarelli et al., 2012 describe supercapacitors to store energy gained by braking on board and using it to accelerate. Those supercapacitors can reduce the total energy consumption of rapid transit trains by 12% compared to trains with braking resistors, by recovering energy from braking. The Swiss Federal Railways company had another approach to reduce their peak demand by turning off the heating for their railway switches during periods of high energy consumption.

Scheduling passenger trains to reduce the peak demand is the topic of other papers. Gu et al., 2013 address the problem of high peak demands for moving block systems, where the passenger trains drive very close to each other. They distribute the starting times of the trains to prevent too many trains from accelerating at the same time. A non-linear program can reduce the peak power demands by 20% to 40% without delaying the arrival of the trains. They compared their solution with the traditional approach of limiting the acceleration of the trains in a block. Adding dwell times at the stations is the approach of T. Albrecht, 2010 to reduce the peak demand. He proposes a genetic algorithm to solve the problem.

Bärmann et al., 2017 address the introduced problem from another perspective. They show how the timetables for trains can be modified to reduce the peak demand of all trains in the network. The running times of all times are fixed and cannot be changed in their presented model, but they modify the dwell times of passenger trains at the stations. Our formulation targets the train operator and optimizes the running times of the trains while adhering to the given and therefore fixed timetables.

Most of the literature is based on time-dependent energy prices or a dedicated price for the peak demand. In this paper the second price model is applied. We did not find any approach in the literature, which adjusts each power consumption of a set of independent energy consumers to lower the peak demand. No approach is applicable to a company whose objective is to reduce the energy cost by coordinating their fleet of electrical locomotives. This papers targets the acceleration and therefore velocity of cargo trains to reduce the combined energy cost of the total energy consumption and the peak demand.

3 Algebraic model

3.1 Kinematic background and field study

This section establishes the kinematic background required for this problem. It explains all formulas used in the solution approaches while laying the ground for the assumptions made in Section 3.3 which greatly decrease the complexity of the problem. Furthermore, this section closes with a field study which measures the accuracy of the introduced formulas measuring the work done by a train.

During their operation the trains move from an origin to a destination. The acceleration changes the velocity and therefore the traveled distance. This acceleration requires energy, which must be paid for.

The acceleration $a(t)$ can change over time, while $a(t) < 0$ decelerates and $a(t) > 0$ accelerates the train. The velocity $v(t)$ and the traveled distance $s(t)$ result from the acceleration. Let $a(0) = v(0) = s(0)$, then the velocity and distance can be calculated as follows:

$$v(t') = \int_0^{t'} a(t)dt \tag{2}$$

$$s(t') = \int_0^{t'} v(t)dt \tag{3}$$

The acceleration requires an acceleration force $F^{\text{ACC}}(t)$, while other forces also apply. Those forces affect the train movement and need to be compensated for. For example, the rolling resistance $F^{\text{ROLL}}(t)$, the downhill force $F^{\text{SLOPE}}(t)$, and the air resistance have an effect. The air resistance depends on the square of the velocity, while the other forces depend on the mass of the train $m^{\text{Locomotive}} + m^{\text{Cargo}}$. To reduce the complexity of the model, the air resistance is not used for further calculation, because the mass of a cargo train is very high compared to the velocities. The forces are calculated as follows:

$$F^{\text{ACC}}(t) = (e \cdot m^{\text{Locomotive}} + m^{\text{Cargo}}) \cdot a(t) \tag{4}$$

$$F^{\text{ROLL}}(t) = (m^{\text{Locomotive}} + m^{\text{Cargo}}) \cdot f^{\text{R}} \cdot g \cdot \cos(\alpha(t)) \tag{5}$$

$$F^{\text{SLOPE}}(t) = (m^{\text{Locomotive}} + m^{\text{Cargo}}) \cdot g \cdot \sin(\alpha(t)) \tag{6}$$

The mass factor e can be approximated with 1.06 (Ihme, 2016, p.39) for cargo trains. The rolling resistant coefficient f^{R} is very low for trains and can be approximated with 0.001 (Ihme, 2016, p.36). $g = 9.81\frac{\text{m}}{\text{s}^2}$ is the gravitational constant and $\alpha(t)$ is the angle of the slope of the track where the train is at the time t. The work done to move a train during the interval $[t_1, t_2]$ is calculated by:

$$W_{t_1,t_2} = \int_{t_1}^{t_2} \left(F^{\text{ACC}}(t) + F^{\text{SLOPE}}(t) + F^{\text{ROLL}}(t)\right) \cdot s(t)dt \tag{7}$$

The total work done over the billing period is multiplied with the working price for the electric cost. The average power $\bar{P}_{t_1,t_2}$ is the derivative of work over time:

$$\bar{P}_{t_1,t_2} = \frac{W_{t_1,t_2}}{t_2 - t_1} \tag{8}$$

As described before, the peak demand $P^{\max}$ also generates cost. In the German railroad electricity network, $P^{\max}$ is the maximum average power over all 15-minute intervals during the billing period. To test the calculation of the work

5

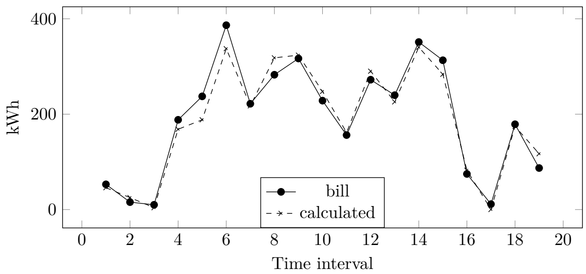


Figure 2: Comparing the calculated and the measured work done by a locomotive traveling from Hamburg to Paulineaue for 04:30 hours.

done, a field test has been conducted. A smartphone with a GPS tracker measured the position of a locomotive traveling from Hamburg to Paulinenaue in 04:30 hours. About 260km were covered by the train with a weight of 757t. After the removal of inaccurate GPS measurements, the distances, velocities, and accelerations where extracted to calculate the work done using the described forces. Figure 2 shows a comparison of the calculated and the billed work done for the tested locomotive. The field test justifies the usage of the described forces to calculate the work done.

3.2 Online approach

In this section, the general strategy is described which coordinates the fleet of electrical locomotives to reduce the peak demand. The mathematical optimization model is embedded in the strategy and is introduced in Section 3.4.

The billed peak demand is the maximum average power over all 15-minute intervals during a calendar year. Obviously, not all schedules are known one year in advance. Only the schedules in the next couple of hours or days are fixed. Consequently, the optimization horizon is set to the near future with known schedules. Therefore, only the work done of the near future is under control, even if the peak demand is based on the whole year. At the start of the optimization horizon some value for the peak demand already exist, because it has been reached sometime during the start of the year and the start of the optimization horizon. This impacts the optimization, because keeping the work done below that historic peak demand does not increase the cost for the peak demand. Only exceeding the historic peak demand should be prevented or at least kept as low as possible. Consequently, the historic peak demand is an input to the optimization. At the start of the year the historic peak demand is zero, but it should be initialized, because some peak demand is definitely going to exist in the upcoming year. A possible approach is to take the peak demand of the past year minus a desired reduction. Those circumstances are not the only reason, the optimization must be repeated regularly during the year.

Maintaining the safety of the trains has the highest priority when implement-

6

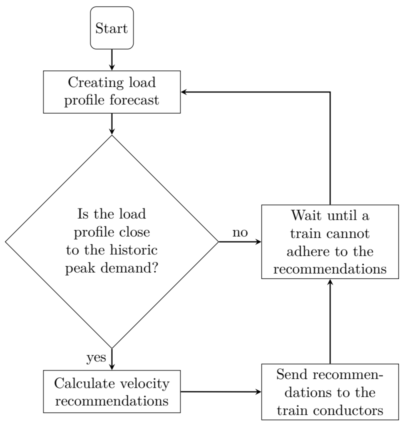


Figure 3: Flowchart of the online planning approach.

ing the approach of coordinating the trains. The train drivers are responsible for the safety and they always take the decision about their trains. Therefore, only driving recommendations can be made. Some events may occur and deviating from the driving recommendations is necessary to ensure the safety of the train and its environment. In such cases, the optimization should be adjusted to incorporate the most recent events. Additionally, the attention of the train drivers should be disturbed as little as possible with external driving recommendations to lower the peak demand.

An online approach of regularly recalculating the velocity recommendations fits all requirements. It regularly adapts with new or updated train schedules as well as unforeseeable events on the track. Calculating a forecast of the load profile as described in Figure 3 reduces the distraction of the train drivers, because only in periods of expected high load profiles, recommendations are made. The upcoming schedules are known so that an estimation of the upcoming work done and therefore peak demand can be made. If this estimation does not increases the historic peak demand no coordination is necessary and the drivers can drive undisturbed. Otherwise, if a high peak demand is expected recommendations are calculated and communicated to the train drivers.

The next two sections 3.3 and 3 explain the process of calculating the velocity recommendations.

3.3 Assumptions

Modeling the described problem requires certain assumptions. We assume the price model of a separate fee for the peak demand. For calculating the work done, the acceleration force, the rolling resistance, and the downhill force are used. The air resistance and other resistances are ignored.

Since the rolling resistance and the downhill force depend on the slope of the track, the slope is assumed to be partially linear. The track from the origin to the destination of each train is divided into multiple track partitions with a linear slope. On German railroads, the maximum slope is about 2.3° (Heirich et al., 2011). This assumption greatly simplifies the equations (6) and (5), because the angle α is constant for each track partition.

Combining the assumption of linear track partitions with the assumption of a constant acceleration over a short period of time leads to constant work done for each train, track partition and time period. The rolling resistance and the downhill force only depend on the slope of the track and are independent of the acceleration. Those forces can be calculated for each track partition before any optimization takes place. Calculating the track partition on which each train is at a given point in time and choosing the correct set of forces remains part of the optimization. A constant acceleration over a short period of time leads to a constant acceleration force over the same time period. With these assumptions all forces required to describe the movement of the train are discrete for each track partition, period of time and train. This paper formulates a non-linear model, a tailored heuristic and a genetic algorithm.

3.4 Non-linear model

3.4.1 Notation

Before diving into the equations of the non-linear optimization model, the sets, parameters and variables are introduced. Table 1 lists all sets used in the non-linear model. Even though the energy cost is calculated for each year the

Table 1: Sets in the non-linear model

$\mathcal{I}$	set of vehicles
$\mathcal{K}_i$	set of time periods in which the train $i \in \mathcal{I}$ operates
$\mathcal{J}_i$	set of all track partitions the vehicle $i \in \mathcal{I}$ traverses before arriving at the destination

optimization horizon is reduced to the upcoming and foreseeable schedules This optimization horizon is divided into time multiple time periods with the duration Δt The start and end of each schedule is represented by the corresponding set of time periods $\mathcal{K}_i$ for each vehicle $i \in \mathcal{I}$ The journey of each vehicle $i \in \mathcal{I}$ traverses multiple track partitions with a discrete slope

Table 2 lists all parameters used in the non-linear model Only the work done during the current optimization horizon is under control, but the peak demand of the whole year is billed. Therefore, p^{hist} is the peak demand which occurred in the current year but before the optimization horizon starts. Accordingly, only the fraction of the peak demand surpassing the historic peak demand is minimized during the optimization. If the peak demand during the optimization horizon is lower than the historic peak demand, no additional cost is generated. At the beginning of the year the historic peak demand p^{hist} is initialized with the value of the past year minus a desired reduction. f_{ij} combines the rolling resistance and the downhill force for the vehicle $i \in \mathcal{I}$ on the track partition

Table 2: Parameters in the non-linear model

Δt	the duration of a single time period
q	the number of time periods per billing interval
c_1	the cost coefficient for the peak demand
c_2	the cost coefficient for the total work done
p^{hist}	the historic peak demand already reached in this year
m_i	the mass of vehicle $i \in \mathcal{I}$
d_i	the total distance from the origin to the destination for the vehicle $i \in \mathcal{I}$
$a_i^{\max}$	the maximum acceleration of the vehicle $i \in \mathcal{I}$
e_i	the mass factor of vehicle $i \in \mathcal{I}$
f_{ij}	the combined rolling resistance and downhill force for vehicle $i \in \mathcal{I}$ on the track partition $j \in \mathcal{J}_i$
l_{ij}	the distance from the origin to the end of the track partition $j \in \mathcal{J}_i$ for the vehicle $i \in \mathcal{I}$
$s_i^{\max}$	the maximum possible distance vehicle $i \in \mathcal{I}$ can travel in Δt
v_{ij}^{track}	the maximum velocity for the vehicle $i \in \mathcal{I}$ on the track partition $j \in \mathcal{J}_i$

$j \in \mathcal{J}_i$. The vehicles are limited by their maximum velocity and the track partitions speed limit v_{ij}^{track}.

Table 3 lists all variables used in the non-linear model.

Table 3: Variables in the non-linear model

Z	the objective value describing the energy cost generated during the optimization horizon
$P^{\max}$	the peak demand during the optimization horizon
S_{ik}	the total distance the vehicle $i \in \mathcal{I}$ traveled at the end of the time period $k \in \mathcal{K}_i$
V_{ik}	the velocity of the vehicle $i \in \mathcal{I}$ at the end of the time period $k \in \mathcal{K}_i$
ΔS_{ik}	the difference of the travelled distance by the vehicle $i \in \mathcal{I}$ between the time periods k and $k-1$
ΔV_{ik}	the velocity change of the vehicle $i \in \mathcal{I}$ between the time periods k and $k-1$
$\Delta S_{ik}^{\text{track}}$	the difference of the travelled distance by the vehicle $i \in \mathcal{I}$ between the time periods k and $k-1$ on the track partition $j \in \mathcal{J}_i$
W_{ik}^{acc}	the work done to accelerate the vehicle $i \in \mathcal{I}$ during the time period $k \in \mathcal{K}_i$
W_{ik}^{track}	the work done to compensate the downhill and rolling resistance force for the vehicle $i \in \mathcal{I}$ during the time period $k \in \mathcal{K}_i$
W_{ik}^{+}	the positive part of the work done by the vehicle $i \in \mathcal{I}$ during the time period $k \in \mathcal{K}_i$
W_{ik}^{-}	the negative part of the work done by the vehicle $i \in \mathcal{I}$ during the time period $k \in \mathcal{K}_i$
X_{ijk}	is 1, if the vehicle $i \in \mathcal{I}$ is on the track partition $j \in \mathcal{J}_i$ at the beginning of the time period $k \in \mathcal{K}$ and 0 otherwise

Figure 4 visualizes many symbols of the model. It plots the velocity, distance and work done by the train i_1 with $m_{i_1} = 1$kg, $f_{i_1} = 1$N and $e_{i_1} = 1$. The set $\mathcal{K}_{i_1}$ contains the elements k_0 representing $t = 0$ to k_9 representing $t = 45$s, since $\Delta t = 5$s. The train arrives already at the time $t = 40$s and has 5 more seconds to spare.

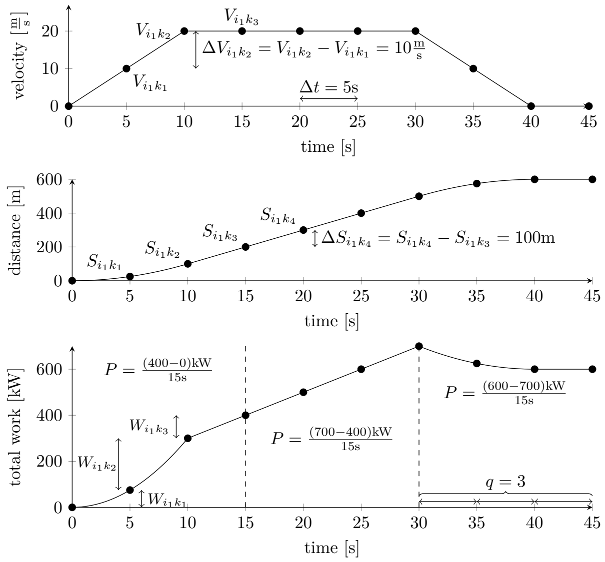


Figure 4: Visualization of symbols used in the model for the train i_1.

3.4.2 Model formulation

The non-linear model is defined as:

$$\text{minimize } Z = c_1 \cdot P^{\max} + c_2 \cdot \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} W^+_{ik} \tag{9}$$

10

subject to

$$\Delta V_{ik} = V_{ik} - V_{i,k-1} \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}_i \qquad (10)$$
$$\Delta S_{ik} = 0.5 \cdot \Delta t \cdot (V_{ik} + V_{i,k-1}) \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}_i \qquad (11)$$
$$S_{ik} = S_{i,k-1} + \Delta S_{ik} \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}_i \qquad (12)$$
$$S_{i|\mathcal{K}_i|} \geq d_i \qquad \forall i \in \mathcal{I} \qquad (13)$$
$$W_{ik}^{\text{acc}} = \frac{1}{\Delta t} \cdot e_i \cdot m_i \cdot \Delta V_{ik} \cdot \Delta S_{ik} \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}_i \qquad (14)$$
$$W_{ik}^{\text{track}} = \sum_{j \in \mathcal{J}_i} f_{ij} \cdot \Delta S_{ijk}^{\text{track}} \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}_i \qquad (15)$$
$$W_{ik}^{+} - W_{ik}^{-} = W_{ik}^{\text{track}} + W_{ik}^{\text{acc}} \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}_i \qquad (16)$$
$$\sum_{j \in \mathcal{J}_i} X_{ijk} = 1 \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}_i \qquad (17)$$
$$X_{ijk} \cdot l_{ij} \leq S_{ik} \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}_i, j \in \mathcal{J}_i \qquad (18)$$
$$(1 - X_{ijk}) \cdot d_i + X_{ijk} \cdot l_{ij} \geq S_{ik} \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}_i, j \in \mathcal{J}_i \qquad (19)$$
$$V_{ik} \leq \sum_{j \in \mathcal{J}_i} X_{ijk} \cdot v_{ij}^{\text{track}} \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}_i \qquad (20)$$
$$\Delta S_{ijk}^{\text{track}} \geq \Delta S_{ik} - s_i^{\max} \cdot (1 - X_{ijk}) \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}_i, j \in \mathcal{J}_i \qquad (21)$$
$$P^{\max} \geq \frac{1}{\Delta t} \sum_{i \in \mathcal{I}} \sum_{\substack{k' \in \mathcal{K} | \\ k' \geq k \wedge \\ k' < k+q}} W_{ik'}^{+} - p^{\text{hist}} \qquad \forall k \in \bigcup_{i \in \mathcal{I}} \mathcal{K}_i \text{ with } (k-1) \bmod q = 0 \qquad (22)$$
$$S_{ik}, W_{ik}^{+}, W_{ik}^{-}, \Delta S_{ik}, \Delta S_{ijk}^{\text{track}} \geq 0 \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}_i, j \in \mathcal{J}_i \qquad (23)$$
$$0 \leq V_{ik} \leq v_i^{\max} \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}_i \qquad (24)$$
$$0 \leq \Delta V_{ik} \leq a_i^{\max} \cdot \Delta t \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}_i \qquad (25)$$
$$P^{\max} \geq 0 \qquad (26)$$
$$X_{ijk} \in \{0; 1\} \qquad \forall i \in \mathcal{I}, k \in \mathcal{K}_i, j \in \mathcal{J}_i \qquad (27)$$

The objective function (9) minimizes the energy cost. With the assumption of a constant acceleration during each time period, the equation (11) describes the change in distance for each vehicle and time period. To ensure that each train reaches its destination, equation (13) bounds the distance traveled for the last time period of each train. Accelerating a train requires the work calculated in the equation (14). The equation (15) calculates the work done by each train in each time period incorporating the rolling resistance and downhill force. To ensure, that the correct track partition is used for this calculation the variable $\Delta S_{ik}^{\text{track}}$ is introduced to measure on which track partition the movement is done. The equations (17), (18) and (19) select the track partition $j \in \mathcal{J}_i$ on which the vehicle $i \in \mathcal{I}$ is at the beginning of the time period $k \in \mathcal{K}_i$. They compare the distance traveled for each vehicle and time period with the distance to the end of all track partitions. Restricting the velocity to the speed limit of each track partition is the purpose of equation (20). The equation (21) stores the combination of the traveled distance per track and time period, which is used in equation (15) to calculate the work done with the corresponding force.

