MASTERARBEIT / MASTER’S THESIS

Titel der Masterarbeit / Title of the Master’s Thesis

“Routing and scheduling of employees with different skills in multiple working locations”

verfasst von / submitted by

Alina-Gabriela Dragomir, BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2015 / Vienna 2015

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 066 915
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Betriebswirtschaft UG2002
Betreut von / Supervisor:	Univ.-Prof. Mag. Dr. Karl Franz Dörner

Routing and Scheduling of employees with different skills in multiple working locations

Alina-Gabriela Dragomir, BSc

University of Vienna

Vienna, 2015

Student ID: a0900463

Degree program: Masterstudium Betriebswirtschaft UG2002

Supervisor: Univ.-Prof. Mag. Dr. Karl Franz Doerner

Chair: Production and Operations Management with International Focus

Abstract

This work provides a solution procedure for a real-world routing and scheduling problem motivated by the difficulties in the Austrian tourism sector. Employees are flexibly scheduled to work in shifts of different skills and locations while all necessary transportation is provided. All schedules comply with detailed time constraint requirements of both employees and employers. Employers additionally define a specific person range for each shift. The routing is modeled as dial-a-ride problem comprising inconvenience constraints, time windows, and route duration constraints. The transport requests are given by the generated schedule and transportation is provided by multiple homogeneous vehicles that share a common depot. The overall objective of the problem is to create appropriate schedules for the employees that comply with an efficient transportation, while fulfilling the requirements of the employers.

The problem is solved by metaheuristics. Two types of instances are solved and compared. Restrictive instances correspond to the traditional approach with rigid person and hour requirements per shift. Flexible instances correspond to a more modern approach using gliding work hours, and using person and hour range requirements. The results of both instance types are presented and compared.

Contents

List of Algorithms

List of Figures

List of Tables

List of Abbreviations

DARP	Dial-a-ride problem
LNS	Large neighborhood search
NRC2010	Nurse Rostering Competition 2010
NRC2015	Nurse Rostering Competition 2015
OV	Objective value
PDP	Pickup and delivery problem
VND	Variable neighborhood descent
VRP	Vehicle routing problem

Mathematical notation

Table 1: Notation for variables in the objective value function f and computational results tables

Variable	Meaning
W	Number of workers
S	Number of shifts
D	Number of days (here $D = 7$)
H	Number of time units in a day (here $H = 24$)
$h^{(w)}$	Total number of working hours of worker w
$h_{max}^{(w)}$	Maximum preferred total number of working hours of worker w
$h_{min}^{(w)}$	Minimum preferred total number of working hours of worker w
$n_{break}^{(w,d)}$	Total number of breaks of worker w on day d
$n_{sc}^{(w,d)}$	Total number of skill changes of worker w on day d
$n_{lc}^{(w,d)}$	Total number of location changes of worker w on day d
$n_{workers}^{(s,h)}$	Number of scheduled workers of shift s in hour h
$n_{workers,max}^{(s)}$	Maximum number of scheduled workers of shift s per hour
$n_{workers,min}^{(s)}$	Minimum number of scheduled workers of shift s per hour
$n_{hours}^{(s)}$	Total number of scheduled hours of shift s
$n_{hours,max}^{(s)}$	Maximum number of scheduled hours of shift s
$n_{hours,min}^{(s)}$	Minimum number of scheduled hours of shift s
$P_{W,hours,max}$	Penalty for exceeding preferred working hours of workers
$P_{W,hours,min}$	Penalty for falling below preferred working hours of workers
P_{break}	Penalty for breaks
P_{sc}	Penalty for skill changes in a single day
P_{lc}	Penalty for location changes in a single day
$P_{workers,max}$	Penalty for exceeding maximum allowed workers per shift per hour
$P_{workers,min}$	Penalty for falling below minimum allowed workers per shift per hour
$P_{hours,max}$	Penalty for exceeding maximum allowed working hours per shift per hour
$P_{hours,min}$	Penalty for falling below minimum allowed working hours per shift per hour
R_d	Number of routes on day d
c_V	Fixed cost per vehicle and day

$l^{(r,d)}$	Total driving time of route r on day d in minutes
$t^{(w,d)}$	Total waiting time of worker w on day d in minutes
$t^{(w,d,r)}$.	Total waiting time of worker w on day d in route r in minutes
f	Objective value of the solution
f_S	Objective value of the schedule
f_R	Objective value of the routing
f_{UB}	Upper bound of the solution
$f_{S,UB}$	Upper bound of the schedule
$f_{R,UB}$	Upper bound of the routing

1 Introduction

Austria is a country that depends heavily on tourism. The mountain areas experience strong seasonal tourism spikes in the winter and summer months. The facilities that flourish and depend on tourism can be found in the narrow valleys of the mountains. These companies include hotels, bed and breakfasts and general accommodation, restaurants, ski lifts and gondolas, spas, swimming facilities, and local suppliers of food, sports equipment, clothes or personal hygiene. Residents are usually located in congested areas, e.g. cities, and only few live in the mountain valleys. Because of the topography of this region there is a disproportion between required and available employees. The required employees are often seasonal workers for all branches of tourism. Many of them are female with responsibility for housework and childcare and are looking only for part-time employment. The household income is often not enough to afford a second car and transportation to a possible workplace is difficult. Public transportation is available but the timetables are usually not compatible with the service schedule of the employers. Shifts often end in the evening or even late at night where public buses are no longer available. Because of these difficulties a combined scheduling and routing optimization is attempted. The service schedules should be flexible in their begin time and end time. Tasks should be handled and finished flexible throughout the day if they are by nature independent of opening hours or do not have direct customer contact and therefore doesn't require personal attendance at specific hours. In addition to the scheduling a possible routing has to be established. This routing should guarantee the transportation of the employees to their assigned workplaces, between workplaces and back home. The routing could be obtained by optimized schedules of public transport, car sharing or publicly or privately financed buses. Also, if locations are not far from each other, walking could be incorporated. A combination of all options is also possible. This work focuses on a solution with multiple uncapacitated vehicles (buses) with a single depot for all transportation needs. The goal is to provide better circumstances for less unemployment by generating better working schedules and transportation.

The remainder of this work is structured as follows: section 2 gives a literature review of scheduling and routing problems that are similar to this topic. It is divided in subsections for each field. In section 3 the problem will be described in more detail. Section 4 describes the solution method and algorithm in detail. Section 5 shows the computational results of the test instances and section 6 finishes with the conclusion.

2 Literature review

This problem can be divided into two parts: scheduling and routing. For the scheduling part there is a lot of research in the areas for personnel scheduling and personnel rostering. For the routing part there is as much literature for vehicle routing with time windows. The difficulty with this problem is the combination of the two parts and its dissimilarity to other researched problems because of the unique constraints. Even when scheduling and routing are combined in the literature the problem differs so significantly from the problem described in this work that it is difficult to find applicable literature. The existing literature is divided into sub problems that are relevant to this problem. Each sub problem is shortly described.

2.1 Personnel scheduling – general

Personnel scheduling is a complex topic. The number of personnel, shifts, tasks, skills, days off and locations alone make this very hard to solve, increasingly difficult with increasing granularity of the minimum planning interval. In addition to that, every problem seems to have its own problem specific constraints. These constraints include the different types of contracts (full-time or part-time) or do not include employee availability at all like in the paper by Kovacs et al. [11]. In some works tasks have to be fulfilled by teams whereas in other works they are fulfilled on an individual level. Most literature works with one to half a dozen shifts per day. Shifts can be distinct or overlapping, their start times and length can be fixed from the beginning or defined during the run of the program. The coverage can be seen as a hard constraint or as a soft constraint. The soft constraint is penalized to influence the solution evaluation. In most papers skills are not included or hierarchical. This is not the case where the skills are heterarchical. The skills are neither hierarchical nor are they mutually exclusive. For most papers personnel costs is the most important factor to minimize or at least included in the optimization. In our case however personnel costs are not considered. The time related constraints are rather specific: each worker can only work when he/she is available while we try to fulfill their preferred working hours per week. Other time constraints that are often found in the literature are the maximum number of consecutive days or maximum number of hours. This applies to our problem only if we include the legal framework [21]. The paper by Dahmen and Rekik [7] is much more similar to our scheduling problem then others. They include the assigned activities to the basic schedule of work and rest times. Also, the employees have skills that are not hierarchical. The problem is solved by implementing a hybrid heuristic with two sub-problems. The first sub-problem constructs the shifts by defining work start and end times. The second sub-problem assigns the activities to the shifts.

2.2 Nurse rostering

Nurse rostering is the allocation of shifts to different nurses in a hospital. Either all nurses are equally qualified, or they posses different skills. If different skills are specified they are usually hierarchical where nurses with skills of a higher hierarchy or nurses with more experience can fulfill the task of less requirements. Often nurses have different contracts. These include part-time or full time contracts as well as a combination of permanently employed nurses or temporary staff based on contracts that have to be paid in addition to the permanently employed staff. One of the main problems with nurse rostering is to determine the days on and days off for each nurse. One constraint is to allow each nurse two (consecutive) days off. Another constraint is to refrain from scheduling nurses to morning shifts right after they were assigned to a night shift.

Jaumard et al. [9] used a column generation model to solve this problem. They designed their problem for permanent staff but also introduced floating staff and overtime to avoid unfeasible solutions if not enough nurses are available to meet demand. The preferences of the nurses which days they want to work or have a free day were considered. In addition to that, they were keen to provide constant care quality by assigning inexperienced nurses to work with experienced nurses. Therefore patients should not be taken care off by two inexperienced nurses.

Nurse rostering is still a very active research field. The first Nurse Rostering Competition 2010 (NRC2010) and second competition in 2015 (NRC2015) produced a wide range of solution methods. The goal in both competitions was to produce feasible rosters by satisfying all hard constraints and to minimize the number of violated soft constraints. Each shift needed specific skills and nurses could not only request days on or off but also specific shifts on or off. The hard constraints are very minimal: all shifts must be assigned to a nurse and a nurse can only work one shift per day. Everything else is viewed as a soft constraint.

Valouxis et al. [20] implemented a two phase approach where in the first phase the day assignment problem is formulated as an integer programming problem. They considered all work/rest patterns for each nurse and used local search to improve those patterns. In the second phase nurses are assigned to the shifts. With their algorithm they won the NRC2010. The strength of their algorithm is the splitting into two less complex problems and a very fast cost evaluation for even the smallest changes.

2.3 Timetabling

Timetabling is best known for schools or universities where classes consists of a number of students that have to be assigned to a room and a teacher. Specific skills are required for each teacher. The rooms have to be large enough to offer each student a seat and must be equipped sufficiently for the requirements of the lesson. A rather unique constraint is the minimization or the prohibition of idle time for students while simultaneously restricting the number of lessons per day. Teachers have a workload limit and pre-assignments have to be considered. If a teacher was already assigned to a class for a specific skill (subject) he or she should be assigned again for this subject.

The problem has similarities to our scheduling problem like the necessary skill for each class, employees (teachers) that offer a range of skills or minimization of idle time (breaks). Furthermore timetabling generates timetables for two parties (teachers and students). The same principle is necessary for this problem where timetables for the employees and shifts are needed. In spite of the similarities the solution methods or problem instances for timetabling cannot be that easily modified to fit this problem.