11

The total work done is split into a positive and negative part in equation (16). In the equation (22) the total power combining all trains is calculated for all q consecutive time periods. This reflects how the energy provider calculates the price for the peak demand. If the power does not exceed the historic peak demand, no extra costs are generated. Otherwise the peak demand will be set accordingly and the cost for exceeding the historic peak demand are reflected in the objective function.

The above defined model implements a price model with an extra cost for the peak demand. Other price models can be reflected as well without changing the core logic of the model. Most of the restrictions in the model calculate the traveled distance and the work done. Only small adjustments are necessary to reflect time-dependent energy prices or other price structures.

On top of the non-linearity, the model is not convex. It is not possible, to find the optimal solution in a reasonable amount of time, if the optimal solution can be found by the standard solvers at all. A tailored heuristic and a metaheuristic are introduced in the next section to find acceptable solutions within the time limitations of the required online approach.

4 Heuristic approach

4.1 Core fundamentals

All following heuristics share the core approach of determining feasible velocity-time graphs for each train. The distance covered by each train must match the distance from the schedule, while adhering to velocity limits of the train and speed limits on the various track partitions. Additionally, the acceleration cannot exceed the technical limitations of the train. Adjusting the velocity of a feasible velocity-time graph is not possible without adjusting the velocity at another point. Otherwise, the total distance traveled by that train would not stay the same.

We developed a framework which generates feasible solutions based on a simple input. The core idea is to introduce a set of artificial, time dependent velocity limits for each train. A velocity limit caps the velocity of a train for a certain period of time. Those velocity limits are time dependent, contrary to the distance dependent velocity limits given by the track partitions. Now, let the train move to its destination as fast as possible, while adhering to all velocity limits. Figure 5 illustrates those artificial velocity limits and the resulting velocity-time graph. Adjusting the artificial velocity limits may result in a different velocity-time graph with a corresponding work done.

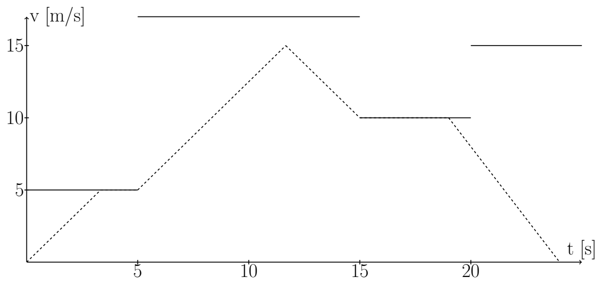


Figure 5: Artificial velocity limits and the resulting velocity-time graph.

The slope of the velocity-time graph is determined by the maximum and minimum acceleration of the train. A train capable of a higher acceleration has a steeper increase in the velocity-time graph. The train in this example accelerates until the velocity limit of $5\frac{\text{m}}{\text{s}}$ is reached after a couple of seconds. This velocity does not change until the artificial velocity limit increases. The train brakes at 11.7s with a velocity of $15\frac{\text{m}}{\text{s}}$ to reach the velocity limit of $10\frac{\text{m}}{\text{s}}$ at 15s. In this case the artificial velocity limit of $17\frac{\text{m}}{\text{s}}$ starting at 5s cannot be reached without exceeding a velocity limit in the future. Reaching the destination with a velocity of 0 is only possible, if the train decelerates at the 19s mark. Note, that in this example the velocity limit of the track is high enough and does not have any impact.

Calculating the resulting velocity-time graph for a train given a schedule and artificial velocity limits is straight forward. The train accelerates with the maximum acceleration if possible. If a time-based artificial velocity limit or a distance-based track partition velocity limit is reached the acceleration is set to 0. The train brakes if any future artificial velocity limit or track partition velocity limit requires a reduction in velocity.

With this framework many solutions can be generated easily by creating a set of artificial velocity limits for each train. The generated solutions are feasible unless the artificial restrictions are too tight so that the destination is not reached in time.

4.2 Tailored heuristic

The tailored heuristic uses the framework of artificial velocity restrictions to reduce the peak demand. In a nutshell, this approach iteratively restricts the velocity of all trains during the time period of the highest energy consumption. During the initialization, artificial velocity limits are generated. Since the calculation of the peak demand is based on the average work done every 15 minutes, the artificial velocity limits are initialized for the same time periods with a duration of 15 minutes. Initially, the velocity limits are as high as possible, thus matching the maximum velocity of each train. Table 4 illustrates the initialization for two trains in the first row. The first train cannot drive faster than $20\frac{\text{m}}{\text{s}}$

13

and the second train $30\frac{\mathrm{m}}{\mathrm{s}}$ respectively.

Table 4: Example how the artificial velocity limits change during the tailored heuristic

	Velocity limit in each time period $[\frac{\mathrm{m}}{\mathrm{s}}]$					
	10:00 - 10:15		10:15 - 10:30		10:30 - 10:45	
Iteration	Train 1	Train 2	Train 1	Train 2	Train 1	Train 2
Initialization	20	30	20	30	20	30
1	20	30	19	30	20	30
2	20	30	18	30	20	30
3	20	29	20	30	20	30
4	19	29	20	30	20	30

Now, the iterative process begins and the movements are calculated as described in the previous section. The trains move as fast as possible to their destinations while adhering to the artificial velocity limits as well as the velocity limits given by the track partitions. With the resulting velocity-time graphs it is possible to calculate the work done by each train and determine the time period with the highest energy consumption. The time periods with the highest energy consumption have a gray background in Table 4 for each iteration. Stop the algorithm, if the current peak demand is not bigger than the historic peak demand, because no additional cost is generated. Otherwise, reduce the artificial velocity limit during the peak period of the train with the most freedom concerning its schedule. This freedom is calculated by subtracting the time of arrival from the latest possible arrival time for each train. A train arriving 10 minutes before schedule therefore has more freedom than a train arriving only 5 minutes before schedule. The velocity-time graph changes for the train with the tightened velocity limits. A reduced velocity reduces the work done during that period of time. If the tightened velocity limits lead to an infeasible train movement (because the train now arrives after the scheduled arrival) or if the velocity limit of the current peak period is already 0, try adjusting the artificial velocity limits of the next train. Stop the algorithm, if the artificial velocity limits of no train can be reduced in the current peak period.

The initialization in the exemplifying Table 4 lead to a peak demand in the time period from 10:15 to 10:30. Note, that the table only lists the velocity limits of the two trains for each iteration of the tailored approach. The work done, travel times and schedules are not listed and are only mentioned in the following explanation of the example. Let train 1 be the train with more freedom than the second train. The velocity limits are reduced from $20\frac{\mathrm{m}}{\mathrm{s}}$ to $19\frac{\mathrm{m}}{\mathrm{s}}$ for the first time in the peak period from 10:15 to 10:30. Now, the velocity-graph for train 1 is calculated again with the updated velocity limits. Still, the same time period has the highest energy consumption and the velocity limits in that time period are reduced again. This reduction in the velocity of the first train finally lowered the energy consumption in the time period from 10:15 to 10:30 below the energy consumption of the first time period from 10:00 to 10:15. The reduction of the velocity limit from 20 to 18 increases the travel time of the first train so that the second train now has more time between the actual and scheduled arrival than the first train. Thus, the velocity limits of the second

14

train are adjusted and lowered from 30 to 29, but this change does not change the peak period 10:00 to 10:15. The velocity limits of the first time period are reduced again, but this time for the first train, because the most recent change switch the order. Even though the peak period changed now to the last one from 10:30 to 10:45, no further improvements can be made, because any reduction of the velocity limit of either train leads to a violation of the scheduled arrival time.

Another example to clarify the mechanics of the tailored heuristic is illustrated in Figure 6. For two trains the velocity-time graph and the corresponding work done combined by both trains are illustrated twice. Firstly, the graphs show the results of the initialization and secondly the graphs show the results of the final result after completing the optimization. Both trains are fast and arrive early during after initializing the movements which lead to a high consumption in the three time periods from 15mins to 60mins, because both trains have a high energy consumption during these periods. The trains are scheduled to start at 10:10 and 10:13. The time periods used by the energy provider to bill the peak demand start every full 15 minutes. That is why $t = 0$ represents 10:00 and the trains start at $t = 10\text{min}$ and $t = 13\text{min}$. During the optimization the heuristic lowers the artificial velocity limits and stretch the operation time of both trains. Especially the second train is restricted during the initial peak periods, because its arrival can be greatly delayed compared to the first train. This change lead distributing the work done more evenly throughout the optimization horizon which lowers the peak demand by 55%.

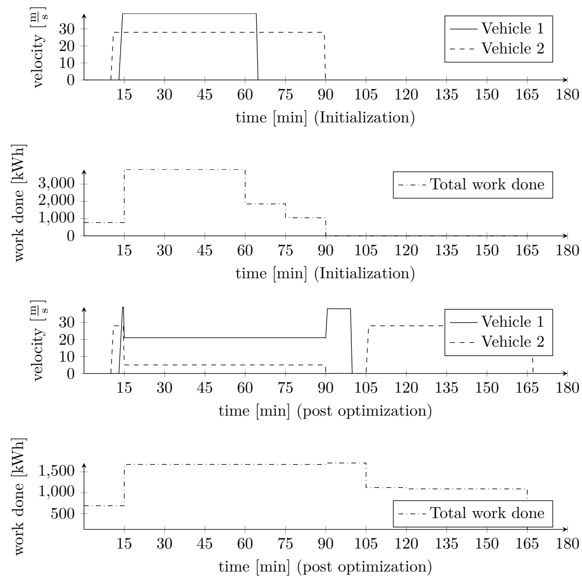


Figure 6: Velocity-time graphs for a fleet of two trains and their combined energy consumption before and after applying the tailored heuristic.

4.3 Genetic algorithm

The genetic algorithm assesses a group of individuals, called a population, by their fitness and generates a new population with individuals based on the individuals of the old population. Genetic algorithms are popular for delivering near-optimal solutions within a reasonable time (Stadtler et al., 2015). Modelling the individuals is a defining step to solve a problem with the genetic algorithm. We are using the artificial velocity limits to model an individual. Each individual contains the artificial velocity limits for each train of the schedule. Initially, the artificial velocity limits are generated randomly. To evaluate an individual the velocity-time graphs for each train are calculated so that the work done and therefore the energy cost can be determined. This fitness value is relevant for the selection process. The roulette wheel selection approach is used to select individuals for recombination, where individuals with a better (lower) fitness value have a higher probability of being selected. Table 5 shows an example of the applied bit crossover approach, where two individuals create a new individual. For each train, the artificial velocity limits of the two parent individuals are selected alternatingly. The child inherits the velocity limits of the first train for the time period 10:00 to 10:15 from the mother, from 10:15 to 10:30 from the father and lastly from 10:30 to 10:45 from the mother. No velocity limits exist during the time period 10:30 to 10:45 for the second train because its schedule ends before 10:30. A few randomly selected individuals are mutated by increasing or decreasing the velocity limits by a small amount as shown in Table 5.

Table 5: Sample of the applied crossover approach of the genetic algorithm

		Velocity limit in each time period $[\frac{m}{s}]$		
Train	Individual	10:00 - 10:15	10:15 - 10:30	10:30 - 10:45
Train 1	mother	20	30	10
	father	25	35	10
	child	20	35	10
	mutation	19	34	11
Train 2	mother	15	20	-
	father	20	30	-
	child	15	30	-
	mutation	16	31	-

5 Computational study

5.1 Data generation

During the computational study, the length, slope, and maximum velocity for the track partitions are based on actual data from the German railroad system. The DB Netze is a subsidiary of Deutsche Bahn and publishes track-related data on their website called *Infrastrukturregister*. A set of 324 track partitions is used to generate random schedules for the trains. Some properties of those tracks are listed in Table 6. The value of the historic peak demand is set to $p^{\text{hist}} = 0$ to produce comparable results. A computer with an Intel Xeon CPU E5-2667 v3 processor, 256gb RAM and Baron as a solver has been used for all studies in this paper. Each instance has a computation limit of 5 minutes because the computation time is limited in the previously described online approach. The genetic algorithm operates with 300 individuals and 10 mutations per generation.

Table 6: Properties of the set of 324 track partitions.

	Distance [km]	Max velocity $[\frac{km}{h}]$	Slope [‰]
Minimum	0.1	50	2.5
Maximum	20.9	280	17.5
Average	3.5	141	6.6

17

5.2 Parametrization of the non-linear approach

Δt is the only parameter of the non-linear model which can be adjusted. All other parameters are given by the track, train or schedule. The duration of Δt determines the frequency of how often the acceleration of each train can change. Despite managerial guidelines on how often the train driver should adjust the acceleration of the train, the value of Δt impacts the computation time and the objective value. Shorter Δt enable finer adjustments of the acceleration but lead to an increasing computational effort. To measure the impact of Δt on the objective value a computational study has been conducted. Every instance is computed six times with the non-linear model, each time with a different value for Δt while keeping the computation time limit at 5 minutes. Only unrealistically tiny instances are computed in this study. To obtain feasible solutions at all with the non-linear model and Δt set to a few seconds, the trains are scheduled to drive very short distances of 5km to 10km. Additionally, the billing interval duration to compute the peak demand is reduced from 15 minutes to 2 minutes. Otherwise all train movements would take place in the same billing interval making the peak demand minimization otiose. Furthermore, for this study, the schedule of each train contains a single-track partition only: $|\mathcal{J}_i| = 1 \quad \forall i \in \mathcal{I}$. This reduces the computational effort even more, because no integer variables remain in the model. The binary X_{ijk} is the only integer variable and indicates the current track partition for each train. Since there is only one track partition for each train, all X_{ijk} variables are 1. Even with these simplifications the optimality-gap is huge if a best bound bigger than 0 can even be found. Furthermore, an instance with three trains (470 variables and 346 equations) delivered, after a computation time of 12 hours, a solution still 3.7 times bigger than the best bound.

Figure 7 illustrates the average gap to the best found solution for instances with 1 to 10 trains.

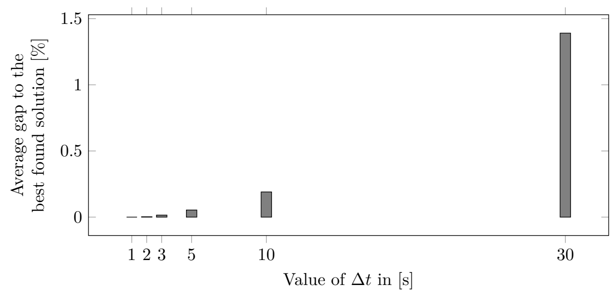


Figure 7: Average gap of the non-linear solution approach for six values for Δt

The gap directly translate to the increase in costs, since $p^{\text{hist}} = 0$. As expected, smaller values for Δt deliver the best solutions on average. Increasing

18

Δt increases the objective value. This trend breaks for larger instances with the increasing computational difficulty. Increasing the number of trains from 10 to 15 skyrockets the objective value of the non-linear model with $\Delta t = 1$s if a feasible solution can even be found in 5 minutes. Despite saving the train driver from adjusting the acceleration every few seconds, small values for Δt can lead to bad solutions without sufficient computational time.

5.3 Solution approach comparison

To compare all solution approaches they solve the same instances and their resulting solutions are compared. The tailored heuristic, the genetic algorithm, the nonlinear formulation and a fast arrival solution are compared. For the fast arrival solution all trains drive as fast as possible to arrive at their destinations as early as possible. The fast arrival solution approach does not deliver competitive solutions but helps valuate other solutions. Despite the default genetic algorithm approach (GA) a second variant is tested which has the artificial velocity limits produced by the tailored heuristic seeded to the initial population (GA inject).

Two set of instances are used for this comparison. Firstly, 35 very small instances are generated as described in the previous Section 5.2. The second set of 84 instances are generated using realistically sized schedules.

Figure 8 illustrates the average gap to the best found solution for each solver on the small instances. Two values for Δt are tested for the non-linear model (non-linear 1s and non-linear 2s).

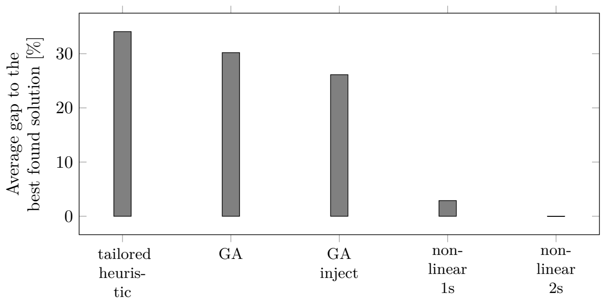


Figure 8: Comparison of all solution approaches on small instances

The fast arrival solutions have an average gap to the best found solution of 188%. Solution obtained with the non-linear model and Baron as a solver always find the best solution among all solution approaches. Setting $\Delta t = 1$ delivers better solution for instances with less than 7 vehicles than setting $\Delta t = 2$ but instances with more than 7 getting too hard to solve and sometimes relatively bad solution are found. Injecting the solution of the tailored heuristic to the genetic algorithm improves the solution.

19

After analyzing the performance of the solution approaches on small instances, realistically sized instances are scrutinized. For this study, 84 instances are generated with 1 to 15 trains and a distance with a length of 80km to 160km per train on 1 to 15 track partitions. Figure 9 illustrates the fraction of instances that could be solved with the non-linear model and Baron as a solver in 5 minutes.

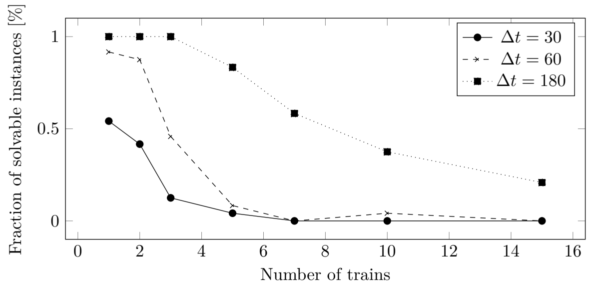


Figure 9: Fraction of solvable instances with a realistic size by the non-linear model in 5 minutes

As expected, more instances can be solved if Δt increases and if the number of trains decreases. Setting $\Delta t = 180$ finds the most solutions, but it also impacts the quality of the computed solutions. Figure 10 illustrates the average gap to the best found solution for the realistic sized instances where all solution approaches found a feasible solution.

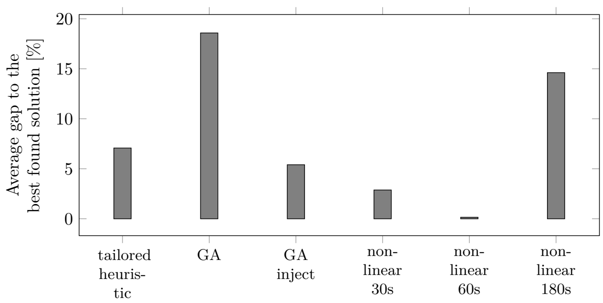


Figure 10: Comparison of all solution approach on realistic sized instances for which all solution approaches found a solution.

20

Even though the non-linear model with $\Delta t = 180$ finds more often a solution than the other two parameterizations for Δt, the objective value is worse on average. The tailored heuristic and the genetic algorithms perform better on the realistic sized than on the small sized instances in comparison to the solutions of the non-linear model. It is noticeable, that in contrast to the small instances, the tailored heuristic performs better than the default genetic algorithm. In conclusion, solutions obtained with the non-linear model and small values for Δt deliver the best results but often no feasible solution is found at all. Figure 11 compares the tailored heuristic and the genetic algorithm on all realistic sized instances including the ones not solvable by the non-linear model in 5 minutes.

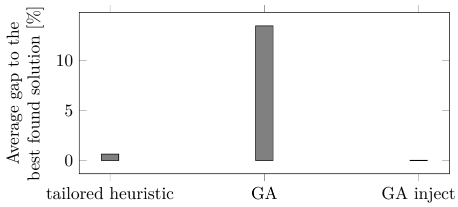


Figure 11: Comparison of the tailored heuristic and the genetic algorithm on realistic sized instances

The genetic algorithm again can improve the solution obtained by the tailored heuristic. The tailored heuristic performs better than the default genetic algorithm.

Both the genetic algorithm and the non-linear model exhausted the 5-minute computational time limit for all instances, whereby the tailored and the fast arrival heuristic never took more than a second. Especially on small instances the average gap and therefore the average increase in cost of the heuristic approaches is substantial. On the contrary, the resulting velocity recommendations created by the heuristic approaches are more structured. Figure 12 illustrates, that the resulting velocity recommendations of the tailored heuristic are more expensive, but better communicable to the train driver.

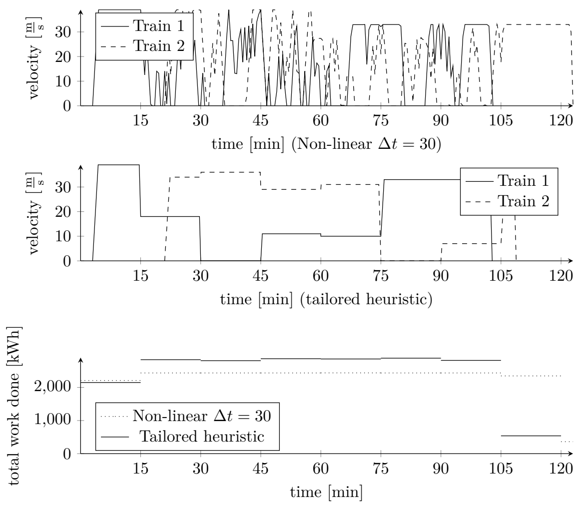


Figure 12: The resulting velocity-time graph of the non-linear $\Delta t = 30$ model and the tailored heuristic, and the total work done in each interval

For this example, we selected an instance on which the tailored heuristic performs relatively weak. The non-linear approach with $\Delta t = 30$ found the best solution and the solution of the tailored heuristic has a gap of 18%. The fine tunable velocity adjustments lower the peak demand and therefore the energy cost but are difficult to execute for the train driver.

5.4 Further computational studies

After comparing the solution approaches, the benefit of coordinating a fleet of trains is scrutinized. Firstly, each train of every schedule is optimized separately with the tailored heuristic. The tailored heuristic runs once for every train in the schedule and calculates the movement of a single train in each run. All other trains and their movements are ignored. After completing the optimization for every train, the energy cost is calculated based on the combined movement of all trains. Secondly, this result is compared to the result when optimizing the whole fleet of trains at once. Figure 13 illustrates the improvement of coordinating the fleet of trains instead of optimizing each train separately. 6400 instances with 2 to 20 trains have been calculated.

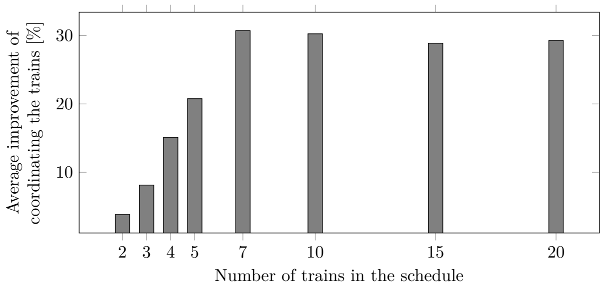


Figure 13: The benefit of coordinating the fleet of trains instead of optimizing each train separately.

As expected, coordinating the trains delivers the better results on average. The savings with two trains are relatively small because the peak demand can maximally be cut in half whereas with three trains the maximum reduction is to a third. With more than 7 trains the savings stabilize around 30% and the standard deviation of the gap reduces from 11.9 with 7 trains to 10.3 with 20 trains. Even small improvements of 3.8% with two trains can be very lucrative for railroad companies with energy cost of 1 million euros, if the operation stick to the optimization during the whole billing period which is typically a year.

Another study has been conducted to measure the impact of the operation time on the energy cost. The operation time is the amount of time a train has to reach its destination. 6400 Instance have been generated and each time the operation time is adjusted from 50% to 150% of the regular duration. Figure 14 illustrates the relative change in the energy cost for different operation time adjustments. The energy cost depends on the historic peak demand as explained in the restriction 22.

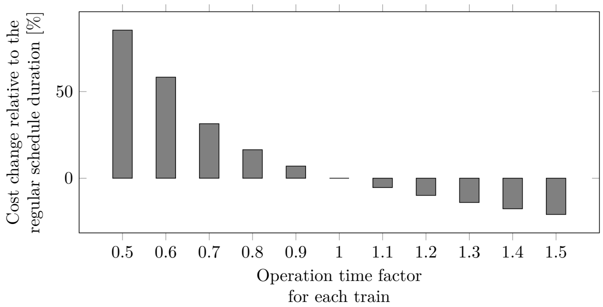


Figure 14: The influence of a extension and shortening of the operation time of each train.

As expected, a reduction in the operation time lead to an increase in the energy cost, because the flexibility to shift the work done reduces. An increase in the operation time increases the flexibility of each train which can reduce the peak demand.

6 Conclusion

We formulated a non-linear mathematical optimization model which coordinates electrical cargo trains to minimize the energy cost. Our field study evaluated the accuracy of the physical formulas which are used in the model. The nonlinearity of calculating the work done and the power consumption makes the model hard to solve. Introducing time-dependent artificial velocity limits greatly simplifies the optimization and can be exploited in various heuristics. A tailored heuristic finds reasonable solutions in less than a second which is necessary for an implementation in an online approach. The developed genetic algorithm improves the solution obtained by the tailored heuristic or deliver reasonable results itself. The non-linear model finds the best solutions for small instances but gets overwhelmed on big instances. All heuristics create better structured velocity recommendations than the results of the non-linear model, which makes them better communicable to and executable by the train driver. For a practical implementation we advise calculating a solution with the tailored heuristic and try to find an improved solution with the non-linear model with as much time as possible provided by the circumstances of the online approach.

Even though this paper optimizes a fleet of electrical cargo trains the core concept can be adjusted for other applications as well. Mainly the calculation of the work done needs to be adjusted, because in many applications the reduction of the peak demand can greatly decrease the energy cost. We showed that reducing the peak demand is very lucrative, especially for railroad companies operating more than 5 trains simultaneously.

In further research the tailored heuristic can be compared to other metaheuristics to better evaluate the performance. Additionally, reformulating the model or calculating helpful starting solutions for the non-linear approach is worth looking into. Since the output of the optimization delivers velocity recommendations, the execution by the train drivers needs to be analyzed. The impact of unforeseeable events may have a big impact on the performance. A robust extension of the optimization could be worth scrutinizing to cope with external influences. The algorithm is planned to be implemented by a railway company in 2021.

References

Albrecht, A., Howlett, P., Pudney, P., Vu, X., & Zhou, P. (2018). The two-train separation problem on non-level track—driving strategies that minimize total required tractive energy subject to prescribed section clearance times. *Transportation Research Part B: Methodological*, *111*, 135–167. https://doi.org/10.1016/j.trb.2018.03.012

Albrecht, A., Howlett, P., Pudney, P., Vu, X., & Zhou, P. (2015). Energy-efficient train control: The two-train separation problem on level track. *Journal of Rail Transport Planning & Management*, *5*(3), 163–182. https://doi.org/10.1016/j.jrtpm.2015.10.002

Albrecht, T. (2010). Reducing power peaks and energy consumption in rail transit systems by simultaneous train running time control. *WIT Transactions on State-of-the-art in Science and Engineering*, *39*.

Bärmann, A., Martin, A., & Schneider, O. (2017). A comparison of performance metrics for balancing the power consumption of trains in a railway network by slight timetable adaptation. *Public Transport*, *9*(1-2), 95–113. https://doi.org/10.1007/s12469-017-0160-4

Ciccarelli, F., Iannuzzi, D., & Tricoli, P. (2012). Control of metro-trains equipped with onboard supercapacitors for energy saving and reduction of power peak demand. *Transportation Research Part C: Emerging Technologies*, *24*, 36–49. https://doi.org/10.1016/j.trc.2012.02.001

DB Netze. (2019). Preisblatt für die nutzung des 16.7-hz-bahnstromnetzes (bahnstromnetz) gültig ab dem 01.01.2019 [Accessed: 28.05.2020].

Fernandez, M., Li, L., & Sun, Z. (2013). 'just-for-peak' buffer inventory for peak electricity demand reduction of manufacturing systems. *International Journal of Production Economics*, *146*(1), 178–184. https://doi.org/10.1016/j.ijpe.2013.06.020

Gu, Q., Tang, T., Cao, F., Karimi, H. R., & Song, Y. (2013). Peak power demand and energy consumption reduction strategies for trains under moving block signalling system. *Mathematical Problems in Engineering*, *2013*, 1–11. https://doi.org/10.1155/2013/940936

Heirich, O., Lehner, A., Robertson, P., & Strang, T. (2011). Measurement and analysis of train motion and railway track characteristics with inertial sensors. *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. https://doi.org/10.1109/itsc.2011.6082908

Ihme, J. (2016). *Schienenfahrzeugtechnik*. Springer Fachmedien Wiesbaden. https://doi.org/10.1007/978-3-658-13541-6

Liu, R. (, & Golovitcher, I. M. (2003). Energy-efficient operation of rail vehicles. *Transportation Research Part A: Policy and Practice*, *37*(10), 917–932. https://doi.org/10.1016/j.tra.2003.07.001

Miyatake, M., & Ko, H. (2010). Optimization of train speed profile for minimum energy consumption. *IEEJ Transactions on Electrical and Electronic Engineering*, *5*(3), 263–269. https://doi.org/10.1002/tee.20528

Nolde, K., & Morari, M. (2010). Electrical load tracking scheduling of a steel plant. *Computers & Chemical Engineering*, *34*(11), 1899–1903. https://doi.org/10.1016/j.compchemeng.2010.01.011

Oldewurtel, F., Ulbig, A., Parisio, A., Andersson, G., & Morari, M. (2010). Reducing peak electricity demand in building climate control using real-time pricing and model predictive control. *49th IEEE Conference on Decision and Control (CDC)*. https://doi.org/10.1109/cdc.2010.5717458

Scheepmaker, G. M., Goverde, R. M., & Kroon, L. G. (2017). Review of energy-efficient train control and timetabling. *European Journal of Operational Research*, *257*(2), 355–376. https://doi.org/10.1016/j.ejor.2016.09.044

Shao, S., Pipattanasomporn, M., & Rahman, S. (2012). Grid integration of electric vehicles and demand response with customer choice. *IEEE Transactions on Smart Grid*, *3*(1), 543–550. https://doi.org/10.1109/tsg.2011.2164949

Stadtler, H., Kilger, C., & Meyr, H. (Eds.). (2015). *Supply chain management and advanced planning*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-55309-7

Tang, H., Wang, Y., Liu, X., & Feng, X. (2020). Reinforcement learning approach for optimal control of multiple electric locomotives in a heavy-haul freight train:a double-switch-q-network architecture. *Knowledge-Based Systems*, *190*, 105173. https://doi.org/10.1016/j.knosys.2019.105173

Wang, Y., Schutter, B. D., van den Boom, T. J., & Ning, B. (2014). Optimal trajectory planning for trains under fixed and moving signaling systems using mixed integer linear programming. *Control Engineering Practice*, *22*, 44–56. https://doi.org/10.1016/j.conengprac.2013.09.011

Wichmann, M. G., Johannes, C., & Spengler, T. S. (2018). An extension of the general lot-sizing and scheduling problem (glsp) with time-dependent energy prices. *Journal of Business Economics*. https://doi.org/10.1007/s11573-018-0921-9

2.2 The quadratic multiple knapsack problem assigning storage locations in a warehouse

The quadratic multiple knapsack problem assigning storage locations in a warehouse

Justus Bonz, Knut Haase

October 5, 2020

Abstract

The assignment of Stock Keeping Units (SKUs) to storage locations is an important problem when trying to minimize the operating cost of order picking. We consider a warehouse of a beverage retailer with a given assignment of SKUs to aisles in the warehouse. A limited number of SKUs can be rearranged to new aisles with the objective of minimizing the number of visited aisles when picking a set of orders. Furthermore, the number of replenishments of SKUs is minimized too. We formulate a quadratic multiple knapsack problem to solve big instances and apply the 2-opt algorithm for improvement. Additionally, we propose a method to generate a smaller set of representative orders with similar properties like the original set of orders. This reduced set of representative orders is used by the optimal model formulation to generate a solution which can be evaluated with the original set of orders. Using a data set from the beverage retailer with 356451 orders over two years shows that rearranging the warehouse once a year can reduce the number of visited aisles by 3%. Rearranging only a subset (5%) of all SKUs in a warehouse yields already a majority (61%) of the possible improvements.

Keywords: warehouse; storage location assignment problem; quadratic multiple knapsack problem; management insights; order picking;data reduction

1 Introduction

The assignment of SKUs to storage locations in the warehouse influences the duration of order picking. All SKUs of an order are retrieved from the assigned storage locations. In many warehouses every order is picked separately, hence order batching is not possible. Grouping SKUs which are

1

ordered frequently together reduces the number of accessed groups per order. Each group can represent a certain area in the warehouse, a rack or any separated unit of storage. Identifying the storage location for all SKUs while minimizing the number of accessed groups when picking orders is the core problem handled in this paper. Figure 1 illustrates the described problem with 4 SKUs and 2 storage groups. In this example, each SKU takes up the same amount of space. There are four SKUs assigned to two storage groups. Switching the assignments of the SKUs B and C improves the solution. When picking the 3 orders the number of accessed storage groups is reduced from 6 to 4. Additionally, no replenishments are necessary any more.

Order Id	Ordered SKUs
1	A, A, B, C
2	A, C, C
3	B, D, D

	Storage group 1	Storage group 2
Assignable SKUs	A, B, C	A, B, C, D
Maximum number of SKUs	2	2
Number of items per SKU	3	2
Initial assignment	A, B	C, D
Rearrangement	A, C	B, D

		Storage group visits	Replenishments
Initial assignment	Storage group 1	3	0
	Storage group 2	3	1
	Total	6	1
Rearrangement	Storage group 1	2	0
	Storage group 2	2	0
	Total	4	0

Figure 1: Visualization of the problem assigning 4 equal sized SKUs to 2 storage groups.

Warehouses can already have a given storage location assignment. However, the composition of orders can follow seasonal or global trends and the assortment can change. Thus, a former storage location assignment may not be optimal anymore. Rearranging SKUs to new storage locations involves a certain amount of effort. This effort includes moving the SKUs, adjusting the labels in the warehouse and changing the data in the computer systems. Additionally, the pickers need to adapt to the changes.

The order, in which the storage groups are visited during the picking process, is fixed. This fixation ensures that fragile SKUs are placed on more robust SKUs. The assignment of storage locations should keep the SKUs in a feasible order of the picking process.

After depleting the stock of a SKU in a storage group, the SKU is replenished from the bulk storage. The frequency of the replenishment depends on the required space of the SKU, the size of a location in the storage group and the demand of the SKU. Therefore, larger and more often requested SKUs should be stored in storage groups with larger storage locations. In this paper, we identify reasonable frequencies for rearranging products in the warehouse to new storage locations. Furthermore, we study the impact of the number of rearranged SKUs on the order picking process. We base our research on a real data set of a beverage retailer containing 356451 orders over two years. More than 1000 products are stored in the warehouse.

The rest of this paper is organized as follows. In Section 2 we list current literature about the describe problem. In Section 3 we present the model solving the describe problem. We formulate a quadratic multiple knapsack problem to solve the model in Section 4.2. We list the results of the computational study in Section 5 using the data set of a beverage retailer. Finally, the Section 6 concludes the paper.

2 Literature

We classify our problem based on the literature review of de Koster et al., 2007. The underlying order picking process is a picker to parts system, where the order picker walks or drives inside the warehouse to pick up the required items. Furthermore, we assume the discrete packing variant, where order batching is not possible. The layout of the warehouse is fixed. The described problem is classified as a storage assignment problem with a separated bulk and pick stock. In such a scenario, the SKUs are retrieved from the pick stock, which gets replenished from the bulk stock.

SKUs can be assigned either to a specific shelf in a rack or to a more general storage group. The storage group can be a rack, a certain area in the warehouse, a vertical carousel (Litvak and Vlasiou, 2010) or any other storage allocation.