In 2011 an international timetabling competition was launched [16]. Very few participants submitted results compared to the NRC2010. The best results were generated by a Brazilian team that developed the GOAL Solver. This solver used a hybrid metaheuristic based on simulated annealing and iterated local search. The research field timetabling is still active and the instances from the competition provide a solid basis to test individual research.

2.4 Vehicle routing with time windows

The vehicle routing problem tackles the problem of servicing a number of customers with a certain amount of vehicles. Each customer is only visited once. The main goal is to minimize the route length or time (if service time is considered). Sometimes customers have certain time windows during which the service has to start.

A good example of a vehicle routing with time windows is the paper by E. Cheng and J.L. Rich [3]. It is in its core a nurse rostering problem, it is however modeled as a vehicle routing problem with time windows. Full-time nurses have to be assigned to patients. They can work overtime, which is paid extra, or part-time nurses can be hired that are paid by the hour. Each nurse is handled like a separate vehicle that has to start and end at her home (depot). Instead of generating rosters and tours for each nurse, the problem was solved as a vehicle routing problem with time windows, including compatibility, and as many depots as nurses. The time windows apply to each patient as well as to each nurse. Each nurse has to finish her tour within her time window. This problem was implemented as a two-phase algorithm. In phase one the tours were built using a randomized greedy algorithm while checking for feasibility (time windows) and while limiting the waiting time before a patient. In the second phase the schedule was improved by destroying the routes for all nurses that work overtime or part-time and assigning their patients to someone else. After that, additional improvement methods were used (e.g. patient swap).

For our case the pickup and delivery problems (with time windows) and the dial-a-ride problem (DARP) have to be examined. The difference between the two is that the DARP usually looks at transporting people and takes their inconvenience into account like long driving times or waiting times [13]. Both the pickup and delivery problem PDP as well as the DARP are sub-classes of the vehicle routing problem with pickup and delivery for the transportation between customers [13]. The PDP deals mostly only with the transportation of goods. Ropke and Pisinger[17] use an adaptive large neighborhood search for their pickup and delivery problem with time windows. The differences to this problem are the following: the amount of vehicles are not limited in this problem, the number should only be minimized because they generate fix cost, and the vehicles used in this problem are not capacitated. In addition to that this problem has various convenience constraints since it deals with the transportation of people and not goods.

The tabu search heuristic for the multi-vehicle dial-a-ride problem by Cordeau and Laporte [5] is very similar to this problem. In their problem they have transportation requests with time windows and they use multiple vehicles with a common depot. In comparison to this problem they have vehicle capacity constraints. The constraints concerning route duration and maximum ride time of a person are part of both problems.

2.5 Dial-A-Ride problem

Since the problem of this work deals with the transportation of persons and their inconvenience is directly taken into account, it can be classified as dial-a-ride problem. Attanasio et al. [1] and Cordeau and Laporte [6, 4] make the distinction between dynamic and static modes where all requests are either unknown in advance and have to be assigned to vehicles already in motion, or are known in advance and can be planned accordingly. The pickup and delivery requests have an origin and destination, whereas all locations can be both due to return trips. Most literature for the static multi-vehicle DARP try to minimize the route length, driving time, number of vehicles or user inconvenience (e.g. waiting time) while using heuristics to find a good solution. Cordeau and Laporte [5] use a tabu search heuristic for the static multi-vehicle DARP. Their problem is almost identical to this, except that their time windows are not compulsory, pickup and delivery time windows are less rigid, and they consider vehicle capacity.

2.6 Routing and scheduling combined

The service technician problems have some similar characteristics to our problem. First of all it is a combination of routing and scheduling. The technicians usually have different skills. These skills are mostly hierarchical in contrast to the heterarchical skills in this problem. The scope is to find the best/shortest routes while satisfying all customers within their specific time window. Technicians can work alone or in teams of 2 or more while specific skills or skill level requirements have to be fulfilled. Often assignments with lesser skill requirement can be fulfilled by overqualified technicians. In some works a learning curve is included where, after a certain amount of time working together with a better qualified technician, the lesser qualified technician can improve its skill level. All of these constraints can be viewed as hard constraints or as soft constraints and can be implemented accordingly. Noncompliance with the hard constraints would result in infeasible solutions whereas noncompliance with the soft constraints reduces the objective value and therefore the quality of the solution. Soft constraints are often implemented as extra cost for outsourcing, costs when a task cannot be fulfilled at all, or penalties, when a task cannot be fulfilled by the necessary skill [11].

The home health care problem [10, 3] is another example that combines routing and scheduling of employees. The employees are often divided into more than one contract group (full-time or part-time) that have to fulfill a task with a certain length within a predefined time window. It differs from the traditional nurse scheduling problem since it includes the routing to the different locations. New patients can be included in the planning, put on a waiting list or rejected altogether if no capacity is available.

Bredström et al. [2] use temporal precedence and synchronization constraints in their problem. They have implemented time windows where vehicles have to be simultaneously at the same node. They also have the problem where a package is dropped-off and picked-up by different vehicles at a transshipment point. Their objectives are the minimization of travel distance and travel time while the workload should be distributed as equally as possible between the vehicles. In contrast to our problem travel time is of low priority since the service duration occupies most of the tour time.

The main difference between all these combined routing and scheduling approaches and our problem is that the scheduling is just a part of routing with time windows. No creation of explicit rosters and timetables is necessary.

3 Problem description

The aim of this work is to match supply and demand of employment by generating flexible individual working schedules[1] while meeting all requirements posed by employers and employees and to provide transportation that minimizes the transport time, the waiting time between the arrival of the vehicle and the time of work, and the number of vehicles since each vehicle generates fixed costs. Schedules for both employees and employers have to be created.[2] Employers are situated in different locations and are in need of employees of different skills. The employers are various companies operating in the tourism sector (hotels, ski lifts, restaurants, etc.). The employers and potential employees have been interviewed to evaluate their needs and requirements. Each shift is clearly defined by the day, working location, and skill needed. The needed workforce for each shift is given by total working hours and a specific time span in which these working hours have to be provided. Each hour has an additional constraint of a minimum and maximum person limit. A more detailed representation of the constraints and requirements of the employers can be found in table 8 on page i in the appendix.

For example a hotel needs at least one person from 6 a.m. to 11 p.m. at the reception while requiring between 20 and 25 working hours during this time frame at the reception. The same hotel might need 12 to 15 working hours in total between 11 a.m. and 2 p.m. for cleaning. The person limits per hour are between 0 and 15 since it does not matter if 15 persons clean the rooms for a single hour or the persons are equally divided within the time limit. Another typical example for a company in the tourism sector is a ski rental. The ski rental might need personnel for waxing of skis. It establishes a time span from 6 a.m. to 4 p.m. while needing 18 to 20

[1]The working schedules created are for all hours of all days. They indicate for each hour if a worker is at home, travelling, on break, or working. If the worker is working he/she is clearly assigned to a specific shift.

[2]Employers give their demand in shifts. Therefore a schedule for each shift is created that indicates exactly which workers are working in each hour.

working hours in total. The ski rental might have a personnel maximum of 3 people since it can only provide 3 work benches and at most 3 people can work at the same time.

The available workforce is given by an individual specification of the hours an employee is available and the respective skills he or she possesses. The skills are independent from each other. A worker needs at least one skill to be scheduled for a shift but can possess up to all available skills. Furthermore employees specify a minimum and maximum amount of hours per week they wish to work. Each employee has a home and transportation from home to the workplace, between workplaces and from workplace to home has to be provided. A more detailed representation of the constraints and requirements of the employees can be found in table 7 on page i in the appendix.

For example employee Geller possesses the skills “reception”, “office” and “sales”. He/she is available on Monday, Tuesday, Thursday and Friday from 7 a.m. to 2 p.m. and again from 5 p.m. to 10 p.m. On Wednesday he/she is available only from 6 a.m. to 2 p.m. Geller is not available on Saturday or Sunday and wants to work between 15 to 20 hours a week. Employee Green possesses the skills “lift operation”, “ski school” and “maintenance”. He/she is available every day of the week from 6 a.m. to 4 p.m.. Green wants to work between 25 to 30 hours per week. Employee Buffay is available from Friday to Sunday from 6 a.m. to 10 p.m. On the other day he/she is not available. Buffay possesses the skills “kitchen”, “cleaning” and “service” and would like to work between 35 and 40 hours per week.

This representation of the data allows the determination of the specific needs of each employer. It also allows the creation of flexible rosters since it is not of relevance at what time a workforce is scheduled or if the same workforce is scheduled on different days of the week. The employees can switch between the different employer locations and between shifts that require different skills, as long as they possess the skills necessary. Since location changes or skill changes pose an inconvenience for the employees, these changes are avoided by using penalties. The scope is to provide good schedules that satisfy all constraints set by employers and employees. The second part of the scope is to provide transportation that minimizes the transport time and the waiting time between the arrival of the vehicle and the end/beginning of the work. Furthermore the number of vehicles and therefore routes should be minimized since each vehicle generates fixed costs. The time windows for transportation is always one hour long. This is partly because of the hourly time grid of the schedules and partly because inconvenient long driving times for workers are to be avoided. As a result workers are not allowed to work in locations that are more than 60 minutes driving time away.

The problem has the following hard constraints:

- Workers can only be scheduled for work, break rooms or travelling if they are generally available.
- A worker cannot be in two places at once.
- A worker can only work in a shift if he/she possesses the necessary skill.
- Workers are not allowed to work more than 12 hours per day (legal constraint).
- Workers are not allowed to work at locations that are more than one hour driving time away[3].
- All transportation has to be provided.[4]
- Unnecessary transportation is forbidden.[5]
- The maximum number of persons in each hour for each shift is not to be exceeded.[6]
- The maximum worked hours for each shift is not to be exceeded.[7]
- The routes have a maximum driving length and a maximum total length of 480 minutes, which corresponds to 8 hours.[8]

[3]The distance limit of one hour is to limit the inconvenient driving time for the worker as well as to comply with the hourly time grid of the solution.

[4]This includes transportation between home and workplace and between different workplaces.

[5]For example if a worker is scheduled for two different shifts in the same location, no transportation between those shifts is allowed.

[6]The employers give their maximum person and hour limit for a reason. They might not have the necessary space to accommodate an additional worker or/and they might not be willing to pay for additional workforce that is not required. Therefore this constraint is viewed as hard.

[7]See footnote 6.

[8]The total length of the route is calculated by the difference between arrival time of vehicle at depot at the end of the tour and departure time from the depot at the beginning of the tour. Therefore the total length includes all idle times between active transportation requests.

The objective value of the solution is calculated by penalizing certain values and is depicted below. The objective value has to be minimized.

Total objective value: the sum of the scheduling and routing objective values:

- the difference between preferred weekly working hours of a worker and the actual weekly working hours,
- the amount of breaks, skill changes per day and location changes per day[9],
- the amount of hours for each shift where the minimum/maximum persons are not fulfilled/exceeded[10],
- the difference between the daily required person hours of a shift and the actual working person hours,
- the total routing costs (fix cost + driving length + waiting times for the workers).

The list above can be summarized in a formula for the objective value: The total objective value (OV) is denoted as f and is given by

$$f = f_S + f_R, \tag{1}$$

where f_S is the OV of the schedule and f_R is the OV of the routing. The scheduling OV f_S consists of the first four terms in the list above. The notation is explained in table 1 on page viii.

[9] A break counts as a skill change.