Many storage assignment approaches group the SKUs according to some measurement of order frequency. Other approaches show the benefit of considering the correlation of products in the assignment (Y. Zhang, 2016). Chiang et al., 2012 developed a heuristic which operates with a measurement for the association between two products. Their assignment of SKUs to aisles reduces the travel distance for order picking by up to 14%. R.-Q. Zhang et al., 2019 looked at demand patters of item groups of two and more items and presented two heuristics to assign products to shelves. A simulated annealing algorithm and a minimum increment heuristic outperform each other depending on the degree of the correlation between the products. Optimizing over a set of past orders is the content of Kress et al., 2016. A maximum of 200 products and 200 orders are analyzed, because larger instances haven't been solvable anymore within a reasonable time. Chiang et al., 2011 formulated a binary integer program to assign new products to empty shelves in a distribution center. They use a fitness value measuring the association between the new products and the empty shelves. Ang et al., 2012 introduce a robust mixed integer program which optimizes the storage location problem that minimizes the worst-case travel time in the warehouse.

Although many different settings have been studied in the literature, some aspects need further research. The optimization of existing storage location assignments can be investigated, if the assortment or the correlation pattern changes over time. Recommended frequencies of the storage location rearrangements are studied in this paper. Rearranging storage locations in a warehouse takes effort. We study the improvement of the order picking process while limiting the number of rearrangements in the warehouse. To the best of our knowledge, the multiple quadratic knapsack problem (Pisinger, 2007, Hiley and Julstrom, 2006) and data reduction approaches have not been applied to this problem yet.

3 Model

Let $\mathcal{O}$ be the set of orders. $\mathcal{J}$ is the set of storage location groups, for example aisles. Furthermore, let $\mathcal{I}$ be the set of SKUs and $\mathcal{I}_j$ with $\mathcal{I} = \bigcup_{j \in \mathcal{J}} \mathcal{I}_j$ being the set of SKUs that can be feasibly stored in the aisle $j \in \mathcal{J}$. The initial assignment $\mathcal{K}$ is a set containing the tuples (i, j) with $j \in \mathcal{J}$ and $i \in \mathcal{I}_j$. This set describes the current state of the warehouse, because a subset of the SKUs may already be stored in certain aisles.

The parameter w_o with $o \in \mathcal{O}$ corresponds to the frequency with which o is picked. Each SKU has a constant volume, the space of each storage location in an aisle is fixed and total number of ordered SKUs is known from the set of orders $\mathcal{O}$. Therefore, b_{ij} is the number of replenishments for the SKU if it is stored in the aisle $j \in \mathcal{J}$.

b_{ij} can be adjusted to normalize the effort of visiting an aisle during the order picking and the effort for replenishing a SKU. c_j is the maximum number of different SKUs that can be stored in the aisle $j \in \mathcal{J}$. d_{io} is 1, if the SKUs $i \in \mathcal{I}$ is demanded within the order $o \in \mathcal{O}$ and 0, otherwise. Not more than α SKUs within the initial assignment $\mathcal{K}$ may be rearranged.

Let F be the objective variable which counts the number of visited aisles when picking the orders $\mathcal{O}$ for a given assignment of SKUs to aisles. F also counts the number of replenishments needed if an aisle runs out of any SKUs. X_{ij} is 1, if the SKU $i \in \mathcal{I}$ is stored in the aisle $j \in \mathcal{J}$ and 0, otherwise. Based on this decision, Z_{oj} is 1, if the aisle $j \in \mathcal{J}$ needs to be visited when picking the order $o \in \mathcal{O}$ and 0, otherwise. Y_i tracks, if the SKU $i \in \mathcal{K}$ was rearranged and is 1 in that case and 0, otherwise.

The following mixed integer program extends the model from Kress et al., 2016 and models the underlying problem:

$$\text{minimize} \quad F = \sum_{o \in \mathcal{O}} \sum_{j \in \mathcal{J}} w_o \cdot Z_{oj} + \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}_j} b_{ij} \cdot X_{ij} \tag{1}$$

subject to

$$\sum_{j \in \mathcal{J} | i \in \mathcal{I}_j} X_{ij} = 1 \qquad \forall i \in \mathcal{I} \tag{2}$$

$$\sum_{i \in \mathcal{I}_j} X_{ij} \leq c_j \qquad \forall j \in \mathcal{J} \tag{3}$$

$$X_{ij} \leq Z_{oj} \qquad \forall j \in \mathcal{J}, i \in \mathcal{I}_j, o \in \mathcal{O} | d_{io} = 1 \tag{4}$$

$$X_{ij} + Y_i \geq 1 \qquad \forall (i,j) \in \mathcal{K} \tag{5}$$

$$\sum_{i \in \mathcal{I}} Y_i \leq \alpha \tag{6}$$

$$X_{ij} \in \{0;1\} \qquad \forall j \in \mathcal{J}, i \in \mathcal{I}_j \tag{7}$$

$$Z_{oj} \geq 0 \qquad \forall j \in \mathcal{J}, o \in \mathcal{O} \tag{8}$$

$$Y_i \geq 0 \qquad \forall i \in \mathcal{I} \tag{9}$$

The objective function (1) minimizes the number of visited aisles and the needed replenishments. Note, that the parameter b_{ij} can be adjusted to normalize the effort of visiting an aisle and replenishing a SKU. (2) ensures, that each SKU $i \in \mathcal{I}$ is assigned to exactly one aisle $j \in \mathcal{J}$, while (3) keeps the capacity of each aisle under control. Coupling the assignments with the

5

orders is the task of the restriction (4). After assigning a SKU to any feasible aisle, that aisle is visited when the order $o \in \mathcal{O}$ is picked. The restriction (5) set Y_i to 1, if the SKU $i \in \mathcal{K}$ gets rearranged to another aisle. Limiting the number of rearranged SKUs is achieved by the restriction (6). The second summand of the objective function (1), and the restrictions (5), (6) and (9) extend the model from Kress et al., 2016.

The model formulation is strongly NP-hard and is only solvable for small instances (Kress et al., 2016). Instances like our data set from the beverage retailer need to be solved heuristically. We propose two solution approaches.

4 Solution approaches

4.1 Reduce data set size

Reducing the number of the input orders is our first approach to solve large instances. The core idea is to create a small set of representative orders $\mathcal{O}'$ with similar properties as the original set of orders $\mathcal{O}$. If the set of representative orders is small enough, the previously described model can be solved optimal. The solution obtained with the representative set of orders can be applied and evaluated with the original data set.

Let $\mathcal{S}$ be the set of all order sizes in $\mathcal{O}$. An order size is the number of distinct SKUs in a given order. Let $\mathcal{O}'_s$ and $\mathcal{O}_s$ be the set of orders with a size of s. The representative order set should have a similar structure like the original data set. $|\mathcal{O}'_s| = [u \cdot |\mathcal{O}|]$ with $0 < u < 1$ keeps the proportion of the order sizes similar to the original data set. Using the original data set, h_{s,i_1,i_2} is the parameter counting the number of orders with the size s containing both SKUs i_1 and i_2 with $i_1 \leq i_2$. $h_{s,i,i}$ represents the number of orders of the size s containing the SKU i. The representative data set is desired to have similar properties with $g_{s,i_1,i_2} = [u \cdot h_{s,i_1,i_2}]$.

Let y_{oi} be 1, if the representative order o contains the SKU i and 0 otherwise. Set $y_{oi} = 0$ for all $o \in \mathcal{O}'$ and $i \in \mathcal{I}$ during the initialization. Iteratively select the SKU i' ordered by the order frequency, set $y_{oi'} = 0$ and solve the following model with the binary variable X_o which is 1, if the representative order o contains the SKU i' and 0 otherwise. After solving the model set $y_{oi'} = X_o$ and select the next SKU or start with the first SKU after reaching the last one. Stop the algorithm if the same solution is generated for all SKUs.

$$\text{minimize} \qquad F = \sum_{s \in \mathcal{S}} \Delta A_s + \sum_{s \in \mathcal{S}} \sum_{i \in \mathcal{I}} \Delta B_{si} \tag{10}$$

6

subject to

$$X_o + \sum_{i \in \mathcal{I}} y_{o,i} \leq s \qquad \forall s \in \mathcal{S}, o \in \mathcal{O}'_s \quad (11)$$

$$\Delta A_s \geq g_{s,i',i'} - \sum_{o \in \mathcal{O}'_s} X_o \qquad \forall s \in \mathcal{S} \quad (12)$$

$$-\Delta A_s \leq g_{s,i',i'} - \sum_{o \in \mathcal{O}'_s} X_o \qquad \forall s \in \mathcal{S} \quad (13)$$

$$\Delta B_{si} \geq g_{s,i,i'} + g_{s,i',i} - \sum_{o \in \mathcal{O}'_s} \sum_{i \in \mathcal{I}} y_{oi} \cdot X_o \qquad \forall s \in \mathcal{S}, i \in \mathcal{I} | \setminus i' \quad (14)$$

$$-\Delta B_{si} \leq g_{s,i,i'} + g_{s,i',i} - \sum_{o \in \mathcal{O}'_s} \sum_{i \in \mathcal{I}} y_{oi} \cdot X_o \qquad \forall s \in \mathcal{S}, i \in \mathcal{I} | \setminus i' \quad (15)$$

$$X_o \in \{0; 1\} \qquad \forall o \in \mathcal{O}' \quad (16)$$

$$\Delta A_s, \Delta B_{si} \geq 0 \qquad \forall o \in \mathcal{O}', s \in \mathcal{S} \quad (17)$$

ΔA_s and ΔB_{si} describe the difference of the desired to the realized properties of the representative data set. ΔA_s counts the deviations to the expected number of orders with the size $s \in \mathcal{S}$ containing the currently selected SKU i'. ΔB_{si} similarly counts the deviations to the expected number of orders with the size $s \in \mathcal{S}$ containing both the currently selected SKU i' and the SKU i. Those difference are minimized in the objective function (10). The restriction (11) ensures that each representative order does not contain more than s SKUs. The restrictions (12) to (15) calculate the differences ΔA_s and ΔB_{si}.

4.2 Knapsack formulation

We apply the quadratic multiple knapsack problem in our second solution approach to reduce the computational effort. The idea is to abstract the number of mutual orders for each pair of two SKUs to simplify the problem. See Figure 2 for an example of this abstraction.

7

Order Id	Ordered SKUs		Number of mutual orders	
1	A, A, B, C		A, B:	1
2	A, C, C	$\longrightarrow$	A, C:	2
3	B, D, D		B, C:	1
			B, D:	1

Figure 2: Extraction of the mutual orders for all pairs of SKUs from the set of orders.

Afterwards, we formulate a model which maximizes the number of mutual orders over all aisles. Only if two SKUs are stored in the same aisle, the number of mutual orders of those two SKUs counts. This approach is described in the section 4.2.1. We extend the solution approach for the special case that $\alpha \geq |\mathcal{K}|$. This means that there is no limitation of how many products can be rearranged. In this case we add the 2-opt algorithm to the solution of the quadratic multiple knapsack problem. This approach is described in the section 4.2.2.

The knapsack approach is a heuristic and may not find the optimal solution. Figure 4 illustrates a small example in which the optimal solution is not found. In this example the number of replenishments is the same for all SKUs and storage group. Additionally, each SKUs can feasibly be stored in each storage group and each storage group has the capacity for three SKUs. The SKUs E and F are not ordered but still need to be assigned to either storage group.

Order Id	Ordered SKUs
1	$\boxed{A}$, $\boxed{B}$, $\boxed{C}$
2	$\boxed{A}$, $\boxed{B}$, $\boxed{C}$
3	$\boxed{A}$, $\boxed{D}$
4	$\boxed{A}$, $\boxed{D}$
5	$\boxed{A}$, $\boxed{D}$
6	$\boxed{E}$
7	$\boxed{F}$

	Assignment 1	Assignment 2
Storage group 1	$\boxed{A}\boxed{B}\boxed{C}$	$\boxed{A}\boxed{D}\boxed{E}$
Storage group 2	$\boxed{D}\boxed{E}\boxed{F}$	$\boxed{B}\boxed{C}\boxed{F}$
Storage group visits	10	9
Knapsack objective	6	5

Figure 3: Example for a bad case for the abstraction of mutual orders.

4.2.1 SKU aisle rearrangement

Let $p_{i_1 i_2}$ be the number of mutual orders of the SKUs i_1 and $i_2 \in \mathcal{I}$. The following model maximizes the sum over the number of mutual orders of all possible SKU-pairs in every aisle:

$$\text{maximize } F = \sum_{j \in \mathcal{J}} \sum_{i_1 \in \mathcal{I}_j} \sum_{\substack{i_2 \in \mathcal{I}_j \\ |i_1 < i_2}} p_{i_1,i_2} \cdot X_{i_1 j} \cdot X_{i_2 j} - \beta \sum_{j \in \mathcal{J}} \sum_{i \in \mathcal{I}_j} b_{ij} \cdot X_{ij} \tag{18}$$

subject to

$$\tag{19}$$

$$\sum_{j \in \mathcal{J}} X_{ij} = 1 \qquad \forall i \in \mathcal{I} \tag{20}$$

$$\sum_{i \in \mathcal{I}_j} X_{ij} \leq c_j \qquad \forall j \in \mathcal{J} \tag{21}$$

$$X_{ij} + Y_i \geq 1 \qquad \forall (i,j) \in \mathcal{K} \tag{22}$$

$$\sum_{i \in \mathcal{I}} Y_i \leq a \tag{23}$$

$$X_{ij} \in \{0;1\} \qquad \forall j \in \mathcal{J}, i \in \mathcal{I}_j \tag{24}$$

$$Y_i \geq 0 \qquad \forall i \in \mathcal{I} \tag{25}$$

9

The objective function (18) adds the number of mutual orders if both SKUs i_1 and i_2 are assigned to the aisle $j \in \mathcal{J}$. The sum of mutual orders cannot be meaningfully compared with the number of replenishments. β normalize those two values in the objective function and is initially set to 1. Solve the model, count the number of visited storage groups with the resulting solution and set $\beta = \frac{\text{sum of mutual orders}}{\text{sum of visited storage groups}}$. Repeat this process until the same solution is obtained after two consecutive iterations. The comparison of both values in the objective function is iteratively adjusted to make them comparable. All restrictions are adopted from the original model.

4.2.2 SKU aisle assignment

This solution approach assumes, that there are no restrictions on the number of rearranged SKUs: $\alpha \geq |\mathcal{K}|$. Therefore, the restrictions (22) and (23) can be removed from the model introduced in the section 4.2.1. Afterwards, the solution of the quadratic multiple knapsack problem is improved by executing the 2-opt algorithm. In optimization, the 2-opt algorithm swaps the solution of every possible pair if it improves the solution and finds use in many applications (Muyldermans et al., 2005 or McGovern and Gupta, 2004 or Borgulya, 2008). See Figure 4 for an illustration of this solution approach.

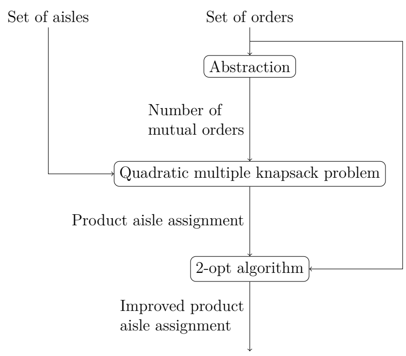


Figure 4: Visualization of the solution approach.

In this application, the 2-opt algorithm checks if switching the position of two SKUs improves the solution. The 2-opt algorithm utilizes directly the set

10

of orders and reintroduces information lost by the abstraction of calculating the number of mutual orders. Swapping the aisles of two different SKUs improves the solution, if fewer aisles need to be visited when picking the set of orders. The algorithm stops if no possible swap improves the solution. Only SKUs of different aisles are evaluated for swapping, since a swap in the same aisle does not affects the solution. Furthermore, only feasible aisles for the SKUs are validated to prevent that a SKUs is assigned to an infeasible aisle. To improve the computation time only a subset of orders is used for the evaluation, since orders without a demand for one of the two swapped SKUs are considered as irrelevant.

4.3 Solution evaluation

The assortment and the frequency of ordered SKUs may change over time. To determine how a assignment of SKUs to aisles performs over time the set of orders testing an assignment can be different from the set of orders used for calculating the assignment. We call the set of orders used for calculating the assignment the set of training orders and the set of orders used for the evaluation the set of testing order. See Figure 5 for a visualization of this evaluation approach.

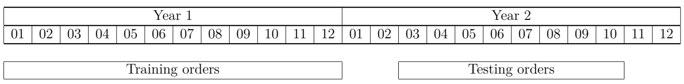


Figure 5: The set of training orders (all orders of the first year) are used for calculating the assignment which is evaluated with the set of testing orders (orders from march to october of the second year).

New SKUs in the testing orders, which have not been ordered in the training set, are randomly assigned to any feasible aisle. Afterwards, the number of visited aisles are counted when picking the testing orders.

5 Computational study

5.1 Artificial data set

A randomly generated data set is used for the first part of the computational study. 20 SKUs in 10000 orders each containing three to seven SKUs are generated in three variants. SKUs of the first variant have no correlation and their probabilities to appear in an order are independent from each other. The second variant implements a medium and the third variant a high correlation between the SKUs. A computer with an Intel Xeon CPU E5-2667 v3 processor, 256gb RAM and CPLEX as a solver has been used for all studies in this paper.

The model formulation calculates the optimal solution, but its not practicable to solve instances with 10000 orders. Up to 900 orders, randomly taken from the 10000 orders, are used to measure the computation time of the optimal formulation approach. Figure 6 illustrates the computation time to find the optimal solution for up to 900 orders with the original model formulation. More time is needed to solve larger instances. Solving the data set variant without correlation between the SKUs is usually the quickest.

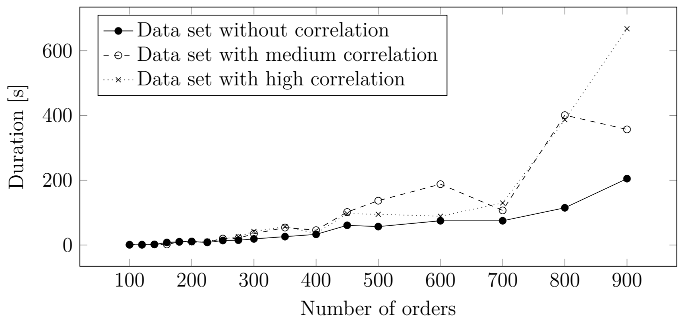


Figure 6: Computation time for the optimal formulation approach with up to 900 orders.

Solving those instances with the knapsack approach takes not more than 2 seconds. Figure 7 displays the gap of the knapsack approach to the optimal solution for instances with up to 900 orders. No clear trend for the data set variants or the number of orders can be found. The solution of the knapsack approach is never more than 2% worse than the optimal solution. The 2-opt algorithm never found an improvement for any of those solutions.