[10] If the maximum persons for a shift are exceeded the schedule is not feasible. This should never occur. However it is still part of the objective value calculation.

$$\begin{aligned}
f_S \quad = \quad & \sum_{w=1}^{W} \Big\{ P_{W,hours,max} \cdot \max\left(h^{(w)} - h^{(w)}_{max},\, 0\right) \\
& + P_{W,hours,min} \cdot \max\left(h^{(w)}_{min} - h^{(w)},\, 0\right) \Big\} \\
+ \quad & \sum_{w=1}^{W} \sum_{d=1}^{D} \left\{ n^{(w,d)}_{break} P_{break} + n^{(w,d)}_{sc} P_{sc} + n^{(w,d)}_{lc} P_{lc} \right\} \\
+ \quad & \sum_{s=1}^{S} \sum_{h=1}^{H} \Big\{ P_{workers,max} \cdot \max\left(n^{(s,h)}_{workers} - n^{(s)}_{workers,max},\, 0\right) \\
& + P_{workers,min} \cdot \max\left(n^{(s)}_{workers,min} - n^{(s,h)}_{workers},\, 0\right) \Big\} \\
+ \quad & \sum_{s=1}^{S} \Big\{ P_{hours,max} \cdot \max\left(n^{(s)}_{hours} - n^{(s)}_{hours,max},\, 0\right) \\
& + P_{hours,min} \cdot \max\left(n^{(s)}_{hours,min} - n^{(s)}_{hours},\, 0\right) \Big\},
\end{aligned} \tag{2}$$

where the max function is given by

$$\max(a, b) = \begin{cases} a & a > b, \\ b & b \geq a. \end{cases} \tag{3}$$

The OV for the routing f_R is given by

$$f_R = \sum_{d=1}^{D} R_d c_v + \sum_{d=1}^{D} \sum_{r=1}^{R_d} l^{(r,d)} + \sum_{d=1}^{D} \sum_{w=1}^{W} t^{(w,d)}. \tag{4}$$

It is also necessary to use partial objective values. The partial OV $f^{(w,d)}$ for a worker w on day d is defined by

$$\begin{aligned}
f^{(w,d)} \quad = \quad & \frac{1}{D} \left(P_{W,hours,max} \cdot \max\left(h^{(w)} - h^{(w)}_{max},\, 0\right) + P_{W,hours,min} \cdot \max\left(h^{(w)}_{min} - h^{(w)},\, 0\right) \right) \\
+ \quad & \left(n^{(w,d)}_{break} P_{break} + n^{(w,d)}_{sc} P_{sc} + n^{(w,d)}_{lc} P_{lc} \right).
\end{aligned} \tag{5}$$

The partial OV of a shift s is defined by

$$
\begin{aligned}
f^{(s)} \quad = \quad & \sum_{h=1}^{H} \Big\{ P_{workers,max} \cdot \max\left(n_{workers}^{(s,h)} - n_{workers,max}^{(s)},\, 0\right) \\
& + P_{workers,min} \cdot \max\left(n_{workers,min}^{(s)} - n_{workers}^{(s,h)},\, 0\right) \Big\} \\
+ \quad & \Big\{ P_{hours,max} \cdot \max\left(n_{hours}^{(s)} - n_{hours,max}^{(s)},\, 0\right) \\
& + P_{hours,min} \cdot \max\left(n_{hours,min}^{(s)} - n_{hours}^{(s)},\, 0\right) \Big\}. \qquad (6)
\end{aligned}
$$

Using the definitions for $f^{(w,d)}$ and $f^{(s)}$ the formula for the scheduling OV f_S can be drastically simplified to

$$
f_S = \sum_{w=1}^{W} \sum_{d=1}^{D} f^{(w,d)} + \sum_{s=1}^{S} f^{(s)}. \qquad (7)
$$

4 Solution method

For the personnel scheduling in general many different solution techniques are used. Most of the literature use either mathematical programming methods, constructive heuristics or improvement heuristics. Few use simulation techniques. Van den Bergh et al. [21] give a good overview over the solution methods for personnel scheduling used by all relevant papers in this area.

4.1 The general structure of the algorithm

For this problem the solution method used is a combination of different metaheuristics. First, four different construction operators are used to generate diverse starting solutions. By working on more than one solution for each problem instance more diversification and a potentially better end solution is possible than with only a single starting solution. The initial schedules are created in fours, one of each construction operator. Therefore the amount of initial solutions is always a multiple of four. The schedules are divided into one hour time grids. All shifts consist of blocks that are one hour long and shift begin and shift end can only occur at the full hour. All solutions contain a scheduling part and a routing part. The initial schedules are improved by a large neighborhood search (LNS) [14] with three destroy and two repair operators. All operators are tailored to the problem at hand. The goal is to optimize the whole solution which consists of a routing and scheduling part that affect each other. The routing of the employees is embedded in the overall optimization process. It is calculated for each schedule and is modeled as a dial-a-ride problem with time windows. The metaheuristic variable neighborhood descent (VND) [12], with three neighborhoods (move, swap and 2-opt*) embedded in a threshold acceptance framework [15], is used for its improvement. Each improvement cycle consists of multiple iterations of the LNS and VND. At the end the solution with the best objective value is selected. A solution is feasible if every worker only works when he/she is available, is never scheduled to work in two places at once, is only scheduled for shifts with a skill he/she possesses and the maximum person and working hour limit for each shift is not exceeded. The best found solution is then printed to three separate output files, one file for each the worker schedules, the demand schedules, and the tours.

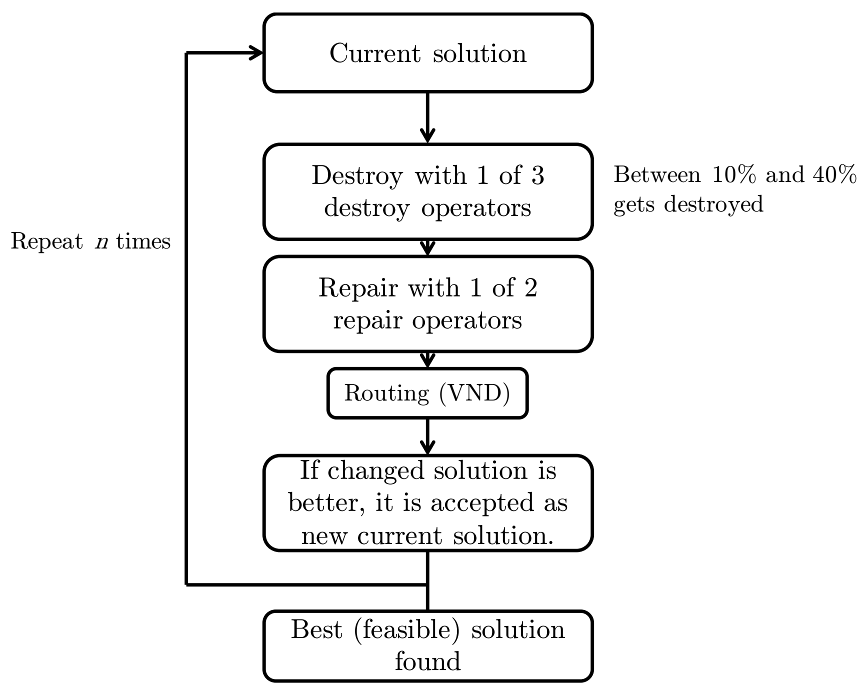


Figure 1: Simplified solution algorithm

4.2 Schedule construction

All schedule construction operators have a systematic approach with at least one randomized part. Method 1, *skill rareness,* satisfies the demand for scarce skills first. The rareness is determined as the proportion between required and available hours for each skill per week. Therefore if workers possess multiple skills, it is more likely that they are scheduled to work with the skill that is more scarce than with skills that are found in abundance. With this method a better division of labor is attempted. Method 2, *random location*, satisfies first the demand of all skills of a random working location. This ensures that the locations are not filled in the same sequence every time. Also, if availability cannot meet demand[11], whole locations are satisfied first whereas others remain empty. This allows a possible elimination of whole locations when improving the problem instances. Method 3, *random shift*, randomly fills a shift. The list of shifts is randomly sorted. The shifts at the top of the list are filled first. Those shifts have therefore a higher probability to have their constraints fulfilled (minimum persons and minimum hours). The minimum demand of the shifts filled with a lesser priority might not be fulfilled completely because the workers are less available since they have already been scheduled for other shifts. Method 4, *location distance*, is similar to *random location* where the demand of a complete location is satisfied first. The difference lies in a modified worker selection function which is explained below. Algorithm 1 on the next page shows how the schedule construction operators work using pseudo code. The overall difference between the operators lies in the prioritization of the order of shifts to fill and in the worker selection function.

One part of each schedule generation method is the function *findBestNextWorker* that is in described in algorithm 1. The function selects the next worker for a certain hour. It selects workers first systematically and later randomly. The worker selection function gives structure as well as randomness to the schedule construction methods. In general following conditions have to be fulfilled to schedule a worker: the worker is generally available, he/she possesses the necessary skill, a worker is not already otherwise engaged and he/she is not traveling. Furthermore workers are only allowed to work at locations that are not more than 60 minutes driving time away from their home location. This constraint ensures a maximum driving

[11] This is the case if there is not enough workforce to fill all shifts.

```
Pick the next shift to be filled (depending on the construction method)
    maxTries = min needed hours
    tries = 0
    while fulfilled hours < min needed hours
        for hour = 0 to 23
            if shift requires hour AND max persons not reached
                findBestNextWorker
                if no worker found
                    go to next hour
                fill schedule
                if min persons fulfilled == FALSE
                    fill hour again
        End for
    tries = tries + 1
    if tries >= maxTries
        break while
    End while
```

Algorithm 1 : Schedule construction

time, decreases inconvenience for the worker, and complies with the hourly time grid.

The worker selection function undergoes following steps: workers are preferred if they are already scheduled at the location currently to be filled and have already practiced the skill that is needed on the current day. Therefore it is ensured that little location changes or skill changes occur, which are both penalized. The first worker found, that fulfills all conditions, is selected. If no worker can be found, the workers are preferred that were scheduled before at the current location. Therefore at least a location change is avoided. If still no worker can be found, methods 1, 2 and 3 select a random worker that is available and possesses the necessary skill. For *location distance* the nearest worker is selected. The nearest worker is found by sorting all workers by the distance between their home and this particular working location. For all steps all hard constraints have to be satisfied.

In general a worker can be at home, working, traveling or on break. Breaks should be avoided since they are inconvenient for the workers. Legal breaks are not planned but workers are not allowed to work more hours than the legal daily limit.[12]

[12] For this problem the legal daily working hour limit is set to 12 hours. It can however be changed to 8, 10 or 13 hours respectively, depending on the legal assumptions made.

The partial objective value for the worker schedules is influenced by the number of skill changes, breaks, location changes, and all hours that exceed or fall below the working hour limits for supply or demand, each weighted by penalties of different heights. The exact calculation of the partial objective value for a worker schedule for a day can be found in equation 5 on page 12.

4.3 Schedule improvement

The initial schedules are improved by a large neighborhood search (LNS) with three destroy and two repair operators. The LNS was proposed by Shaw [18] for a vehicle routing problem (VRP). Each iteration uses one of three destroy and one of two repair operators and then compares the new with the old solution objective value. LNS is a metaheuristic that can be applied to various problems, in this case to personnel scheduling using problem specific operators. The solution is destroyed until a limit between 10% and 40% is reached.[13] The repair operator mends the solution and refills the schedules. After each schedule improvement the routing is updated to allow a comparison of two whole solutions. Only improvements are accepted. The selection of the destroy and repair operators is random. A schematic representation of the schedule improvement with LNS can be seen in figure 2 on the next page. The LNS is terminated when the maximum number of iterations is reached.

Algorithm 2 on page 19 shows the schedule improvement in detail. The schedule is improved for a fixed number of iterations. For each iteration the destruction percentage, the destroy operator and the repair operator are chosen randomly. The functions *destroySchedule* and *repairSchedule* apply the selected operators on the schedule. The separate operators are described in the subsection 4.3.1 on the next page. After the operators have been applied, additional functions for the improvement of the schedule are used. The functions *fulfillWorkersMinHours*, *findMultipleTravellingTimes* and *findBreaks* are explained in more detail in subsection 4.4 on page 22. After the schedule improvement is completed the routing is calculated to compare the new complete solution[14] with the best found solution so far. If the objective value of the modified (new) solution is smaller than the objective value of

[13] A destroy limit between 10% and 40% is randomly selected for each iteration.
[14] A solution consists of a routing and scheduling part.

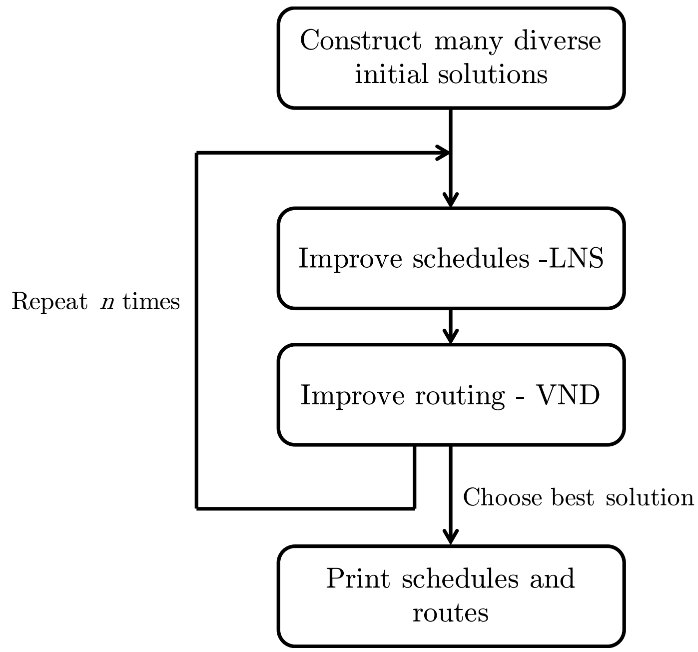


Figure 2: Schedule improvement with LNS

the best found solution so far, it is accepted as the new best found solution. If not, the modified solution is dismissed.

4.3.1 Operators

The *random destroy* operator picks a random worker and a random day. In a third of the cases it destroys the complete solution for this day. In a third of the cases it destroys the solution on this day until a random hour. In the other cases it destroys the solution on this day after a random hour. To destroy means that the worker is deleted from all shifts he had been working in and reset to home. The operator is repeatedly applied until the destroy limit is reached.

The *worker penalty destroy* operator looks at worker schedules with a high partial objective value $f^{(w,d)}$ (see equation (5) on page 12) and therefore schedules where high penalties occur. This is the case if a worker schedule contains many skill changes, location changes, breaks or the required minimum working hours per week[15] are not

[15]Schedules with less or more than 7 days per week are possible, for example 5 days if the scheduling and routing is only necessary for the days from Monday to Friday or even 2 days for the weekend.

```
for it = 0 to maxNumberOfIterations
    randomly choose destruction percentage, destroy operator, repair operator
    destroySchedule
    repairSchedule
    for w = 0 to numberOfWorkers
        fulfillWorkerMinHours
        findMultipleTravellingTimes
        findBreaks
    End for
    Do routing
    If OV of new solution < OV of best found solution
        accept new solution as best found solution
    Else
        dismiss new solution
    it = it + 1
End for
```

Algorithm 2 : Detailed Schedule Improvement

met.[16] All worker schedules are sorted by partial objective value in descending order. In two thirds of the cases the worker schedule with the (next) worst objective value is removed (emptied) until the destroy limit[17] is reached. In one third of the cases a random worker is removed. Since some workers might possess a single skill for which there is no demand, they will never work and will always have a very bad partial objective value. Therefore the random removal is introduced to avoid repetition of always removing the same workers. The complete daily schedule of the selected workers is destroyed. This process is repeated until the destroy limit is reached.

The *shift penalty destroy* looks at shifts where high partial objective values $f^{(s)}$ (see equation (6) on page 13) occur. Figure 3 depicts the general procedure. The partial objective value is influenced by not fulfilling the required person minimum for each hour, over fulfilling the required person maximum for each hour, or under/over fulfillment of the necessary working hours for the whole shift.[18] Each shift has mul-

[16] Since this is a daily evaluation, the penalty for the difference of weekly actually worked hours to the required minimum of weekly worked hours is divided by the number of days of the week.

[17] The destroy limit percentage is calculated from the total amount of worker schedules. That is number of workers times number of days.

[18] Exceeding the person maximum or the maximum working hours for the shift violates a hard constraints and would actually lead to an infeasible solution and is not possible, neither at schedule construction nor during the improvement. However it is, just in case, still part of the partial objective value.

tiple workers scheduled. Each iteration there is a 50 percent chance each to pick the shift with the worst partial objective value or a random shift. This is done to counteract highly unequal demand and supply. If a shift requires a high minimum amount of persons with a skill that very few possesses, the same shift will always be selected to be removed. The workers of the shift with the worst objective value get removed completely from it. Furthermore workers with the same skill that is needed for this particular shift are removed from their shifts. Therefore complete skills are freed and can be rescheduled. Only if the destroy limit is not reached after the first iteration, the next worst shift and skill is selected and also removed.

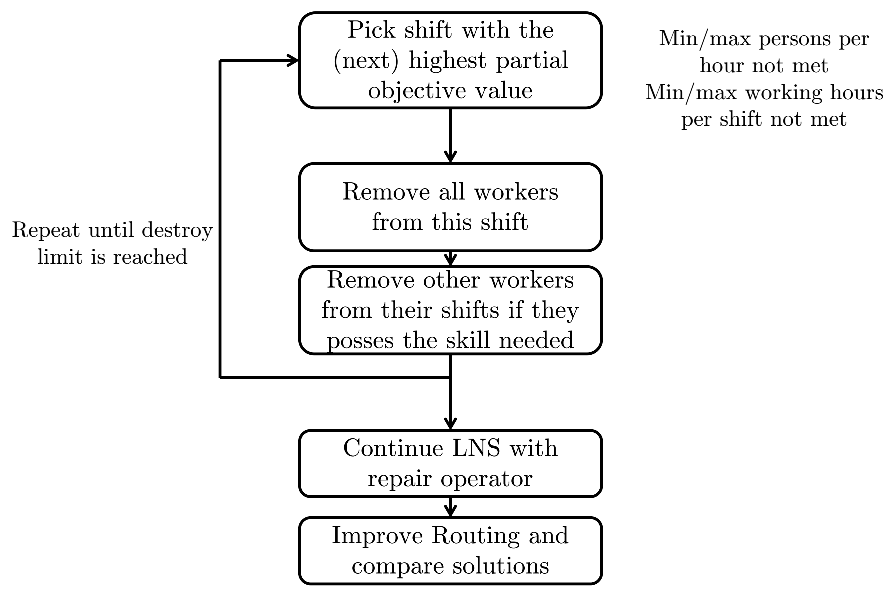


Figure 3: shift penalty destroy operator

The *initial repair* operator first sorts the shifts by their respective objective value and fills them with the same principle as the *findBestNextWorker* function in the schedule construction. The exact calculation of the partial objective value for a shift can be found in equation (6) on page 13. The objective value is worse if the required hours or minimum person limit are not fulfilled. Just like in the schedule construction workers are selected first if they already work in the location with the skill necessary. If no worker is found a worker is selected that at least already works in the location. If again no worker is found, a new worker is selected. For the third option there is a 50 percent chance to select a random worker and a 50 percent chance to select the next nearest worker (home – workplace distance) for this location. Therefore it is a combination of the four construction methods.

The *coverage repair* operator also sorts the shifts by their respective objective value (see equation (6) on page 13). The algorithm goes through the sorted shifts and fills one after the other, the worst one first. For a shift the algorithm picks the worker that can cover the most hours continuously even if that worker has to undergo an additional location and/or skill change. The *initial repair* operator considers each hour separately whereas the *coverage repair* operator considers the whole day. An algorithm for the worker selection for the *coverage repair* operator is depicted in algorithm 3. A worker is available under the following conditions: On this day and in this hour he/she has general availability, the worker is currently at home (not travelling, working or on break at a working location), the worker possesses the skill needed for this shift, the location of the shift is not prohibited by being more than 60 minutes driving time from the workers home location or already visited working location on this day, the worker has not worked already more than the legal limit on this day, and the worker has not worked already more than his/her specified maximum number of required weekly working hours.

$i_w = 0$
for $w = 0$ to numberOfWorkers
    for hour = 0 to 23
        if worker w is available **AND** shift requires hour
            $i_w = i_w + 1$
    End for
End for
Return worker w with highest coverage value

i is the number of hours each worker w can cover on this day
The exact meaning of worker availability is explained in the text

Algorithm 3 : Worker selection of coverage repair operator

After each destroy/repair iteration the routing is redone. The total objective value determines the quality of the schedules including the routing. If the total objective value is better after the permutation, the new schedule is kept. If the total objective value is worse, the new schedule is dismissed.

4.4 Additional functions for schedule improvement

After all repair operators have been applied, additional functions are used to further improve the schedules. Even though the repair operators try to fill the schedules until the maximum needed hours and persons for each shift is fulfilled there is still the possibility that workers have been overlooked if they are selected randomly.

4.4.1 Fulfill minimum required weekly working hours of employees

The function *fulfillWorkersMinHours* tries to additionally schedule workers to work if they have not reached their maximum required working hours per week. Four steps are undergone: First the algorithm looks if a worker is already scheduled to work at all. If this is the case the functions *fillBreaks, addHoursAtShiftBegin*, *addHoursAtShiftEnd* are used. The function *addHoursAnywhere* is used after that or as the only function if the worker schedule is completely empty.

The *fillBreaks* function is used if a worker is already scheduled to work anywhere at all in the week and all his/her breaks are tried to be filled since they pose an inconvenience for the worker. Algorithm 4 shows the basic functionality of the *fillBreaks* function. First it loops through all the workers, days, and hours, and finds all the workers that are scheduled for a break. Then it tries to assign the worker to a shift where he/she actually works. The preferred shift is the one where the worker was scheduled the hour before since it avoids an additional skill change. If a move to this shift is possible is determined by the following: the shift requires the current hour to be filled, the maximum number of persons have not been reached for this hour, and the maximum hours have not been reached for the overall shift. If one of those conditions is not fulfilled the algorithm tries to assign the worker to any other shift that has in addition to the previous mentioned conditions the following: the shift has to be part of the current day and location, and has to require a skill that the worker possesses.