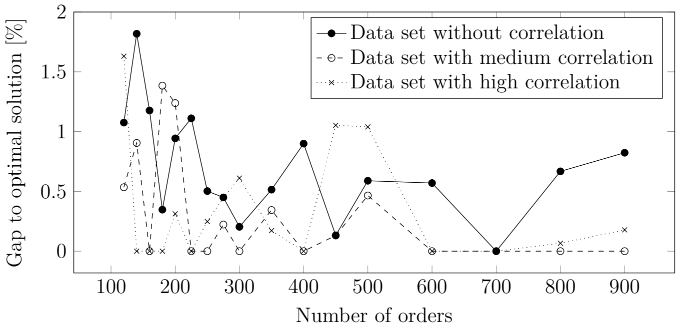


Figure 7: Optimality gap of the knapsack approach on instances with up to 900 orders

Figure 8 compares the same solutions on 9000 new orders. The solutions obtained with up to 900 input orders are evaluated with the remaining 9000 orders, while all orders are generated by the same distribution. A negative gap means that the solution of the knapsack approach performs better with the new orders than the solution which is optimal on the training orders. No clear trend can be seen.

13

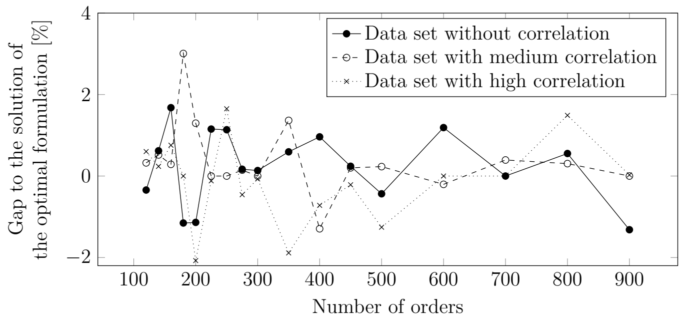


Figure 8: The solutions are generated by the knapsack approach and the optimal formulation with up to 900 orders. Those solutions are evaluated on 9000 new orders.

Only the knapsack and the reduce data set size approach are used to find a solution for all 10000 orders. Figure 9 illustrates the computational time for all three data set variants depending on number of representative orders with the reduce data set size approach. Generating the representative orders takes a couple of seconds and most of the time is usually needed to optimally solve the assignment over the set of representative orders. The computation time increases with the number of representative orders. Solving the instances with the knapsack approach took 30 seconds for all three data set variants. Improving the solution with the 2-opt algorithm over all 10000 orders takes less than a second.

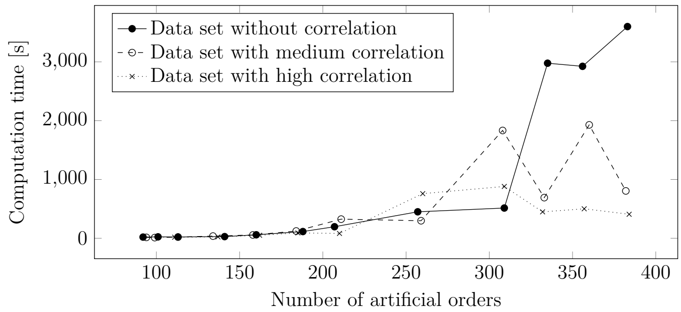


Figure 9: Computation time for the reduce data set size approach.

Figure 10 compares the solutions of the reduce data set size approach with the knapsack approach, after improving all solutions with the 2-opt algorithm. All solutions are, with a maximum difference of less than 0.3%, very close to each other. Increasing the number of representative orders does not necessarily improves the solution. None of the three variants consistently delivers better solutions when reducing the data set size.

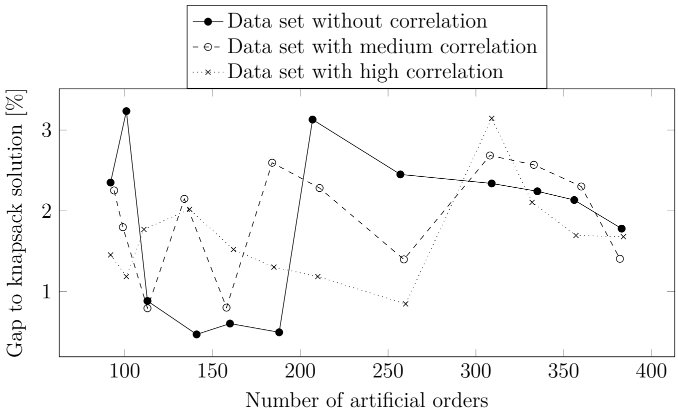


Figure 10: Performance of the reduce data set size approach compared to the knapsack approach, both improved by the 2-opt algorithm.

5.2 Data set of a beverage retailer

The data set of the beverage retailer contains more than 1000 SKUs which are assigned to 15 aisles. All 356451 orders from August 2017 until the August 2019 are given.

The orders from the first year are used to calculate a solution with all solution approaches. Those solutions are evaluated in Figure 11 for the first and second year. It is not possible to obtain the optimal solution in a reasonable amount of time. Therefore, the computation of the original model formulation was stopped after one hour. Only one minute was enough to get the solution of the knapsack approach and reducing the order set size took 45 minutes while reducing the number of orders to 1000. Improving the solution with the 2-opt algorithm took around one hour for all solution approaches. The knapsack approach finds the solution which performs the best in both years, especially when improved with the 2-opt algorithm. All objective values are closer to each other when the solution is evaluated with the orders of the second year.

16

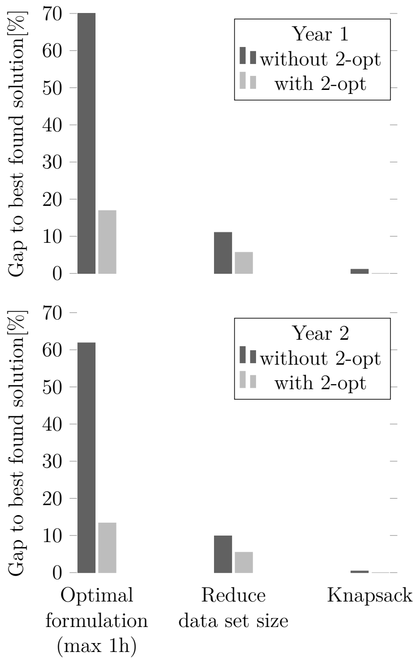


Figure 11: Solution performance when optimizing for the orders of the first year from the beverage retailer.

Two further studies have been conducted to gain managerial insights. Firstly, we studied the possible improvements when rearranging the assignment in the warehouse after one year. The knapsack solution with the orders from the first year has been evaluated with the orders of the second year. Because new SKUs come to the assortment and structural order changes of already existing SKUs occur, the solution from the first year may leave room for improvement. An improvement of 2.7% can be found when the knapsack approach is applied with the orders from the second year. Therefore, regular warehouse rearrangements can improve the assignment of SKUs to storage

17

groups. When rearranging all SKUs in the warehouse is not possible, switching a few SKUs can still benefit the assignment. Figure 12 illustrates the fraction of this improvement gained, when rearranging only a certain fraction of the SKUs in the warehouse. Limiting the number of rearrangements to 30 (2.6% of the number of SKUs stored in the warehouse) yields already 22.5% of the possible improvements.

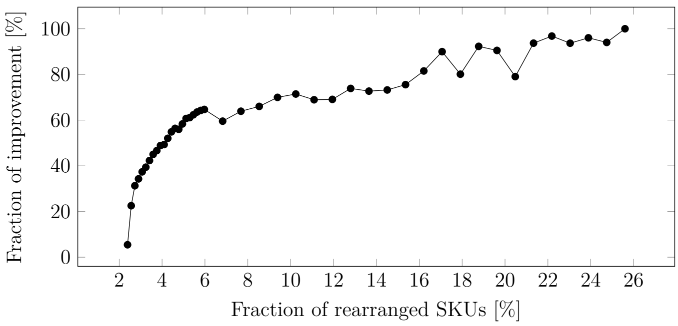


Figure 12: Solution performance when optimizing for the orders of the first year from the beverage retailer.

The second study looks into the frequency of rearrangements. The knapsack solution with the orders from the first year (month 1 to 12) is the baseline for the comparison. A second solution is obtained with the order from month 2 to 13. Both solutions are evaluated on the 14th month. This comparison is conducted until orders from month 12 to 23 is evaluated with the order from month 24 as illustrated in Figure 14.

18

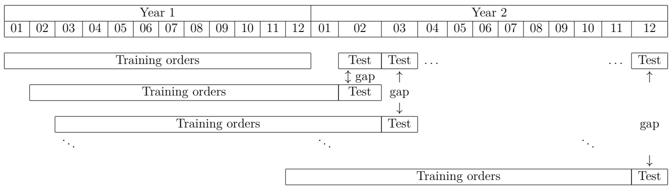


Figure 13: Schema of the computational study presented in Figure 14.

Figure 14 illustrates the improvement of rearranging the assignments with the most recent orders. Utilizing the most recent information about the past orders and adjusting the warehouse assignments accordingly can reduce the order picking effort. The performance of an assignment of SKUs to storage groups diminishes over time.

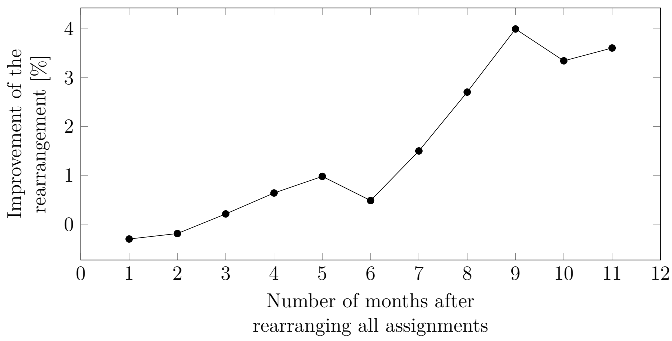


Figure 14: Improvement after rearranging.

6 Conclusion

We formulated a quadratic multiple knapsack problem to solve the assignment of SKUs to aisles, while considering replenishments and rearrangements

19

based on an already existing assignment. The 2-opt algorithm can improve the solution of the quadratic multiple knapsack problem, if there are no limitations on the number of rearranged SKUs compared to the already existing assignment. Furthermore, we introduced a method to generate a representative set of orders to reduce the size of instances.

Using past orders to prepare the assignment for future orders is a good indicator, but a changing assortment and trends in the order patterns of the SKUs worsen formerly assignments. Based on a real data set we show that rearranging the warehouse at least once a year can reduce the objective value by 2.7%. Reducing the size of the set of input orders does not yet deliver better solutions than the multiple knapsack problem formulation. A more sophisticated approach to generate the set of representative orders can be studied in further research to improve the representativeness. Even if the solution of the quadratic multiple knapsack problem is 2% worse than the optimal approach, it quickly finds a solution even for big instances. Another advantage is the simplified input, because it merely requires the number of mutual orders of each pair of SKUs instead a big set of orders. This may be handy if, as part of a forecast, the assignments of SKUs in a warehouse are planned for future orders. Furthermore, rearranging only a subset of all SKUs in the warehouse leads to an improvement. In our study limiting the number of rearranged products to 2.6% of the assortment already lead to 22.5% of the possible improvements. A derived warehouse strategy is rearranging regularly a small fraction of the assortment in the warehouse.

In further research the solution of the quadratic multiple knapsack problem can be compared to other solution approaches in the literature. Researching the influence of multiple different demand patterns on the frequency and scale of rearrangements is another possibility to gain valuable insights. Further research on the number of required orders when the solution of the quadratic multiple knapsack problem becomes stable is of interest. The number of SKUs in the warehouse as well as the average number of SKUs per order may be of relevance.

References

Ang, M., Lim, Y. F., & Sim, M. (2012). Robust storage assignment in unit-load warehouses. *Management Science*, *58*(11), 2114–2130. https://doi.org/10.1287/mnsc.1120.1543

Borgulya, I. (2008). An algorithm for the capacitated vehicle routing problem with route balancing. *Central European Journal of Operations*

Research, *16*(4), 331–343. https://doi.org/10.1007/s10100-008-0062-2

Chiang, D. M.-H., Lin, C.-P., & Chen, M.-C. (2011). The adaptive approach for storage assignment by mining data of warehouse management system for distribution centres. *Enterprise Information Systems*, *5*(2), 219–234. https://doi.org/10.1080/17517575.2010.537784

Chiang, D. M.-H., Lin, C.-P., & Chen, M.-C. (2012). Data mining based storage assignment heuristics for travel distance reduction. *Expert Systems*, *31*(1), 81–90. https://doi.org/10.1111/exsy.12006

de Koster, R., Le-Duc, T., & Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, *182*(2), 481–501. https://doi.org/10.1016/j.ejor.2006.07.009

Hiley, A., & Julstrom, B. A. (2006). The quadratic multiple knapsack problem and three heuristic approaches to it. *Proceedings of the 8th annual conference on Genetic and evolutionary computation - GECCO '06*. https://doi.org/10.1145/1143997.1144096

Kress, D., Boysen, N., & Pesch, E. (2016). Which items should be stored together? a basic partition problem to assign storage space in group-based storage systems. *IISE Transactions*, *49*(1), 13–30. https://doi.org/10.1080/0740817x.2016.1213469

Litvak, N., & Vlasiou, M. (2010). A survey on performance analysis of warehouse carousel systems. *Statistica Neerlandica*, *64*(4), 401–447. https://doi.org/10.1111/j.1467-9574.2010.00454.x

McGovern, S. M., & Gupta, S. M. (2004). 2-opt heuristic for the disassembly line balancing problem. In S. M. Gupta (Ed.), *Environmentally conscious manufacturing III*. SPIE. https://doi.org/10.1117/12.516155

Muyldermans, L., Beullens, P., Cattrysse, D., & Oudheusden, D. V. (2005). Exploring variants of 2-opt and 3-opt for the general routing problem. *Operations Research*, *53*(6), 982–995. https://doi.org/10.1287/opre.1040.0205

Pisinger, D. (2007). The quadratic knapsack problem - a survey. *Discrete Applied Mathematics*, *155*(5), 623–648. https://doi.org/10.1016/j.dam.2006.08.007

Zhang, R.-Q., Wang, M., & Pan, X. (2019). New model of the storage location assignment problem considering demand correlation pattern. *Computers & Industrial Engineering*, *129*, 210–219. https://doi.org/10.1016/j.cie.2019.01.027

Zhang, Y. (2016). Correlated storage assignment strategy to reduce travel distance in order picking. *IFAC-PapersOnLine*, *49*(2), 30–35. https://doi.org/10.1016/j.ifacol.2016.03.006

2.3 Application of a multi-objective multi traveling salesperson problem with time windows

Application of a multi-objective multi traveling salesperson problem with time windows

Justus Bonz

Abstract

The pilgrimage to Mecca, which is called Hajj, is the largest annual pedestrian crowd management problem in the world. During the Hajj, the pilgrims are accommodated in camps. For safety reasons, exact times and directions are given to the pilgrims who are moving between holy sites. Despite the importance of complying with those schedules, violations can often be conjectured. Directing a small workforce between the camps to monitor the pilgrims' compliance with the schedule is an important matter, which will be dealt with in this paper. A type of multi-objective multiple traveling salesperson optimization problem with time windows is introduced to generate the tours for the employees monitoring the flow of pilgrims at the campsite. Four objectives are being pursued: As many pilgrims as possible (1) should be visited with a preferably small workforce (2), the tours of the employees should be short (3) and employees should have short waiting times between visits (4). A goal programming, an enumeration and an interactive approach are developed. The topic of supported and non-supported efficient solutions is addressed by determining all efficient solutions with the enumeration approach. The suitability of the approaches are analysed in a computational study, while using an actual data set of the Hajj season in 2015. For this application, the interactive approach has been identified as the most suitable approach to support the generation of an offer for the project.

Keywords: multi-objective multiple traveling salesperson, time windows, Hajj, mass gatherings, non-supported efficient solutions

1 Background

Each year the Hajj attracts two to four million people. See [Haa+16] for a detailed description of the pilgrimage. During the different movements of pilgrims between the holy sites the pilgrims can either walk, use a bus or the Mecca Metro. See Figure 1 for images of the metro. The movements of the pilgrims are illustrated in Figure 2.

Before moving between different locations most of the pilgrims are located in camps in Mina. See Figure 3 for images of the tent city of Mina. Only registered ticket holders in dedicated camps are permitted to use the metro. Between 300 000 and 400 000 pilgrims use the metro each Hajj season. There are three metro stations in Mina, Muzdalifah, and Arafat, respectively. They are illustrated in Figure 4. See Figure 5 for images of two metro stations. To prevent congestion and to assure a steady flow of pilgrims from their camps to

Figure 1: Images of the Mecca Metro.

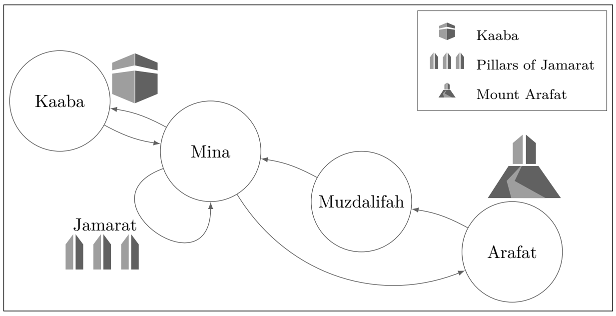


Figure 2: Movements of the pilgrims in the region of Mecca during the Hajj.

the metro stations a schedule is distributed. The schedule for a camp specifies the departure time and the path to the metro station. Since all the pilgrims of a camp cannot depart instantaneously, the expected duration needed for the pilgrims leaving the camp is given. The number of pilgrims in each camp is known.

The compliance of this schedule is crucial [HJAA07]. However, it is still frequently violated, which leads to the necessity of identifying those violations. An analysis of the pilgrims' arrival time at the metro station during the Hajj in 2016 indicates major problems with the schedule compliance, which is illustrated in Figure 6. It shows the fraction of pilgrims arriving on time at the corresponding metro stations. Only 36% of all pilgrims reached their metro station at the scheduled time. The compliance has been calculated with a 15-minute time buffer before and after each scheduled arrival time.

The goal of this paper is to direct a small and homogeneous workforce between the camps and detect which camps violate the schedule. It is not strictly necessary to observe all camps, but as many as possible with a reasonably scaled workforce. The employees of the workforce start their duty at any metro station and can only visit camps assigned to that station. This reduces the complexity

2

Figure 3: Tent city of Mina.

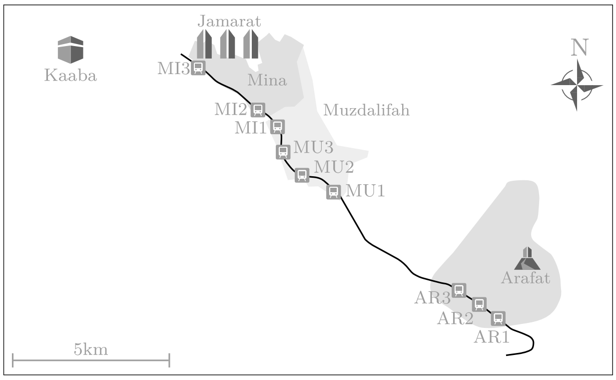


Figure 4: Illustration of the metro stations in Mina (MI), Muzdalifah (MU) and Arafat (AR).

for the employees of finding the correct camps in the tent city and empowers them to guide lost pilgrims to their correct metro station. A schematic representation of a metro station and some associated camps are given in Figure 7.