The function *addHoursAtShiftBegin* and *addHoursAtShiftEnd* try to add additional working hours at the begin or end of a series of working hours (or shift in the eyes of the worker) if the minimum required working hours of the worker have still not been fulfilled. This is the second and third sub function of the function

```
loop trough all workers, days and hours
    if worker is on break
        if possible
            assign worker to shift from previous hour
        else
            assign worker to any shift possible
End loop
```

The possibility of an assignment is explained in the text.

Algorithm 4 : Fill breaks

fulfillWorkersMinHours. For this, the schedule is searched and every time a worker begins/ends work it tries to add an additional hour before or after. Preferred are, of course, shifts where the worker already works the next/previous hour. If that is not possible any other shift is tried out. For this, the same conditions apply as for the *fillBreak*s. However there are additional conditions that have to be fulfilled: for the worker to actually work he/she not only has to be available for this additional hour but also for the hour before/after because a time slot has to be reserved for transportation. This is not necessary for the *fillBreaks* because the worker is already at a working location and these conditions have already been positively evaluated.

Steps one to three are only feasible if the worker is already scheduled to work anywhere at all in the week. The fourth step *addHoursAnywhere* is used for both empty and already filled schedules. It again loops trough all days and hours and tries to schedule the worker to any shift where all constraints are fulfilled. The complete function *fulfillWorkersMinHours* is illustrated in algorithm 5. After every sub function the algorithm checks if the maximum required working hours have been fulfilled and terminates accordingly.

```
for each loop worker where maxRequiredWorkingHours is not fulfilled
    if worked hours > 0
        fillBreaks()
        addHoursAtShiftBegin()
        addHoursAtShiftEnd()
    End if
    addHoursAnywhere()
End for each loop
```

Algorithm 5 : Fulfill worker min hours

4.4.2 Find multiple travelling times

The schedule is implemented as an one hour time grid. Depending on how the schedule is filled, multiple travelling times can occur. Such cases are forbidden since the pickup and delivery time window is limited to one hour. Therefore the function *findMultipleTravellingTimes* is used to eliminate consecutive travelling slots or travelling slots that are not necessary. Unnecessary travelling slots are for example if travelling is set between two hours where worker is at home or at the same location. First, all unnecessary travelling slots between two hours at home are removed. The worker is set to home. Second, all unnecessary travelling slots between two hours at the same location are removed. The algorithm undergoes following steps: 1. Try to assign the worker to the shift from the previous hour, 2. Try to assign the worker to the shift from the next hour, 3. Try to assign the worker to any shift in this location, and 4. Assign worker to break room. All steps check for the same conditions as in *fillBreaks*.

Now it is ensured that travelling only occurs between two different working locations and in-between home and a working location. If multiple travelling slots occur between the home of the worker and a working location, the travelling is deleted and worker is set to home. If multiple travelling occurs in-between two working locations, the first travelling is deleted and assigned to the previous shift if possible. If not possible, the worker is assigned to the break room.

4.4.3 Find breaks

Breaks occur mostly because a worker was set to travelling at a point where travelling was not necessary or more than one time slot was reserved for travelling. To further eliminate unnecessary breaks, the *findBreaks* function is used. This function tries to assign the worker to an active shift, preferably the one he worked in before the break, or any other shift in this location on this day that requires workers at this hour, that requires a skill the worker possesses, and has not jet reached its limit of maximum persons per hour or maximum hours per shift. If no shift can be found the worker remains in the break room.

4.5 Routing

The second part of the problem is the routing. A route contains nodes which have a type indicating whether it is a pickup or delivery node. Each node is assigned a worker, a day, an hour as the time window, a time and a waiting time in minutes. The routing is modeled as a dial-a-ride problem with time windows.

4.5.1 Heuristic variable neighborhood descent (VND)

The VND is a heuristic that uses local search to explore a finite set of solutions limited by the type of local search used. A local change is accepted as the new best solution if the value of the objective function is smaller (for minimization problems) or larger (for maximization problems) respectively. It does this until no more improvement can be discovered and a local optimum has been found. By changing the type of the local search a solution can be obtained that corresponds to a local optima for all local search neighborhoods. The more neighborhoods are used, the nearer the solution is to the global optimum [8]. The VND accepts only improvements of a solution and is prone to get stuck in a local optimum. Therefore it is often combined with simulated annealing, threshold acceptance or other methods that allow the solution to decrease in quality and avoid local optima [12]. For the problem of this work a threshold acceptance method was used.

As an initial solution a tour is created for each request (a request is a pickup and delivery pair). The initial solution is then improved by the VND with the three neighborhoods move, swap and 2-opt*. The VND improves the solution in the first neighborhood (move) until no more improvement can be found and therefore finds the best solution in the current neighborhood. It then moves to the next neighborhood (swap) to allow the algorithm to escape the local optima. After it found an improvement in the second neighborhood it falls back to the first neighborhood (move) and the process is repeated. Only after no more improvement could be found in the second neighborhood (swap) it moves on to the third neighborhood (2-opt*). Only after no more improvement can be found in the third neighborhood (2-opt*) the VND is terminated. When VND terminates, it found a local optimum for all neighborhoods. This process is illustrated in figure 4 on the following page. To further escape local optima the algorithm allows worse solutions by using a threshold acceptance similar to Polacek et al. [15]. As mentioned before, the VND continues (stops) after no other improvement in the current (last) neighborhood could be found within a fixed number of iterations. Before that however, after a certain (smaller) number of iterations a worse solution is accepted up to a certain threshold. The solution is accepted if the objective value is less than the OV of the best found solution so far plus the threshold. If a solution was accepted, the algorithm jumps back to the first neighborhood. The threshold however is halved every time the third neighborhood is reached. It is therefore reduced with time to allow less and less perturbation. The parameters of the VND and the threshold acceptance have been tuned for this specific problem and multiple problem instances.

4.5.2 Initial routing

For the initial routing solution a direct tour is created from the depot, to the pickup, to the delivery, and back to the depot. The times of pickup and delivery are chosen so that the driving time is minimized and no waiting time occurs for the employee. Waiting time occurs if a worker is forced to wait at a working location either between delivery and begin of shift or between end of shift and pickup. There is no waiting time if a worker is picked-up or delivered from/to his/her home. For the initial routing the best possible times are selected. If a worker gets picked-up from home and delivered to a working location the latest possible time is selected for pickup and delivery to avoid waiting time. Delivery will be at the full hour when his/her shift

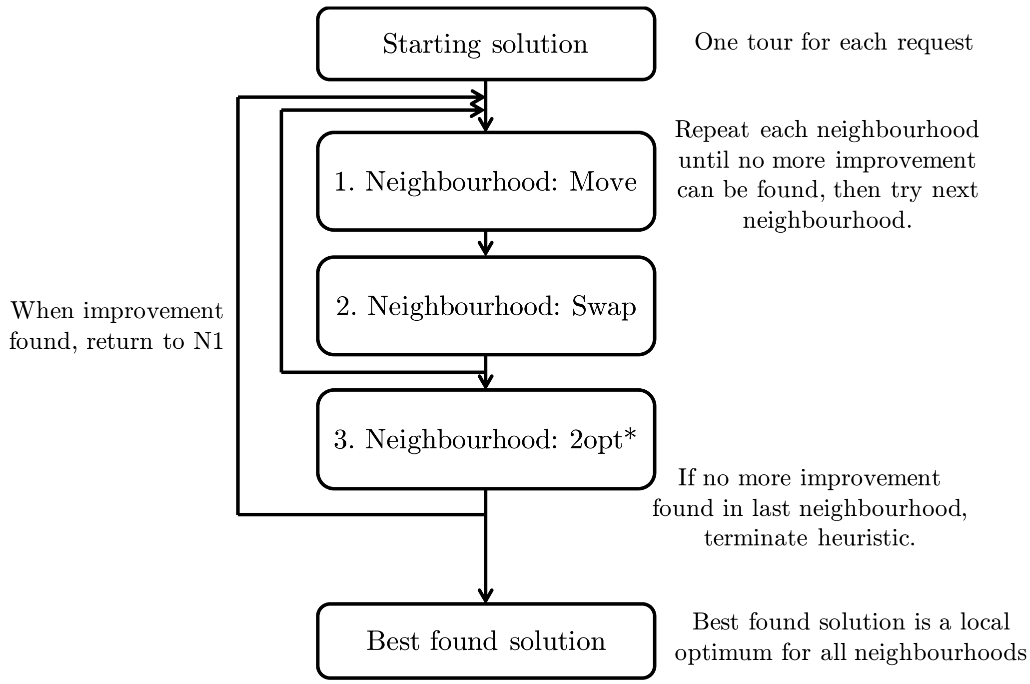


Figure 4: Route improvement with VND (without threshold acceptance)

starts and pickup exactly the time before that it takes to reach his workplace (distance in minutes). An example: The shift begin for a worker is 8 a.m. The distance[19] between his/her home location and the workplace is 22 minutes. The worker will be picked-up at 7:38 a.m. from home and delivered at 8:00 a.m. at the working location.

4.5.3 Route improvement

If any local move should be performed depends on the feasibility and decrease of objective value. A move is feasible if all pickups and deliveries occur within their respective time windows and the time spans between the nodes are at least as big as the travelling time (distance) between them. To avoid unnecessary calculations only the objective values of the two affected routes are calculated and compared. The objective value of a single route f_r is given by

$$f_r = c_V + l^{(r,d)} + \sum_{w=1}^{W} t^{(w,d,r)}. \tag{8}$$

[19] All distances are given in minutes.

For each route we add: the fix costs c_V, the total driving time of this route on this day $l^{(r,d)}$, and the total waiting time of all the workers on this day for this route. The partial objective values of both affected routes are compared to evaluate the quality of the move. A comparison of both complete routing solutions is not necessary since the objective values of the routes that have not been changed are not affected.

The first neighborhood is the move. For the move a random tour and request within that tour is selected. The algorithm tries to move this request to another random tour. The new cost are calculated for the tour where the request is removed. If it was the only request and the tour is now empty the new cost are zero because a tour without requests gets deleted to minimized the number of vehicles and reduce the overall fix costs. If other requests are still present the change in distance is calculated and a possible change in waiting times for the other requests. Also, the other remaining requests in this hour are moved to the best possible position to minimize waiting times without changing their order. Remaining requests in other hours on the same tour are not allowed to be moved at all. This may lead to waiting times even if only one request remains in the same hour. This can happen since the requests of the other hours are not allowed to be moved. For example, a delivery to a workplace can have waiting time because the car needs a certain amount of time to reach the next pickup that is scheduled in the next hour. The new costs are also calculated for the tour where the move is inserted. If the hour of the recipient tour is empty the best possible times are calculated considering the requests in the hours before and after the current hour. If the hour already contains one or more requests, all possible positions are considered without changing the order of the nodes already present in the tour but with adapting the pickup and delivery times to the best possible configuration. Therefore there are 6 possible combinations for the order of the requests if there is already one request present in the recipient tour in the current hour, and pickup must always occur before delivery. A schematic illustration can be found in figure 5 on the next page. The number of possible combinations grows quadratically with the number of requests in one hour. It is however limited by the time grid and the maximum driving time of one hour. The move is performed if it is feasible and leads to a cost decrease in the two routes affected. Since requests are tied to a specific hour, moves within the same tour are not performed.