The employees operate a smartphone with an application that instructs the employees to move to certain camps. After the arrival, the employees monitor whether the pilgrims already left, are leaving or are still waiting for their departure and enter this information into the smartphone.

Due to camp sizes of a few thousand pilgrims it is expected that the dispatching process lasts a certain period of time. To ensure that the observed dispatching status of the camp is correct, the employee is bound to stay a certain fraction of the duration of the dispatching process at the camp. The employee's arrival time is negligible if they stay the given time during the departure.

The components of this project have already been developed and tested during the Hajj in 2015. For this prototype, a workforce was gathered and they have been directed to the camps to monitor the dispatching process. They used

Figure 5: Aerial view of two metro stations.

a mobile application to send the dispatching status to a central server, which stores the incoming data. Figure 8a is a screenshot of the mobile application. After receiving the dispatching statuses at the server, the results have been analysed. The mobile application stored the location. An example track of an employee is displayed in Figure 8b. The pilgrims of only 30% of the camps departed as scheduled during their scheduled departure time. Walking to a camp and sending the dispatching status to a server is called a task. A tour consists of multiple tasks for a single employee.

A set of tours can be evaluated regarding four different goals. The *first goal* is to observe the dispatching process of as many pilgrims as possible. The *second goal* is to minimize the number of employees. The *third goal* is minimizing long waiting times between two tasks for the employees and the *fourth goal* is to minimize the total distance the employees must walk.

An offer for the responsible metro operator in Saudi Arabia should be made. This offer should contain a bundle of different solutions. Each solution contains a set of tours with their individual performance for the four goals. The offer receiver wants to be involved in the decision-making process and select their preferred solution without being overwhelmed by too many possible solutions. For the numerical studies, the real data of the camps' locations as well as their distances are used.

The remainder of this paper has the following structure: The second section addresses literature relevant for this paper. In section 3, a mixed-integer formulation is introduced. The results of a numerical study are presented in section 4. First, the goal programming approach is described, followed by an enumeration approach. An interactive approach is the solution approach discussed at last. All solution approaches use the actual schedule data. This includes the actual location of the camps with their corresponding number of accommodated pilgrims as well as their scheduled departure time. Finally, a conclusion is drawn in section 5.

2 Literature

Multi-objective combinatorial optimization (MOCO) is the simultaneous approach of several objectives in a finite but often large set of feasible solutions [CJ98]. Those objectives are usually conflicting [EG08]. The general MOCO

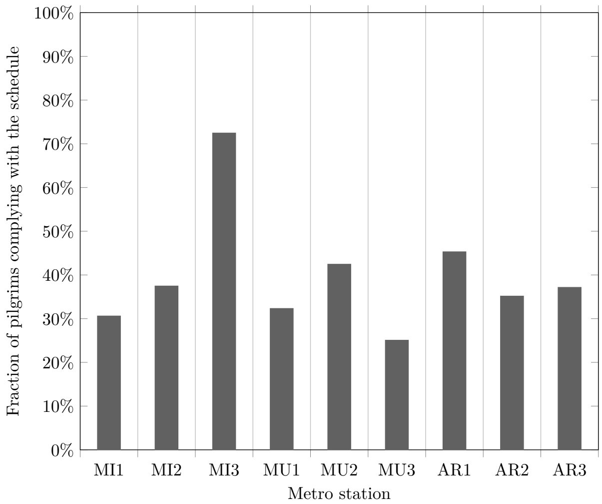


Figure 6: Analysis of the pilgrims' arrival time at the metro stations during the Hajj 2016 (MU: Muzdalifah, MI: Mina, AR: Arafat).

framework can be formulated as

$$\text{minimize}\{f_1(x), \ldots, f_{|K|}(x)\} \tag{1}$$

subject to $x \in S$, where S is a finite set of feasible solutions and the solution x a vector of discrete decision variables. K is the set of objective functions with $|K| \geq 2$. MOCO problems find a lot of applications and are therefore well studied [Ulu+99]. A solution optimizing every objective simultaneously does generally not exist, but the best compromise for a decision maker (DM) can be found [EG04]. The best compromise of a DM is part of the set of efficient solutions and depends on their utility function [Jas02]. A solution is efficient (or Pareto-optimal) if there does not exists another feasible solution that does not perform worse in any objective and better in at least one objective [TTU00].

The set of efficient solutions of a MOCO problem can be divided into supported and non-supported efficient solutions [EG00]. Let f_k be the objective value and λ_k the weight for the objective function k with $\lambda_k \geq 0 \forall k \in K$ and $\sum_{k \in K} \lambda_k = 1$. When using the single objective function:

$$\text{minimize} \sum_{k \in K} \lambda_k f_k \tag{2}$$

only the set of supported efficient solutions can be obtained [UT94]. The set of non-supported efficient solutions therefore contains the solutions that cannot be

5

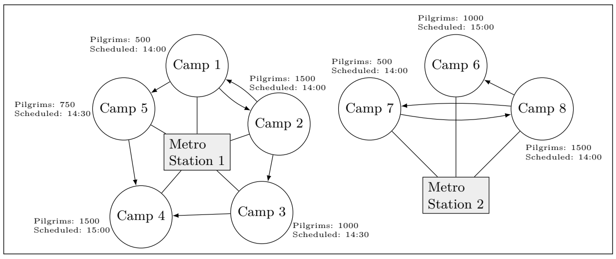


Figure 7: Metro station with camps illustration.

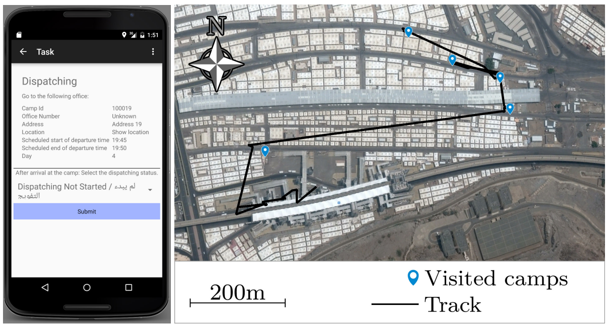
found with this single objective function. For a non-supported efficient solution x' compared to the set of supported efficient solutions S no set of weights can be found, so that:

$$\sum_{k \in K} \lambda_k f_k(x') \leq \sum_{k \in K} \lambda_k f_k(x) \qquad x \in S \tag{3}$$

There are several approaches for MOCO problems. Goal programming is a well known technique even though only supported efficient solutions can be obtained. Goal programming minimizes the sum of the deviations of the individual objective values from their target value [TJR98]. Knowledge of the utility function of the DM is required to weight the deviations of the different objectives [Dye72]. Gilbert et al. proposed an interactive method, which generates efficient solutions with the integration of the DM in the optimization process [GHR85] iteratively. Two more interactive approaches based on local optimality have been proposed by Paquete, Schiavinotto, and Stützle [PSS07]. Yet various genetic algorithm approaches like [KCS06] and [FF95] other methods have been developed to find the approximated set of efficient solutions. Ulungu et al. describe a multiobjective simulated annealing method to approximate the set of efficient solutions, since the problems may be too complex for exact methods [Ulu+99]. A genetic local search approach was proposed by Jaszkiewicz to generate a set of approximated efficient solutions even for relatively large instances [Jas02].

The multiple traveling salesperson problem (mTSP), a generalization of the traveling salesperson problem (TSP), describes the determination of multiple tours that cover all cities and each salesperson starts and ends their journey in the same city [Bek06]. When adding time window contraints each city can only be served after a permitted starting time and before a permitted ending time [Gen+98]. The multi-objective TSP addresses two or more objectives commonly minimizing the number of tours, the total time required and the total tour cost [JST08]. Bowerman, Hall and Calamai proposed an approach to the urban school bus routing problem. Multiple objectives are minimized, e.g. the total bus route length, the remaining walking distance for the students, as well as balancing the length of the routes [BHC95]. Many heuristics have been developed to solve multi-objective TSP. Hansen illustrated the applicability of using substitute scalarizing functions to guide a tabu search heuristic [Han00]. A genetic

6



(a) Screenshot of the mobile application.

(b) Tracks of an employee visiting camps.

Figure 8: Figures from the pilot project.

algorithm has also been used to solve the multi-objective TSP [Jas02]. Florios and Mavrotas used a multi-objective mathematical programming method to generate the exact Pareto set in the multi-objective TSP [FM14].

3 Multi-objective multiple traveling salesperson problem with time windows

The problem can be formulated as a directed graph, see Figure 7. The camps and metro stations are represented by nodes. The arcs represent compatible connections between camps and metro stations. As stated in the first section, not every camp must be visited, since a reasonable workforce should be found. Furthermore, it is assumed that the employees start their duty at a metro station and can only visit camps whose pilgrims are assigned to that station. Restrictions concerning the working hours are not taken into consideration. The following notation formalized the problem as a mathematical model:

Sets

I	Nodes	
C	Camps	$C \subset I$
D	Metro stations	$D \subset I, C \cap D = \emptyset$
C_i	Camps associated to metro station $i \in D$	$C_i \subset C$
E	Set of arcs connecting compatible nodes	
E^H	Set of arcs with waiting times longer than t^H	$E^H \subset E$
E^L	Set of arcs connecting camps with similar starting times	$E^L \subset E$
I_i^p	Set of immediate predecessor nodes of node $i \in I$	
I_i^s	Set of immediate successor nodes of node $i \in I$	
K	Set of objective functions	

Parameters

p_i	Number of pilgrims located in camp $i \in C$
s_i	Scheduled departure time for pilgrims of camp $i \in C$
w_{ij}	Time needed to walk from node $i \in I$ to $j \in I$
a_i	Monitoring time needed for an employee at camp $i \in C$
$t_i^{\max}$	Maximum allowed delay for an employee at the camp $i \in C$
t^H	Maximum waiting time

Variables

x_{ij}	1 if node $i \in I$ is visited directly after node $j \in I$, 0 otherwise
t_i	time after s_i when a camp $i \in C$ is visited

The objective functions $K = \{\text{pilgrims}, \text{employees}, \text{wait}, \text{walk}\}$ are defined as the following:

$$\text{minimize} \quad f_{\text{pilgrims}} = \sum_{j \in C} \sum_{i \in I_j^p} p_j (1 - x_{ij}) \tag{4}$$

$$\text{minimize} \quad f_{\text{employees}} = \sum_{i \in D} \sum_{j \in C_i} x_{ij} \tag{5}$$

$$\text{minimize} \quad f_{\text{wait}} = \sum_{(i,j) \in E^H} (s_j - (s_i + a_i + w_{ij})) x_{ij} \tag{6}$$

$$\text{minimize} \quad f_{\text{walk}} = \sum_{i \in I} \sum_{j \in I_i^s} w_{ij} x_{ij} \tag{7}$$

where (4) minimizes the number of pilgrims who are not visited, (5) the number of employees in duty, (6) the sum of all waiting times between two tasks, which are longer than t^H and therefore considered as too long and (7) minimizes the total walking time over all employees.

The following constraints must be satisfied:

$$\sum_{i \in I_j^p} x_{ij} \leq 1 \qquad \forall j \in C \tag{8}$$

$$\sum_{j \in I_i^s} x_{ij} \leq 1 \qquad \forall i \in C \tag{9}$$

$$\sum_{j \in I_i^s} x_{ij} - \sum_{j \in I_i^p} x_{ji} = 0 \qquad \forall i \in I \tag{10}$$

$$(s_i + a_i + w_{ij}) x_{ij} + t_i \leq s_j + t_j \qquad \forall (i,j) \in E^L \tag{11}$$

$$x_{ij} \in \{0; 1\} \qquad \forall i \in I, j \in I_i^s \tag{12}$$

$$0 \leq t_i \leq t_i^{\max} \qquad \forall i \in C \tag{13}$$

The constraints (8), (9), and (10) are well-known network flow constraints. The constraint (11) detains short cycles and ensures a correct order of visits. Those constraints do not disable any connection if x_{ij} is zero and $t_i \leq s_j + t_j$ remains. It is only a matter of maintaining a correct time reference point so that $s_j > t_i^{\max} \forall i, j \in C$. The subtour elimination constraints can be lifted [DL91]:

$$u_i - u_j + (|C| - 1) x_{ij} + (|C| - 3) x_{ji} \leq |C| - 2 \qquad \forall i, j \in E^L | i \neq j \tag{14}$$

with $1 \leq u_i \leq |C| - 1 \quad \forall i \in C$ to strengthen the formulation.

Figure 9 illustrates the parameters and variables for the movement from camp i to camp j with $i, j \in C$. The graph consists of camps and metro

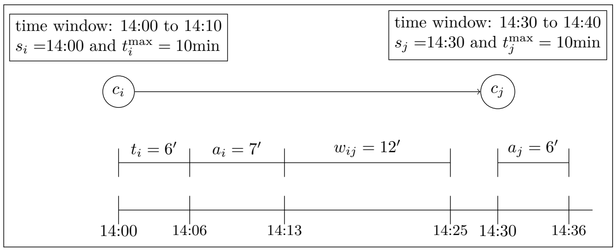


Figure 9: Illustration of the variables and parameters for the movement from camp $i \in C$ to camp $j \in C$.

stations $I = D \cup C$ and $D \cap C = \emptyset$. A camp can only be associated to a single metro station $C_i \cup C_j = \emptyset \quad \forall i, j \in D, i \neq j$. The arcs of the graph describe possible connections between the nodes. Two camps associated to the same depot $d \in D$ are connected if it is possible to reach the destination camp before the time window closes:

$$E = \{(i,j) | (i,j) \in C_d \wedge s_i + a_i + w_{ij} \leq s_j + t_j^{\max}\} \tag{15}$$

Every camp is always connected to its associated metro station. There are two subsets of arcs E, which contain arcs with certain properties. The first subset E^H connects nodes with long waiting times. They should be avoided if possible, because long waiting times for employees are disadvantageous:

$$E^H = \{(i,j) | i, j \in C \wedge s_j - s_i - a_i - w_{ij} \geq t^H\} \tag{16}$$

The second subset E^L connects nodes with similar starting times. Their time windows overlap and the order when a camp is visited may change depending on the realizations of t_i and t_j:

$$E^L = \{(i,j) | i, j \in C \wedge s_i + a_i + w_{ij} + t_i^{\max} \geq s_j\} \tag{17}$$

Many computational studies have been conducted to evaluate the scalability of such problems ([Sol87], [CLM01] or [MC16]). The focus of this paper is the suitability of different approaches for MOCO problems.

4 Solution approaches

4.1 Data description and model performance

This paper is based on the real data set containing 3 metro stations and 104 camps accommodating 178131 pilgrims. All departures are scheduled within 8 hours. The average time to walk from one camp to another (w_{ij}) is about 10 minutes, and the average monitoring duration (a_i) is 8 minutes. On average, a camp must be visited not later than 7 minutes after the scheduled departure.

9

A computer with an Intel Xeon CPU E5-2667 v3 processor, 256gb RAM and CPLEX as a solver has been used for all studies in this paper.

Solving a single instance of the model with this data does not take longer than a second. Subtours are only possible when two camps with similar starting times and long monitoring durations are close together. This is rarely the case since the set E^{L} contains only 111 arcs. Lifting the subtour elimination constraints with (14) does not reduce the calculation time for this instance. The stronger formulation becomes more important when many camps have similar starting times, because they enable more possible subtours.

4.2 Goal programming

The goal programming approach is one method to address multiple objectives [TJR98]. All objective values are combined in a single objective function. This approach requires the knowledge of the objective weights of the DM to assess the deviation of the individual objectives from their optima. The objective function minimizes the sum of deviations of the objective k from its individual optima f_k^* weighted with $\bar{\lambda}$:

$$\text{minimize} \quad F = \sum_{k \in K} \bar{\lambda}_k |f_k^* - f_k| \tag{18}$$

Since $f_k^* = 0 \quad \forall k \in K$ in our application, the objective function can be simplified:

$$\text{minimize} \quad F = \sum_{k \in K} \bar{\lambda}_k f_k \tag{19}$$

This simplification leads to a normalized weighted objective function and is a special case of the goal programming approach, since the individual optima of the objective are zero. In order to make a comparison between the deviations of the objectives easier for the DM, the weights can be replaced to normalize the objective value by its upper bound $\bar{\lambda}_k = \frac{\lambda_k}{\kappa_k}$. λ_k is the weight of the DM for the normalized objective value $k \in K$. The calculation of the upper bounds κ_k is listed in Table 1. The resulting objective function minimizes the weighted sum of normalized objective values:

$$\text{minimize} \quad F = \sum_{k \in K} \lambda_k \frac{f_k}{\kappa_k} \tag{20}$$

Table 1: Calculation of κ_k.

Objective	κ_k	Description
pilgrims	$\sum_{i \in C} p_i$	no pilgrim is visited
employees	$\lvert C \rvert$	one employee per camp
walk	$\sum_{i \in D} \sum_{j \in C} (w_{ij} + w_{ij})$	pendulum tours
slack	$\max f_{\text{slack}}$	maximum slack

The weights λ_k of the DM are unknown. Testing multiple sets of weights can give valuable insights in the structure of the efficient frontier. Systematically varying all goals in certain steps while ensuring that $\sum_{k \in K} \lambda_k = 1$ delivers a

Table 2: Results of different step sizes for generating the weights for the goal programming approach.

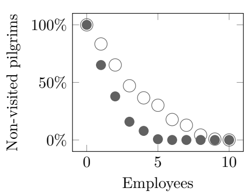
Step size	Instance count	Distinct efficient solutions	Calculation time
100%	4	2	00 : 00 : 01
50%	10	4	00 : 00 : 03
20%	56	24	00 : 00 : 13
10%	286	102	00 : 01 : 09
5%	1 771	220	00 : 07 : 32
2%	23 426	436	01 : 33 : 05
1%	176 851	533	11 : 02 : 41

Table 3: An overview of the weights for a step size of 10% for the goal programming approach.

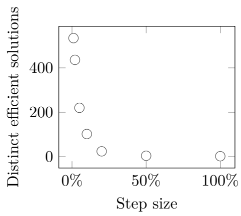
#	$\lambda_{\text{pilgrims}}$	$\lambda_{\text{employees}}$	λ_{walk}	λ_{slack}
1	0	0	0	1
2	0	0	0.1	0.9
3	0	0	0.2	0.8
4	0	0	0.3	0.7
⋮	⋮	⋮	⋮	⋮
153	0.2	0.4	0.1	0.3
154	0.2	0.4	0.2	0.2
155	0.2	0.4	0.3	0.1
⋮	⋮	⋮	⋮	⋮
283	0.9	0	0	0.1
284	0.9	0	0.1	0
285	0.9	0.1	0	0
286	1	0	0	0

set of weights with constant intervals. Decreasing the step size increases the number of weight combinations. With a step size of 10% each combination of the weights λ_k in $\{0; 0.1; 0.2; \ldots; 0.9; 1\}$ is applied to the objective values as long as $\sum_{k \in K} \lambda_k = 1$. An extract of the weights for a step size of 10% is listed in Table 3. All possible combinations of weights changing by 10% with a sum of 1 are used for the computational study. The results obtained by solving the goal programming approach with the objective function (20) and the constraints (8) to (13) for different step sizes are listed in Table 2.

A given step results in a certain number of instances. Each instance represents a realization of $\lambda_k \forall k \in K$. There are more instances than distinct efficient solutions, because different weights may still result in the same solution. Decreasing the step size delivers very few additional efficient solutions compared to a high increase in instances and therefore calculation time. The instances and the corresponding amount of distinct and efficient solutions is illustrated in Figure 10b. Even a very large number of instances cannot deliver the complete set of efficient solutions. Obviously, some efficient solutions may be skipped due to gaps that are too large while varying the weights. Additionally, the non-supported solutions are excluded, because the underlying problem utilizes integer variables [UT94]. Furthermore, even 534 solutions are too many to include for example in a quotation for the DM. Even if a lot of solutions are skipped, the trade-off between visited pilgrims and the number of employees for the solutions at hand can be looked at. To gain some insights of the efficient frontier, the results of the goal programming approach with a step size of 1% can be analysed. Here, the maximum and minimum amount of pilgrims that have not been visited is illustrated in Figure 10a for each number of employees. The remaining objective values have not been plotted in this figure, but the benefit of an additional employee can be estimated.



(a) The minimum and maximum amount of non-visited pilgrims for each number of employees of the goal programming approach with a step size of 1%.



(b) Relation of the step size and the resulting amount of distinct efficient solutions of the goal programming approach from Table 2.

Another computational study has been conducted to show the objectives conflict. Table 4 lists the objective values and weights for each objective in 12 rows. A small (0.1) a medium (0.52) and a high (0.88) weight has been assigned to each objective with equal weights for the remaining three objectives. As expected, a high weight reduces the corresponding objective value. A weight of 52% is already enough to push the number of non-visited pilgrims to 140. Punishing the walking distance with a high weight leads to the trivial solution of not deploying any employees at all. Increasing the number of employees from 5 to 6 decreases the number of non-visited pilgrims by 1790.

4.3 Enumeration of efficient solutions

Finding all efficient solutions is not guaranteed by using the goal programming approach. One possible way described in [SC04] of calculating all efficient solutions is to calculate one identified efficient solution iteratively and restrict all feasible solutions dominated by it until no further solution can be found. Let $r = 1, \ldots, R$ be the iteration count with R as the amount of passed iterations. f_{rk} is the objective value of goal $k \in K$ in iteration r. y_{rk} is a binary variable and 1 if the current solution is superior to the solution of the iteration r in goal k. Otherwise it is zero. The objective function of this approach minimizes the sum of the individual normalized objective values and is defined as $r' = R + 1$

12

Table 4: Objective values of the goal programming approach for 12 different weights.

pilgrims		employees		slack		walk	
λ	f	λ	f	λ	f[s]	λ	f[m]
0.1	**44436**	0.3	3	0.3	5151	0.3	20114
0.52	**140**	0.16	6	0.16	14555	0.16	32710
0.88	**0**	0.04	6	0.04	14555	0.04	33756
0.3	1344	**0.1**	**6**	0.3	14555	0.3	31374
0.16	3134	**0.52**	**5**	0.16	14555	0.16	31260
0.04	115940	**0.88**	**1**	0.04	0	0.04	8349
0.3	1344	0.3	6	**0.1**	**20125**	0.3	30662
0.16	3134	0.16	5	**0.52**	**4510**	0.16	35885
0.04	3134	0.04	6	**0.88**	**0**	0.04	38781
0.3	1285	0.3	5	0.3	4510	**0.1**	**38507**
0.16	16872	0.16	5	0.16	21434	**0.52**	**24067**
0.04	178131	0.04	0	0.04	0	**0.88**	**0**

for a new iteration:

$$\text{minimize} \quad F_{r'} = \sum_{k \in K} \frac{f_{r'k}}{\kappa_k} \tag{21}$$

Note that no weights are defined for the individual goals. Those weights lead only to the order in which the solutions are found. The objective values are normalized to avoid scaling issues. The constraints (8) to (13) and two additional constraint blocks are required for this approach:

$$\epsilon \leq f_{rk} - f_{r'k} + M_k(1 - y_{mk}) \qquad \forall k \in K, r = 1, \dots, R \tag{22}$$

$$\sum_{k \in K} y_{rk} \geq 1 \qquad \forall r = 1, \dots, R \tag{23}$$

The equations (22) and (23) force a solution to be found, which has a better objective value in at least one of the objectives. ϵ is a small positive number and does not exclude any solutions, since all objective values are integer. This approach can list all efficient solutions. Due to the increasing number of constraints with each iteration, the numerical study was aborted after 52 hours of calculation. In that time, 254 distinct efficient solutions have been found while taking the longest time for the last iterations. The calculation time per iteration is illustrated in Figure 11. Only 17 equal solutions have been observed when comparing the solutions with the solutions of the goal programming approach. The goal programming approach found solutions systematically distributed over the efficient frontier, whereas the enumeration approach delivered solutions relatively close to each other. The objective function in the enumeration approach does not change, so that each further iteration will deliver an adjacent solution. Even though the calculation time is too long for a practical use this approach still gives valuable insights in the set of supported efficient solutions. Those insights are discussed in the section 4.5.

13

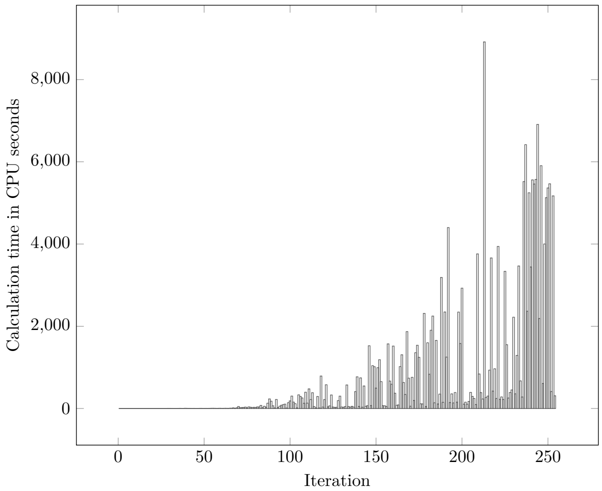


Figure 11: Calculation time in CPU seconds for each iteration of the enumeration approach.

4.4 Augmecon2

Augmecon2 is an algorithm to find the exact pareto set in MOCO problems [MF13]. This algorithm improves the Augmecon algorithm described in [Mav09]. The core idea is to optimize a single objective while bounding the remaining objective values. Iteratively adjusting the bounds of all except one objective delivers many efficient solutions. Firstly, the range of every objective value is obtained by creating the payoff table. Table 5 is the payoff table and lists the objective values for all objectives while optimizing a single objective only and improving the remaining objectives afterwards in a lexicographic order.

The best case scenario for the pilgrim objective is to visit all pilgrims. To achieve this solution at least 6 employees are necessary. Optimally for the next

Table 5: Payoff table for the Augmecon2 approach

Minimized objective k	Resulting objective function values f_{pilgrims}	$f_{\text{employees}}$	f_{wait}	f_{walk}
pilgrims	0	6	2190	40406
employees	178131	0	0	0
wait	0	7	0	41277
walk	178131	0	0	0

14

Table 6: Subset of the solutions obtained with the Augmecon2 algorithm with $\epsilon_{\text{wait}} = 2190$ and $\epsilon_{\text{walk}} = 41277$

$\epsilon_{\text{employee}}$	Pilgrims	Employees	Wait	Walk
6	0	6	2190	40406
5	1190	5	2190	40836
4	14038	4	0	38446
3	28217	3	0	33167
2	67447	2	0	21592
1	115940	1	0	8349

objective, not a single employee is in duty with the consequence of visiting not a single pilgrim. It is possible to reduce the wait objective to 0 while simultaneously visiting all pilgrims, but 7 employees are necessary. Again no employee is in duty, when optimizing the last objective, because not a single step needs to be done. The range for each objective, range_k, is the maximum minus the minimum value of each column in the payoff table.

With this payoff table every objective can be divided into equidistant steps, which serve as the bounds during the optimization. For example, the objective range of minimizing the number of employees can be divided by 7 resulting in the bounds: $0, 1, \ldots, 7$. Similarly, the objective range of wait are divided by 20 and walk by 40. Now, while optimizing a single objective, all other objectives are bound to one of the resulting values when dividing the objective range. This process is repeated for every possible combination of bounding the three objectives. Dividing each objective into more steps delivers more solutions which takes a longer computation time than solving all possible combinations when dividing the objectives in less steps.

In each iteration one of those steps ϵ_k is chosen as a bound for each objective. The following objective function minimizes the number of pilgrims and maximizes the normalized SLACK_k variables for the three remaining objectives:

$$\text{minimize} \qquad F = f_{\text{pilgrims}} - 0.1 \cdot \Big(\sum_{k \in \mathcal{K} \setminus \{\text{pilgrims}\}} 10^{-\text{index of}(k)} \cdot \frac{\text{SLACK}_k}{\text{range}_k} \Big) \quad (24)$$

Adding weights to the normalized slack variable enforces a sequential optimization of the remaining objectives.

Despite the model relevant equations (8) to (13) the following restrictions bound the remaining three objectives:

$$f_k + \text{SLACK}_k = \epsilon_k \qquad \forall k \in \mathcal{K} \setminus \{\text{pilgrims}\} \quad (25)$$

The three remaining objectives cannot be worse than the given bound ϵ_k. The objective value can be lower (better), while increasing the slack, which is normalized and maximized in the objective function (24). Solving the model for all possible combinations of the values for ϵ_k delivers the exact pareto set. Note, that some computational improvements can be achieved by iterating over the combinations as described in [MF13].

This solution approach found 215 solutions in 8 minutes. A short extract is listed in Table 6. This mainly shows the effect of lowering the number of employees on the number of non-visited pilgrims. An increasingly growing number

15

Table 7: Subset of the solutions obtained with the Augmecon2 algorithm with $\epsilon_{\text{wait}} = 2190$ and $\epsilon_{\text{employee}} = 7$

ϵ_{walk}	Pilgrims	Employees	Wait	Walk
41277	0	6	2190	40406
40245	0	7	2190	39913
39213	140	7	2190	38867
38181	1035	7	2190	37931
37149	2294	7	2190	37066
36117	4079	7	2190	35897

of pilgrims are missed when decreasing the number of employees. Employing only 5 instead of 6 employees misses 1190 pilgrims, while reducing the number of employees once again, reduces the number of missed pilgrims by 12848. The other two objectives change simultaneously, because a reduced number of employees trivially leads to a reduced total walking time. This approach point out the trade-offs between the various objectives. Note, that the trade-off depends on the step size of the ϵ_k bounds. A smaller step size than 1 (which was used in Table 6) is futile, because the number of employees is always integer.

Table 7 scrutinizes the effects of adjusting the walk objective. It is possible as shown in the first row, to employ only 6 persons and still visit all pilgrims while fulfilling the walk objective. Limiting the walk objective further, decreases the number of visited pilgrims and again this increase is growing.

Calculating those trade-off representations can help inform the DM about the connection of the underlying problem.

4.5 Gilbert et al.'s interactive approach

The two previous approaches have shown that the set of efficient solutions is too big to be included in an offer completely. The approach of Gilbert et al. includes the interaction with the DM [GHR85]. The benefit of this approach compared to other approaches is the simple explanation and fast execution concerning the cooperation with the DM. Start with any efficient solution with the objective values $f_k \forall k \in K$. During each iteration m, the DM selects the objective $\bar{k} \in K$ they want to improve next. The DM defines an $\epsilon^m_{\bar{k}} > 0$ for the objective $\bar{k}$ based on his or her utility function. $\epsilon^m_{\bar{k}}$ is the amount the objective $\bar{k}$ should be improved in iteration m if possible. Afterwards, the model with the objective function (26) subject to the constraints (8) to (13) is solved $|K| - 1$ times. Each time one of the remaining objective functions $K \setminus \bar{k}$ is minimized. An improvement in objective $\bar{k}$ is achieved by taking into account a loss in the objective function k while introducing an upper bound for all remaining objectives.

$$\text{minimize} \quad f^m_k \tag{26}$$

Subject to (8) to (13) and:

$$f^m_{\bar{k}} \leq f^{m-1}_{\bar{k}} - \epsilon^m_{\bar{k}} \tag{27}$$

$$f^m_{k'} \leq f^{m-1}_{k'} \qquad \forall k' \in K \setminus \{\bar{k}, k\} \tag{28}$$

Constraint (27) ensures an improvement in the objective $\bar{k}$ selected by the DM. A deterioration of the objective value for k is accepted while ensuring that all other

16

Table 8: Results of a sample implementation of Gilbert et al.'s interactive approach [GHR85] where solutions chosen by the DM are marked with a star *.

It.	Objectives f_k				Improvement		Deterioration
m	pilgrims	employee	slack	walk	$\epsilon_{\bar{k}}^{m}$	$\bar{k}$	k
0^*	0	6	2190s	40406s			
1^*	1815	6	0	40198s	600s	slack	pilgrims
2^*	934	7	0	40153s	500	pilgrims	employees
2	1035	6	2190s	39777s	500	pilgrims	employees
2	1285	6	0	41403s	500	pilgrims	employees
3	11344	7	0	38895s	100s	walk	pilgrims
3^*	140	7	2190s	38867s	100s	walk	slack
4	1654	7	0	38805s	600s	slack	pilgrims
4^*	140	7	0	40231s	600s	slack	walk

objective values cannot increase in constraint (28). Out of the $|K|-1$ solutions, the DM selects the feasible solution best matching their utility function, which is the starting point for the following iteration. This approach takes advantage of the utility function of the DM while their involvement additionally raises the acceptance of the offer. It also simplifies the process for the DM, because choosing from a small set of alternatives requires less effort and preparation than quantifying certain weights for various objectives that are difficult to compare.

The result of a sample implementation is given in Table 8. Each iteration contains the results of one or more solutions with their corresponding k and $\epsilon_{\bar{k}}^{m}$. The DM chose one solution of each iteration, which was used as the base for the following iteration. The solutions chosen by the DM are marked with a star *.

4.6 Comparison

Many indicators measuring the performance of MOCO problems are introduced in the literature (e.g. [OJS] or [ZKT08]). Four different performance indicators listed in [ZKT08] are calculated comparing the resulting solution set of all four solution approaches. Table 9 lists all performance indicators. The first two performance indicators are cardinality-based. Overall Non-dominated Vector Generation (ONVG) counts the number of generated solutions. Very high and very low values for the ONVG indicate a bad performance. The *Error Ratio*, indicating the ratio of the solutions which are not efficient, is not listed in Table 9 because it is zero for all solution approaches. The *Coverage of two Sets* indicator compares the solution set of two approaches. It counts how many solutions are found by both approaches relatively to the total number of solutions found by either approach. The total number of solutions is hinted in each row of the coverage indicator in Table 9. Another performance indicator calculates the *Distribution* by taking the average difference of each solution with its closest neighbor. The closest neighbor is the solution with the smallest sum of differences in all objectives. Lastly, the *Overall Pareto Spread* indicates the area of the efficient solutions which is covered by one approach. It multiplies all normalized differences between the maximum and minimum value of each objective.

Table 9: Performance indicators for the all four solution sets.

Metric	Goal	Enumeration	Augmecon2	Gilbert et al. [GHR85]
ONVG	533	254	215	9
Coverage (Goal)	100%	4.5%	7.3%	1.0%
Coverage (Enumeration)	9.4%	100%	3.9%	1.2%
Coverage (Augmecon2)	18.6%	4.7%	100%	2.3%
Coverage (Gilbert et al. [GHR85])	55.6%	33.3%	55.6%	100%
Distribution	2.5	0.5	4.4	52.6
Spread	0.9	0.00014	0.02	0.000002

Gilbert et al.'s approach [GHR85] delivers a reasonable small and therefore easy to grasp set of solutions. The *Distribution* indicates a big variety between the solutions. The spread is very small because only the area of the solutions relevant for the DM is examined. The goal programming approach spreads nearly around all solutions due to the parametrization of the weights. After 52 hours the enumeration approach was stopped and, in that time, many similar solution have been found in a small area. This means that the goal programming approach skips many supported and non-supported efficient solution even with small changes in the weights for the objectives. Augmecon2 delivers well distributed solutions with a relatively big spread while also finding a many different solutions. Even though the number of solutions found by the Augmecon2 approach is still too big to present to the DM they still can be utilized to illustrate the trade-offs between the different objectives as done in section 4.4. Those insights can help the DM while interacting with Gilbert et al.'s [GHR85] approach. Alternatively, the DM could, if he is interested, be guided through systematically structured solutions obtained by the Augmecon2 approach.

5 Managerial implications

While applying the solution approaches to the described problem broader insights emerged apart from the performance indicators that compare the computational results. Despite being practical and applicable, the obtained solutions cannot only be used for creating a quotation for the responsible decision makers. It is also usable for the operation itself. The selected solution contains the number of employees and the sequence of camps (paths) each individual employee should visit. A smartphone application stores the path for each employee to help them locate the camps. The next camp including the shortest route to it is displayed to the employee. Upon arrival, the employee can log the dispatching status of the pilgrims in the camp.

Knowing the path for each employee before the event starts has several advantages. Firstly, the employees can be trained with their own specific path, since finding the correct camps can be difficult in crowded areas. Furthermore, no internet connection and communication between the smartphone application and a central server is necessary during the operation. The network stability cannot be guaranteed in harsh climate conditions and with a myriad of connected devices. Training the employees can therefore reduce the number of mistakes done during the operation. After completing the operation, the execution can be analysed, if the smartphone tracked the positions of every employee. The planned and actual travel and waiting times can be compared and the parametrization can be tweaked for future operations. Additionally, the observations by the employees have an increased credibility, because the presence at the correct camp at the correct time is recorded. This empirical data might be important when accusing a group of pilgrims violating of the schedules.