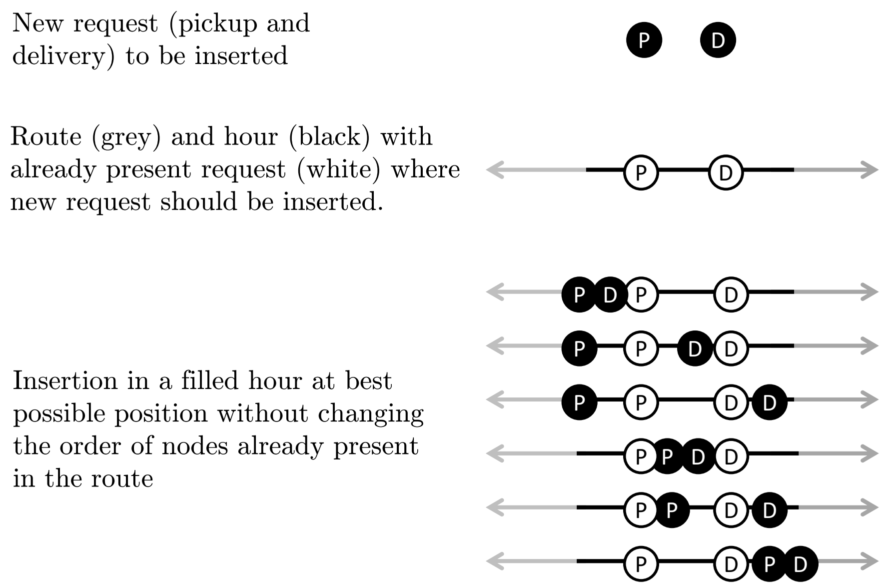


Figure 5: Request insertion

The second neighborhood is the swap. For the swap two random tours and requests are selected. Since all requests have to be served within a specific hour, both requests have to be scheduled in the same hour. If the swap is feasible and leads to a cost decrease it is performed. The calculation and insertion of both requests are performed with the same algorithm as the move.

For the third neighborhood a simplified 2-opt* is used. A randomly selected tours is cut at a random hour and the sequences of requests are swapped between them in the same order. An attachment in reverse order is not possible because of the time window constraints. For the 2-opt* only the pickup and delivery times of the requests in the randomly selected hour are recalculated to ensure feasibility. All other request remain as they were in their original route. The order within the hour is not changed.

The goal of the routing is to minimize the objective value as given in equation equation (4) on page 12. The OV is influenced by the travel time, the number of vehicles and the total waiting time for each worker until shift begin or pick-up.

5 Computational results

Other papers with similar problems rarely used benchmark instances. Mostly the research was based on real world problems or self generated instances were used. [21] The most used benchmark instances similar to this problem where the ones from the NRP competition 2010 or the instances for the VRPTW (Vehicle routing problem with time windows) from Solomon (1987) [19]. Since the scheduling constraints are often unique and in any case very individual, general problem instances can rarely be used. Therefore problem instances are often self generated or benchmark instances adapted.

Due to the unique constraints and the real world origin of this problem self generated instances were used for the problem described in this work.[20] The problem instances were randomly generated based on the interviews conducted on a possible workforce[21] and the companies in the tourism sector in Pongau. Big and small instances were generated. Big instances consist of 20 working locations, 76 workers, 10 skills, 7 days/week, 24 hours/day and about 650 shifts. Small instances consist of 5 working locations, 50 workers, 10 skills, 7 days/week, 24 hours/day and about 70 shifts. Each instance is provided in a flexible and a restrictive version. Flexible instances correspond exactly to the problem description specifying a preferred range of required persons and hours for the shifts, whereas restrictive instances specify a precise number of required persons and hours for each shift.

The algorithm was written in Java and calculated on a computer running on Windows 8.1, with an Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz processor with 8.00 GB RAM and on the Vienna Scientific Cluster 3. The output is created using .txt files. These are then copied to MS Excel and automatically formatted using conditional formatting.

[20] Random instances of different sizes were generated and provided by the AIT Austrian Institute of Technology, Mobility Department, Dynamic Transportation Systems, 1210 Vienna.

[21] The possible workforce were persons currently registered at the Austrian Arbeitsmartkservice AMS and therefore unemployed.

5.1 Solution evaluation using an upper bound

Since the objective function is comprised of weighted penalties a direct comparison between instances is not possible. Therefore the objective values of the best found solution is compared to the upper bounds and therefore to the worst feasible solution possible. The upper bound for the schedule ($f_{S,UB}$) is basically an empty schedule. It is calculated by the sum of the penalties of not fulfilled minimum persons and hours for each shift and the penalties for the not fulfilled minimum required working hours per week. The upper bound for the routing ($f_{R,UB}$) is the worst possible solution where a separate vehicle for each transportation request is used. Results for the big instances can be found in table 3 and results for the small instances can be found in table 4. The penalties used for all instances can be found in table 2.

Variable	Meaning	Penalty Value
$P_{W,hours,max}$	Penalty for exceeding preferred working hours of workers	100
$P_{W,hours,min}$	Penalty for falling below preferred working hours of workers	100
P_{break}	Penalty for breaks	10
P_{sc}	Penalty for skill changes in a single day	10
P_{lc}	Penalty for location changes in a single day	10
$P_{workers,max}$	Penalty for exceeding maximum allowed workers per shift per hour	100
$P_{workers,min}$	Penalty for falling below minimum allowed workers per shift per hour	100
$P_{hours,max}$	Penalty for exceeding maximum allowed working hours per shift per hour	30
$P_{hours,min}$	Penalty for falling below minimum allowed working hours per shift per hour	30

Table 2: Penalties

The computational results show, that by allowing a more flexible approach better solution can be obtained. This is clearly visible in the solution of the bigger problem instances. The flexible instances and solutions can fulfill more of the required minimum persons per hour and minimum hours per shift than the restrictive instances. Therefore one can conclude that flexible worker schedules are superior when it comes to fulfilling the requirements of both employees and employers while still assuring enough workforce. The solution objective values can be found in tables 4 on page 34 and 3 on the following page. The flexible instances are up to 48 percent better than restrictive instances. The sample shift schedules (figure 20 on page xiv and figure 22 on page xvi in the appendix) show clearly that with flexible problem instances there are far less shifts where the minimum required working hours could not be fulfilled than with restrictive problem instances. This is also true for the worker schedules. Workers might not always reach their required minimum working hours per week, but in general they work more with flexible instances.

5.2 Routing parameters

The constraints for the routing can be found in table 5. The parameters for the threshold acceptance embedded in the VND metaheuristic of the routing can be found in table 6 and were tuned as follows: All parameters except one were fixed. The not fixed parameter was set to a number of values. For each parameter setting the VND was applied to 10 test instances 10 times. After each setting the value that performed best on average (smallest average partial objective value) was fixed. The parameter for the number of iterations until a worse solution is allowed once was always in proportion to the number of total iterations. The values 1/2, 1/3, 2/3, 1/4 and 3/4 of the total number of iterations were tried. The result is 1/3 of 7500 iterations.

Constraint	Minutes
Maximum route total length	480[a]
Maximum route driving length	480
Fixed cost of vehicle	100
Waiting time per minute	1

Table 5: Constraints routing

[a]480 Minutes equals to 8 hours.

Instance	locations	workers	$f_{S,UB}$	f_S	%_S[a]	$f_{R,UB}$	f_R	%_R[b]	f_{UB}	f	%_total[c]
0_flexible	20	76	1 025 020	218 320	78,7%	104445	47102	54,9%	1129465	265422	76,5%
0_restrictive	20	76	1 555 430	1 167 990	24,9%	129798	51248	60,5%	1685228	1219238	27,7%
1_flexible	20	76	1 397 010	660 710	52,7%	120875	50301	58,4%	1517885	711011	53,2%
1_restrictive	20	76	1 428 070	1 045 620	26,8%	126406	52764	58,3%	1554476	1098384	29,3%
2_flexible	20	76	1 651 710	929 930	43,7%	132886	54756	58,8%	1784596	984686	44,8%
2_restrictive	20	76	1 698 750	1 328 920	21,8%	122359	47010	61,6%	1821109	1375930	24,4%
3_flexible	20	76	1 446 250	692 120	52,1%	133649	57824	56,7%	1579899	749944	52,5%
3_restrictive	20	76	1 477 520	1 086 560	26,5%	133411	56047	58,0%	1610931	1142607	29,1%
4_flexible	20	76	1 564 950	787 840	49,7%	136820	60419	55,8%	1701770	848259	50,2%
4_restrictive	20	76	1 597 010	1 190 930	25,4%	129206	50615	60,8%	1726216	1241545	28,1%

Table 3: Computational results, big instances, 20 000 iterations

[d]

[a] Percentage that indicates how much better the solution objective value is compared to the upper bound. The higher the value is, the better is the solution.

[b] See footnote a

[c] See footnote a

[d] $f_{S,UB}$= Upper bound schedule, f_S = Objective value schedule, $f_{R,UB}$ = Upper bound routing, f_R = Objective value routing, f_{UB} = Upper bound total solution, f = Objective value total solution.

Instance	locations	workers	$f_{S,UB}$	f_S	%_S[a]	$f_{R,UB}$	f_R	%_R[b]	f_{UB}	f	%_total[c]
0_flexible	5	50	243 450	4 730	98,1%	31 252	14 833	52,5%	274 702	19 563	92,9%
0_restrictive	5	50	229 350	6 380	97,2%	34 734	13 849	60,1%	264 084	20 229	92,3%
1_flexible	5	50	245 450	4 640	98,1%	32 761	15 615	52,3%	278 211	20 255	92,7%
1_restrictive	5	50	231 350	5 350	97,7%	34 502	13 966	59,5%	265 852	19 316	92,7%
2_flexible	5	50	240 750	4 750	98,0%	34 837	14 943	57,1%	275 587	19 693	92,9%
2_restrictive	5	50	226 650	4 570	98,0%	38 350	16 408	57,2%	265 000	20 978	92,1%

Table 4: Computational results, small instances, 20 000 iterations

[d]

[a]Percentage that indicates how much better the solution objective value is compared to the upper bound. The higher the value is, the better is the solution.

[b]See footnote *a*

[c]See footnote *a*

[d]$f_{S,UB}$= Upper bound schedule, f_S = Objective value schedule, $f_{R,UB}$ = Upper bound routing, f_R = Objective value routing, f_{UB} = Upper bound total solution, f = Objective value total solution.

Parameter	Value
Number of continuous unsuccessful iterations until VND is terminated	7500
Number of iterations until a worse solution is allowed once	2500
Threshold (percentage a solution is allowed to be worse)	15%

Table 6: Parameters VND and threshold acceptance used for the routing

5.3 LNS operator performance

The LNS uses three destroy and two repair operators. To evaluate the performance of each operator all possible combinations were tested with 1000 iterations and all four schedule construction methods. The combinations where as follows: Random destroy with initial repair, with coverage repair, and with both repair operators. Worker penalty destroy with initial repair, with coverage repair and with both repair operators. Shift penalty destroy with initial repair, with coverage repair or with both repair operators. All destroy operators with initial repair, coverage repair or with all repair operators. All graphs are depicted in figure 6 on page xi and 7 on page xii in the appendix. Following conclusions can be drawn: The coverage repair operator works always a little bit better than the initial repair repair operator. The worker penalty destroy operator works a bit better than the other two destroy operators. However, the best results are obtained if all operators are combined. Figure 8 on page xiii shows that improvements can be found until about 20000 iterations. For the computational results in table 3 on page 33 and 4 on the previous page only 10000 iterations (for the small instances) or 2500 iterations (for the big instances) were used since the computational time already reached on average 30 hours.

6 Conclusion

This work presented two combined metaheuristics to solve a scheduling and routing problem with a real world origin motivated by the difficulties in the Austrian tourism sector. The heuristics used (LNS for the scheduling and VND for the routing) seem to work quite good compared to the worst possible feasible solution, that is calculated as an upper bound and depicted in the tables of the computational results. However, to truly evaluate the quality of the solution an exact calculation of the optimal solution of the smaller instances is necessary.

The solution obtained is a system optimal solution where individual schedules are not as good as they could be for the benefit of the overall solution. Breaks, skill changes and location changes pose an inconvenience for the workers but allow to better satisfy the demand of the employers and the required working hours of the employees. The bundling of workers for the routing leads to less flexibility concerning the schedules, longer driving times and longer waiting times for the workers but to less overall driving time and therefore to less overall routing cost. The needs of the employers and of the employees have to be balanced. Also, a good schedule and a good routing have to be balanced since all of those requirements express contrasting characteristics. This balance has been establish by setting the penalties as seen in table 2 on page 31. Those penalties can be thus set as to shift the objective to mostly (or only) consider the requirements of one party.

As the proposed problem is quite simplified an expansion of the routing in particular would be quite interesting. The incorporation of footpaths, public transport and private owned cars would make this problem much more realistic. The challenges with this expansion would lie in the inclusion in the algorithm as well as in the necessary data collection and processing. For the inclusion of footpaths a second distance matrix is necessary. For the inclusion of public transport the coordinates of all bus stops, the timetable of each line and all driving times have to be determined. For the inclusion of private owned cars following additional information is necessary: An assignment of workers to households, an assignment of cars to households, whether a worker owns a car and has a drivers license, whether a worker can use a car that is assigned to his/her household, and the number of parking spots reserved for the workforce at each working location.

For even more flexibility in the solutions the time grid could be increased to a 30 minute or even a 15 minute time grid. This however would increase computational time. Also, a decrease in time blocks is imaginable if the requirements of employers and employees fit. For both changes however a slight adaption of the routing is necessary since momentarily the time windows for the routing correspond to the hourly time grid of the overall solution.

Furthermore a practical evaluation of the proposed solutions by the affected employees and employers would be of interest. With a constructive feedback the penalty weights of each inconvenience could be adjusted. This would lead to schedules that are more precisely tailored to the needs of both employees and employers.

7 Acknowledgments

The author wishes to thank the AIT[22] for providing the data collection and the generation of the problem instances. I would like to thank Pamela C. Nolz, Ulrike Ritzinger and Karl F. Doerner[23] for helpful discussions. Furthermore I thank especially David Müller and Briseida Sarasola for detailed discussions, encouragement and support. This work has been partially funded by the Austrian Federal Ministry for Transport, Innovation and Technology (BMVIT) within the program Talents under grant 844869 (project TEPMOS, project partners: Mobilitätszentrale Pongau, HERRY Consult).

References

[1] Andrea Attanasio, Jean-François Cordeau, Gianpaolo Ghiani, and Gilbert Laporte. Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*, 30(3):377–387, 2004.

[2] David Bredström and Mikael Rönnqvist. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. *European journal of operational research*, 191(1):19–31, 2008.

[3] Eddie Cheng and Jennifer Lynn Rich. A home health care routing and scheduling problem. *Technical report TR98-04, Department of Computational and Applied Mathematics, Rice University*, 1998.

[4] Jean-François Cordeau and Gilbert Laporte. The dial-a-ride problem (darp): Variants, modeling issues and algorithms. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1(2):89–101, 2003.

[22] AIT Austrian Institute of Technology, Mobility Department, Dynamic Transportation Systems, 1210 Vienna

[23] University of Vienna, Department of Business Administration, 1090 Vienna

[5] Jean-François Cordeau and Gilbert Laporte. A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 37(6):579–594, 2003.

[6] Jean-François Cordeau and Gilbert Laporte. The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, 153(1):29–46, 2007.

[7] Sana Dahmen and Monia Rekik. Solving multi-activity multi-day shift scheduling problems with a hybrid heuristic. *Journal of Scheduling*, pages 1–17, 2014.

[8] Pierre Hansen and Nenad Mladenović. Variable neighborhood search: Principles and applications. *European journal of operational research*, 130(3):449–467, 2001.

[9] Brigitte Jaumard, Frederic Semet, and Tsevi Vovor. A generalized linear programming model for nurse scheduling. *European journal of operational research*, 107(1):1–18, 1998.

[10] PM Koeleman, Sandjai Bhulai, and M Van Meersbergen. Optimal patient and personnel scheduling policies for care-at-home service facilities. *European Journal of Operational Research*, 219(3):557–563, 2012.

[11] Attila A Kovacs, Sophie N Parragh, Karl F Doerner, and Richard F Hartl. Adaptive large neighborhood search for service technician routing and scheduling problems. *Journal of scheduling*, 15(5):579–600, 2012.

[12] Nenad Mladenović and Pierre Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997.

[13] Sophie N Parragh, Karl F Doerner, and Richard F Hartl. A survey on pickup and delivery problems. part ii: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, 58(2):81–117, 2008.

[14] David Pisinger and Stefan Ropke. Large neighborhood search. In *Handbook of metaheuristics*, pages 399–419. Springer, 2010.

[15] Michael Polacek, Richard F Hartl, Karl Doerner, and Marc Reimann. A variable neighborhood search for the multi depot vehicle routing problem with time windows. *Journal of heuristics*, 10(6):613–627, 2004.

[16] Gerhard Post, Luca Di Gaspero, Jeffrey H Kingston, Barry McCollum, and Andrea Schaerf. The third international timetabling competition. *Annals of Operations Research*, pages 1–7, 2013.

[17] Stefan Ropke and David Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, 40(4):455–472, 2006.

[18] Paul Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *Principles and Practice of Constraint Programming - CP98*, pages 417–431. Springer, 1998.

[19] Marius M Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2):254–265, 1987.

[20] Christos Valouxis, Christos Gogos, George Goulas, Panayiotis Alefragis, and Efthymios Housos. A systematic two phase approach for the nurse rostering problem. *European Journal of Operational Research*, 219(2):425–433, 2012.

[21] Jorne Van den Bergh, Jeroen Beliën, Philippe De Bruecker, Erik Demeulemeester, and Liesje De Boeck. Personnel scheduling: A literature review. *European Journal of Operational Research*, 226(3):367–385, 2013.

A Sample instances

A.1 Worker information

Worker	1	2	3	4	...
Min working hours/week	30	40	35	45	...
Max working hours/week	40	48	40	48	...
Skills	4	4	4,8	8	...
Availability Monday	09:00-18:00	08:00-17:00	00:00-24:00	09:00-17:00	...
Tuesday	09:00-18:00	08:00-17:00	00:00-24:00	09:00-17:00	...
Wednesday	09:00-18:00	08:00-17:00	00:00-24:00	09:00-17:00	...
Thursday	-	08:00-17:00	00:00-24:00	09:00-17:00	...
Friday	-	08:00-17:00	00:00-24:00	09:00-17:00	...
Saturday	09:00-18:00	-	00:00-24:00	09:00-17:00	...
Sunday	09:00-18:00	-	00:00-24:00	-	...

Table 7: Sample worker information

A.2 Shift information flexible and restrictive

Location	Day	Time span	Skill	min hours	max hours	min pers.	max pers.
0	0	07:00-19:00	8	8	15	0	2
0	1	08:00-13:00	4	10	15	1	3
0	1	13:00-20:00	4	10	15	0	3
0	3	06:00-20:00	7	10	15	0	5
1	0	08:00-10:00	4	2	8	1	4
1	0	10:00-17:00	4	10	20	0	3
1	2	07:00-15:00	8	8	12	0	3
2	0	07:00-15:00	7	15	20	0	-
...	...	...	...	...	...	...	...

Table 8: Sample shift information flexible instances

Location	Day	Time span	Skill	required hours	required persons
0	0	07:00-19:00	8	12	1
0	1	08:00-13:00	4	10	2
0	1	13:00-20:00	4	14	2
0	3	06:00-20:00	7	14	1
1	0	08:00-10:00	4	6	3
1	0	10:00-17:00	4	14	2
1	2	07:00-15:00	8	8	1
2	0	07:00-15:00	7	16	2
...	...	...	...	...	...

Table 9: Sample shift information restrictive instances

B Sample solutions

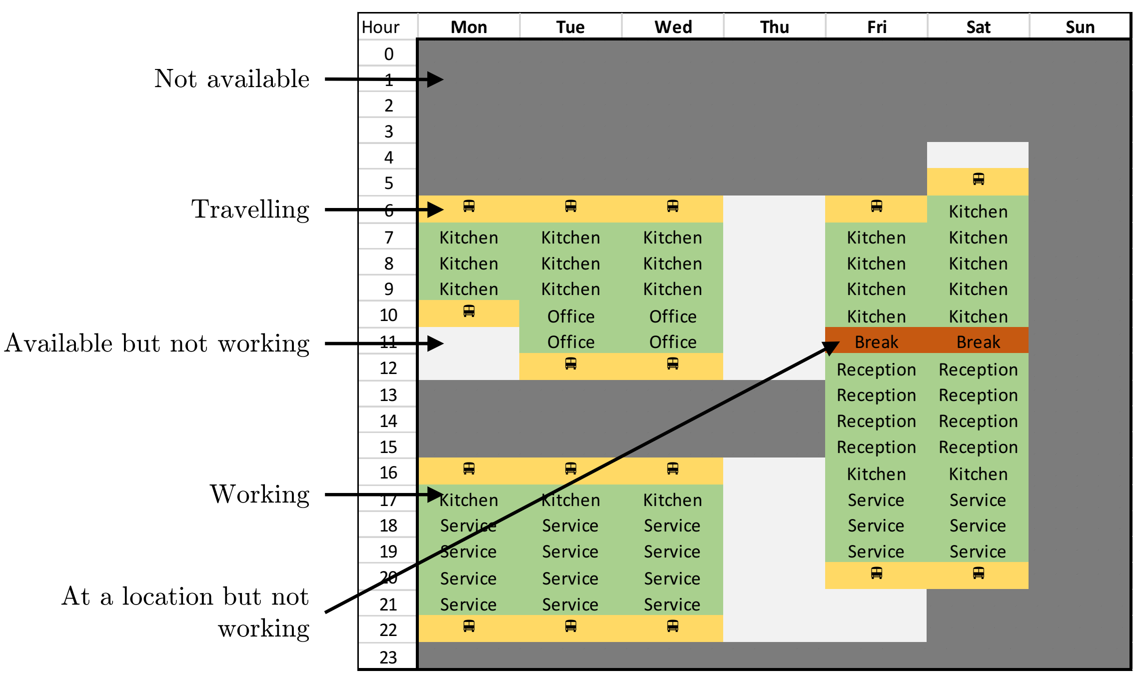


Figure 9: Sample worker schedule

B.1 Worker schedules for flexible instances

The following worker schedules are samples. They where created by the small flexible instance number 0, with 5 locations and 50 workers. There are the following differences between the sample worker schedule as seen in figure 9 and the following worker schedules: For each working hour it is only indicated at which location the worker is working. The first line shows the worker identification. Since all locations have to be unique, the numbering of the working locations starts with 0 and goes until the total amount of working locations minus one. The numbering of workers (or their homes) start right after that. Therefore for this instance the working locations are numbered from 0 to 4 and the worker identifications are numbered from 5 to 55. The second line shows how many hours a worker is actually scheduled to work in comparison to the minimum and maximum desired working hours. The line between the days and the first hour indicates how many skill changes this worker undergoes on this day. Break rooms have a separate skill and count therefore as a skill change. The worker schedules are depicted in figures 10 to 14 on page v in the appendix.