6 Conclusion and outlook

If an automated system, which monitors the exact departure times of all pilgrims, is not available, manually recording the departure times is necessary. There are many different aspects which can be measured and evaluated when directing a small workforce between the camps. Four of those aspects have been considered in this paper. While solving a MOCO problem, many different OR methods with individual advantages and disadvantages can be applied. The goal programming approach illustrates the trade-off between the different goals even if the weights of the DM are not known. Unfortunately, too many supported efficient solutions are generated to be presented to the DM. Additionally, the non-supported efficient solutions are not obtained. The enumeration approach compensates this disadvantage by listing both the supported and non-supported efficient solutions. For this problem, the time needed to list all efficient solutions is not practical. The Augmecon2 approach delivers well spread solutions without overwhelming the DM with a myriad of solutions. It also can be used to show the trade-offs between the four objectives. Gilbert et al.'s interactive approach [GHR85] helps to systematically search through the efficient solutions even if the utility function of the DM is not known. The involvement of the DM in the decision process also raises their acceptance. Due to this consideration, Gilbert et al.'s interactive approach [GHR85] was chosen to select an efficient solution for the described problem. The presented approach does not include any fairness criterion concerning the tasks for the employees. In further research, the resulting routes for the employees could be compared in terms of length and waiting times.

References

[Bek06] Tolga Bektas. "The multiple traveling salesman problem: an overview of formulations and solution procedures". In: *Omega* 34.3 (2006), pp. 209–219. DOI: 10.1016/j.omega.2004.10.004.

[BHC95] Robert Bowerman, Brent Hall, and Paul Calamai. "A multi-objective optimization approach to urban school bus routing: Formulation and solution method". In: *Transportation Research Part A: Policy and Practice* 29.2 (1995), pp. 107–123. DOI: 10.1016/0965-8564(94)E0006-U.

[CJ98] Piotr Czyzżak and Adrezej Jaszkiewicz. “Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization”. In: *Journal of Multi-Criteria Decision Analysis* 7.1 (1998), pp. 34–47. DOI: 10.1002/(SICI)1099-1360(199801)7:1<34::AID-MCDA161>3.0.CO;2-6.

[CLM01] J-F Cordeau, G. Laporte, and A. Mercier. “A unified tabu search heuristic for vehicle routing problems with time windows”. In: *Journal of the Operational Research Society* 52.8 (2001), pp. 928–936. DOI: 10.1057/palgrave.jors.2601163.

[DL91] Martin Desrochers and Gilbert Laporte. “Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints”. In: *Operations Research Letters* 10.1 (Feb. 1991), pp. 27–36. DOI: 10.1016/0167-6377(91)90083-2.

[Dye72] James S Dyer. “Interactive goal programming”. In: *Management science* 19.1 (1972), pp. 62–70. DOI: 10.1287/mnsc.19.1.62.

[EG00] Matthias Ehrgott and Xavier Gandibleux. “A survey and annotated bibliography of multiobjective combinatorial optimization”. In: *Or Spectrum* 22.4 (2000), pp. 425–460. DOI: 10.1007/s002910000046.

[EG04] Matthias Ehrgott and Xavier Gandibleux. “Approximative solution methods for multiobjective combinatorial optimization”. In: *Top* 12.1 (2004), pp. 1–63. DOI: 10.1007/BF02578918.

[EG08] Matthias Ehrgott and Xavier Gandibleux. “Hybrid metaheuristics for multi-objective combinatorial optimization”. In: *Hybrid metaheuristics*. Springer, 2008, pp. 221–259. DOI: 10.1007/978-3-540-78295-7_8.

[FF95] Carlos M Fonseca and Peter J Fleming. “An overview of evolutionary algorithms in multiobjective optimization”. In: *Evolutionary computation* 3.1 (1995), pp. 1–16. DOI: 10.1162/evco.1995.3.1.1.

[FM14] Kostas Florios and George Mavrotas. “Generation of the exact Pareto set in multi-objective traveling salesman and set covering problems”. In: *Applied Mathematics and Computation* 237 (2014), pp. 1–19. DOI: 10.1016/j.amc.2014.03.110.

[Gen+98] Michel Gendreau et al. “A generalized insertion heuristic for the traveling salesman problem with time windows”. In: *Operations Research* 46.3 (1998), pp. 330–335. DOI: 10.1287/opre.46.3.330.

[GHR85] Kenneth C Gilbert, David D Holmes, and Richard E Rosenthal. “A multiobjective discrete optimization model for land allocation”. In: *Management Science* 31.12 (1985), pp. 1509–1522. DOI: 10.1287/mnsc.31.12.1509.

[Haa+16] Knut Haase et al. “Improving Pilgrim Safety During the Hajj: An Analytical and Operational Research Approach”. In: *Interfaces* 46.1 (2016), pp. 74–90. DOI: 10.1287/inte.2015.0833.

[Han00] Michael Pilegaard Hansen. “Use of substitute scalarizing functions to guide a local search based heuristic: The case of moTSP”. In: *Journal of Heuristics* 6.3 (2000), pp. 419–431. DOI: 10.1023/A:1009690717521.

[HJAA07] Dirk Helbing, Anders Johansson, and Habib Zein Al-Abideen. "Dynamics of crowd disasters: An empirical study". In: *Physical review E* 75.4 (2007), p. 046109. DOI: 10.1103/PhysRevE.75.046109.

[Jas02] Andrzej Jaszkiewicz. "Genetic local search for multi-objective combinatorial optimization". In: *European journal of operational research* 137.1 (2002), pp. 50–71. DOI: 10.1016/S0377-2217(01)00104-7.

[JST08] Nicolas Jozefowiez, Frédéric Semet, and El-Ghazali Talbi. "Multi-objective vehicle routing problems". In: *European journal of operational research* 189.2 (2008), pp. 293–309. DOI: 10.1016/j.ejor.2007.05.055.

[KCS06] Abdullah Konak, David W Coit, and Alice E Smith. "Multi-objective optimization using genetic algorithms: A tutorial". In: *Reliability Engineering & System Safety* 91.9 (2006), pp. 992–1007. DOI: 10.1016/j.ress.2005.11.018.

[Mav09] George Mavrotas. "Effective implementation of the ε-constraint method in Multi-Objective Mathematical Programming problems". In: *Applied Mathematics and Computation* 213.2 (July 2009), pp. 455–465. DOI: 10.1016/j.amc.2009.03.037.

[MC16] Douglas Moura Miranda and Samuel Vieira Conceição. "The vehicle routing problem with hard time windows and stochastic travel and service time". In: *Expert Systems with Applications* 64 (2016), pp. 104 –116. DOI: https://doi.org/10.1016/j.eswa.2016.07.022.

[MF13] George Mavrotas and Kostas Florios. "An improved version of the augmented ε-constraint method (AUGMECON2) for finding the exact pareto set in multi-objective integer programming problems". In: *Applied Mathematics and Computation* 219.18 (May 2013), pp. 9652–9669. DOI: 10.1016/j.amc.2013.03.002.

[OJS] T. Okabe, Yaochu Jin, and B. Sendhoff. "A critical survey of performance indices for multi-objective optimisation". In: *The 2003 Congress on Evolutionary Computation*. IEEE. DOI: 10.1109/cec.2003.1299759.

[PSS07] Luis Paquete, Tommaso Schiavinotto, and Thomas Stützle. "On local optima in multiobjective combinatorial optimization problems". In: *Annals of Operations Research* 156.1 (2007), pp. 83–97. DOI: 10.1007/s10479-007-0230-0.

[SC04] John Sylva and Alejandro Crema. "A method for finding the set of non-dominated vectors for multiple objective integer linear programs". In: *European Journal of Operational Research* 158.1 (2004), pp. 46 –55. DOI: https://doi.org/10.1016/S0377-2217(03)00255-8.

[Sol87] Marius M. Solomon. "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints". In: *Operations Research* 35.2 (Apr. 1987), pp. 254–265. DOI: 10.1287/opre.35.2.254.

[TJR98] Mehrdad Tamiz, Dylan Jones, and Carlos Romero. "Goal programming for decision making: An overview of the current state-of-the-art". In: *European Journal of operational research* 111.3 (1998), pp. 569–581. DOI: 10.1016/S0377-2217(97)00317-2.

[TTU00] Jacques Teghem, Daniel Tuyttens, and Ekunda Lukata Ulungu. "An interactive heuristic method for multi-objective combinatorial optimization". In: *Computers & Operations Research* 27.7 (2000), pp. 621–634. DOI: 10.1016/S0305-0548(99)00109-4.

[Ulu+99] Ekunda Lukata Ulungu et al. "MOSA method: a tool for solving multiobjective combinatorial optimization problems". In: *Journal of Multi-Criteria Decision Analysis* 8.4 (1999), pp. 221–236. DOI: 10.1002/(SICI)1099-1360(199907)8:4<221::AID-MCDA247>3.0.CO;2-O.

[UT94] Ekunda Lukata Ulungu and Jacques Teghem. "Multi-objective combinatorial optimization problems: A survey". In: *Journal of Multi-Criteria Decision Analysis* 3.2 (1994), pp. 83–104. DOI: 10.1002/mcda.4020030204.

[ZKT08] Eckart Zitzler, Joshua Knowles, and Lothar Thiele. "Quality Assessment of Pareto Set Approximations". In: *Multiobjective Optimization*. Springer Berlin Heidelberg, 2008, pp. 373–404. DOI: 10.1007/978-3-540-88908-3_14.

A Summaries

A.1 Reducing the combined peak energy demand for a fleet of electrical locomotives

English Summary

We consider a German railroad company with a high energy demand. Energy providers incentivize a low peak demand of their customers by pricing the peak demand. In the electricity networks for railroads in Germany the peak demand is priced with a constant factor. Coordinating the fleet of locomotives of the railroad company can decrease the peak demand and therefore their energy costs. A real-time approach is presented and an algebraic non-linear model for generating train movements, which minimizes the energy cost. The input of the model are train schedules including train properties, track topologies and velocity limits along the track. The output are train speed profiles for all trains in operation. To the best of our knowledge such a model determining the velocity of multiple trains while calculating the physical work done to minimize the electrical peak demand does not exist yet. To evaluate the mathematical framework and therefore the fundamental equations of a model a field study has been conducted to compare the measured and calculated amount of work done. A tailored heuristic and a genetic algorithm are developed to find good solutions in a reasonable time. Both heuristic approaches are based on time-based velocity limits to simplify the control over the movement of each train. This innovative approach solves the difficulty of computing feasible velocity-time graphs for each train while ensuring the arrival after a certain distance at the scheduled destination. On small instances, the non-linear model finds on average the best solutions with a computational time limit of 5 minutes. The non-linear model struggles to find a feasible solution on bigger instances where the tailored heuristic finds the best solutions. Combining the solution obtained with the tailored heuristic with the genetic algorithm improves the solution. The cost savings of coordinating all trains instead of optimizing each train individually is on average 3.8% when operating two trains, over 20% when operation five trains and around 30% for seven or more trains. These results advice to take the peak demand into account when optimizing the energy consumption.

German Summary

Wir betrachten ein deutsches Eisenbahnverkehrsunternehmen mit einer hohen elektrischen Spitzenlast. Energieversorger bepreisen die Spitzenlast damit

Kunden diese möglichst niedrig halten. In Deutschland wird die Spitzenlast mit einem konstanten Kostenfaktor bepreist. Die Koordination einer Flotte von elektrischen Lokomotiven von Eisenbahnverkehrsunternehmen kann die Spitzenlast und damit die Energiekosten senken. Wir stellen eine Echtzeitanwendung und ein nichtlineares mathematisches Optimierungsmodell vor, welche Geschwindigkeitsprofile für alle Züge generiert, um die Energiekosten zu senken. Der Input des Modells sind die Fahrpläne der Züge, die Zugeigenschaften, die Steigungen der Gleise und Geschwindigkeitsbegrenzungen der Fahrstrecke. Das Modell gibt die Geschwindigkeitsprofile aller beteiligten Züge aus. Nach unserem besten Wissen gibt es bisher kein solches Modell, welches die Geschwindigkeitsprofile der Züge erstellt, während es die verrichtete Arbeit berechnet, um die Spitzenlast zu senken. Wir haben eine Feldstudie durchgeführt, die das mathematische und physikalische Grundgerüst evaluiert indem die abgebuchte und berechnete verrichte Arbeit verglichen wird. Eine Maßgeschneiderte Heuristik und ein genetischer Algorithmus wurden entwickelt, um vertretbare Lösungen in kurzer Zeit zu generieren. Beide heuristischen Ansätze basieren auf zeitlich beschränkten Geschwindigkeitsbegrenzungen zur besseren Kontrolle der Geschwindigkeitsprofile. Dieser innovative Ansatz gewährleistet, dass zulässige Geschwindigkeitsprofile generiert werden, durch die jeder Zug nach der im Fahrplan angegebenen Distanz am Ziel ankommt. Mit kleinen Instanzen findet der Ansatz mit dem nichtlinearen Modell im Durchschnitt die besten Lösungen, wobei die Rechenzeit auf 5 Minuten beschränkt wurde. Bei größeren Instanzen findet dieser Ansatz allerdings oft keine oder nur schlechtere Lösungen als die Heuristiken. Die Lösung der Maßgeschneiderten Heuristik kann man mit dem genetischen Algorithmus noch weiter verbessert werden. Die Kosteneinsparungen der Koordination, im Gegensatz zu einer unabhängigen Optimierung der einzelnen Züge, beträgt durchschnittlich 3.8% wenn zwei Züge, über 20% wenn fünf Züge und 30% wenn sieben oder mehr Züge im Einsatz sind. Die Erkenntnisse dieses Artikels legen nahe, dass die Spitzenlast in der Kostenoptimierung berücksichtigt werden sollte.

A.2 The quadratic multiple knapsack problem assigning storage locations in a warehouse

English Summary

The assignment of Stock Keeping Units (SKUs) to storage locations is an important problem when trying to minimize the operating cost of order picking. We consider a warehouse of a beverage retailer with a given assignment of SKUs to aisles in the warehouse. A limited number of SKUs can be rearranged to new aisles with the objective of minimizing the number of visited aisles when picking a set of orders. Furthermore, the number of replenishments of SKUs is minimized too. We formulate a quadratic multiple knapsack problem to solve big instances and apply the 2-opt algorithm for improvement. Additionally, we propose a method to generate a smaller set of representative orders with similar properties like the original set of orders. This reduced set of representative orders is used by the optimal model formulation to generate a solution which can be evaluated with the original set of orders. Using a data set from the beverage retailer with 356451 orders over two years shows that rearranging the warehouse once a year can reduce the number of visited aisles by 3%. Rearranging only a subset (5%) of all SKUs in a warehouse yields already a majority (61%) of the possible improvements.

German Summary

Die Zuordnung von Artikeln zu Lagerplätzen hat einen großen Einfluss auf die Kosten der Kommissionierung. Wir betrachten ein Kommissionierlager eines Großhändlers für Getränke mit einer bestehenden Zuordnung von Artikeln und Lagerplätzen. Eine limitierte Anzahl an Artikeln darf in neue Gänge umgelagert werden, um die Anzahl der Zugriffe auf die Gänge während der Kommissionierung zu minimieren. Außerdem soll die Anzahl der Nachfüllvorgänge der Artikel minimiert werden. Wir formulieren ein quadratisches Rucksackproblem, um große Instanzen zu lösen und wenden den 2-opt Algorithmus zur Verbesserung an. Zusätzlich stellen wir eine Methode vor, einen kleinen Datensatz an Bestellungen zu erzeugen, der einen ähnliche Eigenschaften wie ein großer Datensatz hat. Der repräsentative kleine Datensatz kann dann in das mathematische Optimierungsmodell einfließen, welches die Problemstellung optimal löst. Wir zeigen anhand des Datensatzes des Getränkegroßhändlers mit 356451 Bestellungen in zwei Jahren, dass die Umsortierung des Kommissionierlagers die Anzahl der besuchten Gänge um 3% senkt. Selbst die Umsortierung von nur 5% der Artikel im Kommissionierlager sorgen hierbei bereits für einen Großteil (61%) der insgesamt möglichen Verbesserungen.

A.3 Application of a multi-objective multi traveling salesperson problem with time windows

English Summary

The pilgrimage to Mecca, which is called Hajj, is the largest annual pedestrian crowd management problem in the world. During the Hajj, the pilgrims are accommodated in camps. For safety reasons, exact times and directions are given to the pilgrims who are moving between holy sites. Despite the importance of complying with those schedules, violations can often be conjectured. Directing a small workforce between the camps to monitor the pilgrims' compliance with the schedule is an important matter, which will be dealt with in this paper. A type of multi-objective multiple traveling salesperson optimization problem with time windows is introduced to generate the tours for the employees monitoring the flow of pilgrims at the campsite. Four objectives are being pursued: As many pilgrims as possible (1) should be visited with a preferably small workforce (2), the tours of the employees should be short (3) and employees should have short waiting times between visits (4). A goal programming, an enumeration, Augmecon2 and an interactive approach are developed. The topic of supported and non-supported efficient solutions is addressed by determining all efficient solutions with the enumeration approach. The suitability of the approaches are analysed in a computational study, while using an actual data set of the Hajj season in 2015. For this application, the interactive approach has been identified as the most suitable approach to support the generation of an offer for the project.

German Summary

Der Hajj ist die Pilgerfahrt nach Mekka und das größte jährlich stattfindende Fußgänger Crowdmanagement Problem der Welt. Währen der Hajj sind die Pilger in Camps untergebracht. Aus Sicherheitsgründen bekommen die Pilger Zeitpläne mit Wegbeschreibungen zugewiesen. Leider werden, trotz der großen Bedeutung für die Sicherheit, die Zeitpläne häufig nicht eingehalten. In diesem Artikel geht es um die Koordination einiger Aufseher, die die Einhaltung der Zeitpläne überwachen. Ein multi-objective multiple traveling salesperson Optimierungsproblem mit Zeitfenstern wird vorgestellt welches Touren für die Aufseher zwischen den Pilgercamps berechnet. Vier Ziele werden dabei berücksichtig: Möglichst viele Pilger sollen überwacht werden (1) wobei aus Kostengründen möglichst wenig Aufseher eingesetzt werden sollen (2). Dabei sollen die Touren der Aufseher möglichst kurz sein und wenig Wartezeiten zwischen den Camp Besuchen beinhalten. Vier Lösungsansätze

werden implementiert: Goal programming, Enumerationsansatz, Interaktiver Ansatz und der Augmecon2 Algorithmus. Dabei wird die Eigenschaft, ob eine Lösung supported efficient oder non-supported efficient ist, aufgegriffen. Die Eignung der Lösungsansätze werden anhand eines Datensatzes von der Hajj 2015 analysiert. Dabei stellt sich heraus, dass der Interaktive Ansatz am geeignetsten für die Erstellung eines Angebots für das Vorhaben ist.

B Affidavit

I hereby declare, Justus Bonz, in lieu of an oath, that I have written the dissertation entitled *Essays of applied mathematical optimization in logistics* autonomously - and if in cooperation with other scientists as described in the attached statement according to § 6 Abs. 4 of the doctoral regulations of the Faculty of Business Administration dated July 9, 2014 – resp. and that I did not use any other aids than those I indicated herein. The parts taken literally or by sense from other works than mine are marked as such. I assure that I did not take advantage of any commercial doctoral consultation nor was my work accepted or judged insufficient in an earlier doctoral procedure at home or abroad.

Hamburg, February 16, 2021	
Place, Date	Justus Bonz