Worker	5						
Working Hours	21/	30-40					
Hours/Days	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Skills	1	1	1	0	0	0	0
0							
1							
2							
3							
4							
5							
6							
7							
8							
9							
10	L1	L1	L1				
11	L1	L1	L1				
12	L1	L1	L1				
13	L1	L1	L1				
14	L1	L1	L1				
15	L1	L1	L1				
16	L1	L1	L1				
17							
18							
19							
20							
21							
22							
23							

Figure 10: Worker schedule 5, flexible

Worker	6						
Working Hours	27/	40-48					
Hours/Days	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Skills	1	3	1	1	1	0	0
0							
1							
2							
3							
4							
5							
6							
7							
8							
9	L1	L0		L1			
10	L1	L0	L1	L1	L1		
11	L1	Break	L1	L1	L1		
12	L1		L1	L1	L1		
13	L1	L1		L1	L1		
14	L1	L1		L1	L1		
15	L1			L1	L1		
16							
17							
18							
19							
20							
21							
22							
23							

Figure 11: Worker schedule 6, flexible

Worker	7						
Working Hours	41/	35-40					
Hours/Days	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Skills	1	1	1	1	1	0	0
0							
1							
2							
3							
4							
5							
6							
7		L0	L0	L0	L0		
8	L1	L0	L0	L0	L0		
9	L1			L0	L0		
10				L0			
11				L0			
12				L0			
13	L0	L0	L0	L0			
14	L0	L0	L0	L0			
15	L0	L0	L0	L0			
16	L0	L0	L0	L0			
17	L0	L0	L0	L0			
18	L0	L0	L0	L0			
19	L0		L0				
20							
21							
22							
23							

Figure 12: Worker schedule 7, flexible

Worker	8						
Working Hours	21/	45-48					
Hours/Days	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Skills	1	3	1	3	1	0	0
0							
1							
2							
3							
4							
5							
6							
7							
8							
9		🚍		🚍	🚍		
10		L0		L0	L1		
11		L0		L0	L1		
12	🚍	L0	🚍	L0	L1		
13	L0	L0	L0	L0	L1		
14	L0	L0	🚍	Break	L1		
15	L0	L0		L0	L1		
16	🚍	🚍		🚍	🚍		
17							
18							
19							
20							
21							
22							
23							

Figure 13: Worker schedule 8, flexible

Worker	9						
Working Hours	36/	38-40					
Hours/Days	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Skills	1	1	0	1	1	0	0
0							
1							
2							
3							
4							
5							
6	🚍	🚍		🚍	🚍		
7	L0	L1		L1	L0		
8	L0	L1		L1	L0		
9	L0	L1		L1	L0		
10	L0	L1		L1	L0		
11	L0	L1		🚍	L0		
12	L0	L1			L0		
13	L0	L1			L0		
14	L0	L1			L0		
15	L0	🚍			L0		
16	L0				L0		
17	L0				L0		
18	L0				L0		
19	🚍				🚍		
20							
21							
22							
23							

Figure 14: Worker schedule 9, flexible

B.2 Worker schedules for restrictive instances

The following schedule examples are from the small restrictive instance 0, with 5 locations and 50 workers. The design is identical to the flexible schedule examples and is explained in the section B.1 on page ii. The worker schedules are depicted in figures 15 to 19 on page viii in the appendix.

Worker	5						
/orking Hou	10/	30-40					
Hours/Days	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Skills	1	1	1	0	0	0	0
0							
1							
2							
3							
4							
5							
6							
7							
8							
9							
10	L0						
11	L0						
12	L0						
13	L0						
14	L0						
15	L0	L0					
16	L0	L0	L1				
17							
18							
19							
20							
21							
22							
23							

Figure 15: Worker schedule 5, restrictive

Worker	6						
/orking Hou	23/	40-48					
Hours/Days	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Skills	1	1	1	1	1	0	0
0							
1							
2							
3							
4							
5							
6							
7							
8	🚌	🚌	🚌	🚌	🚌		
9	L0	L1	L1	L1	L0		
10	🚌	L1	L1	🚌	L0		
11		L1	L1		L0		
12		L1	L1		L0		
13		L1	L1		L0		
14		L1	L1		L0		
15		L1	L1		L0		
16		🚌	🚌		🚌		
17							
18							
19							
20							
21							
22							
23							

Figure 16: Worker schedule 6, restrictive

Worker	7						
/orking Hou	33/	35-40					
Hours/Days	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Skills	1	1	1	1	1	0	0
0							
1							
2							
3							
4							
5							
6							
7	🚌	🚌	🚌	🚌	🚌		
8	L0	L0	L0	L0	L1		
9	🚌	🚌	L0	L0	L1		
10			L0	L0	L1		
11			L0	L0	L1		
12			L0	L0	L1		
13			L0	L0	L1		
14			L0	L0	L1		
15		🚌	L0	L0	L1		
16	🚌	L1	L0	L0	L1		
17	L0	🚌	L0	L0	🚌		
18	🚌		🚌	🚌			
19							
20							
21							
22							
23							

Figure 17: Worker schedule 7, restrictive

Worker	8						
/orking Hou	26/	45-48					
Hours/Days	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Skills	1	2	1	1	2	0	0
0							
1							
2							
3							
4							
5							
6							
7							
8							
9	🚌	🚌		🚌	🚌		
10	L1	L0	🚌	L0	L0		
11	L1	L0	L1	L0	L0		
12	L1	L0	L1	L0	L0		
13	L1	L0	🚌	L0	L0		
14	L1	L0		L0	L0		
15	L1	L0		L0	L0		
16	🚌	🚌		🚌	🚌		
17							
18							
19							
20							
21							
22							
23							

Figure 18: Worker schedule 8, restrictive

Worker	9						
/orking Hou	26/	38-40					
Hours/Days	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Skills	1	1	1	4	1	0	0
0							
1							
2							
3							
4							
5							
6							
7				🚌			
8		🚌	🚌	L0	🚌		
9		L1	L1	Break	L1		
10	🚌	L1	L1	Break	L1		
11	L1	L1	L1	L0	L1		
12	L1	L1	L1	L0	L1		
13	L1	L1	L1	L0	L1		
14	🚌	🚌	🚌	L0	🚌		
15				L0			
16				L0			
17				L0			
18				🚌			
19							
20							
21							
22							
23							

Figure 19: Worker schedule 9, restrictive

B.3 Shift schedules for flexible instances

The shift schedules are depicted as follows: Each row corresponds to a shift. The first columns contain following information: Shift number, location, day, skill required, minimum required person for each hour, total fulfilled hours, and the required hour span. The following columns are the hours of the day (from 0 to 23). The requirements of each shift is given by the required time span, a minimum and maximum number of required workers, and a minimum and maximum of required working hours during that time span. The colors in the solution example can be read as follows: Dark Grey cells indicate that these hours are outside of the required time span. Light Grey cells indicate that these hours are inside the time span, but no worker is scheduled and the minimum required persons constraint is still satisfied.[24] Green cells indicate that the minimum amount of required persons is fulfilled but the maximum amount is not exceeded. Orange cells indicate that the minimum required persons are not fulfilled in that hour. Inside each hour and shift the worker ID that is scheduled is displayed. In the column that indicates the number of fulfilled hours following color is used: Orange if the minimum required hours for this shift is not fulfilled.
The shift schedules for the flexible instances are depicted in figures 20 on page xiv to 21 on page xv in the appendix.

B.4 Shift schedules for restrictive instances

The shift schedules for the restrictive instances are exactly like the shift schedules for the flexible instances and can be found in figure 22 on page xvi to 23 on page xvii in the appendix.

[24]This is the case if the minimum amount of required persons is zero.

B.5 Routes

The routes for each solution are given as a text output. For each day, for each route all the nodes are given with the following information: Is it a pickup or delivery point, the location number (for example L3) or H for the home of a worker, the worker ID (for example W52) that is being picked-up or delivered, the time (@07:19) and the waiting time (WT0 if no waiting time occurs). Additionally for each tour the largest number of simultaneously transported persons is given and if the route is feasible. The distinction between flexible and restrictive instances does not make much difference for the routing. Some example tours can be found in figure 24.

Route/Car 1	
Max # of passengers: 2	
Feasibility true	
- P H W53	@08:29 WT0
- P H W40	@08:58 WT0
- D L0 W53	@09:00 WT0
- D L0 W40	@09:00 WT0
- P L0 W43	@11:00 WT0
- D H W43	@11:39 WT0
- P L1 W37	@13:00 WT0
- D H W37	@13:12 WT0
- P H W51	@15:45 WT0
- D L1 W51	@16:00 WT0
- P L1 W6	@16:00 WT0
- D H W6	@16:14 WT0

(a) Route 1

Route/Car 4	
Max # of passengers: 3	
Feasibility true	
- P H W38	@06:15 WT0
- P H W30	@06:46 WT0
- D L1 W30	@07:00 WT0
- D L1 W38	@07:00 WT0
- P H W21	@10:38 WT0
- D L3 W21	@11:00 WT0
- P L3 W10	@13:00 WT0
- P L3 W49	@13:00 WT0
- P L3 W21	@13:00 WT0
- D H W49	@13:11 WT0
- D H W10	@13:26 WT0
- D H W21	@13:49 WT0

(b) Route 4

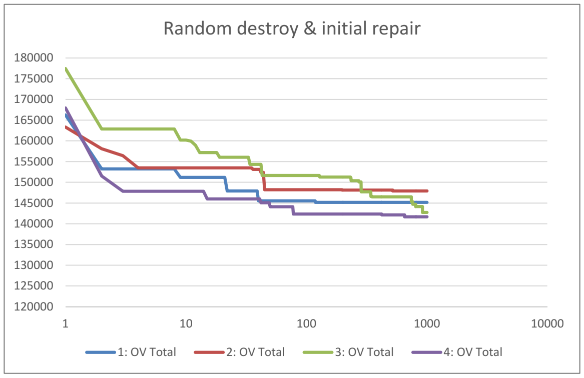
Route/Car 7	
Max # of passengers: 2	
Feasibility true	
- P H W29	@08:17 WT0
- D L0 W29	@09:00 WT0
- P L0 W7	@10:00 WT0
- D H W7	@10:15 WT0
- P L1 W38	@11:00 WT0
- D H W38	@11:18 WT0
- P H W13	@12:31 WT0
- P H W45	@12:39 WT0
- D L0 W13	@13:00 WT0
- D L0 W45	@13:00 WT0
- P L0 W48	@13:00 WT0
- P L0 W29	@13:00 WT0
- D L1 W48	@13:25 WT3!
- D H W29	@13:44 WT0
- P L1 W30	@15:00 WT0
- D H W30	@15:14 WT0

(c) Route 7

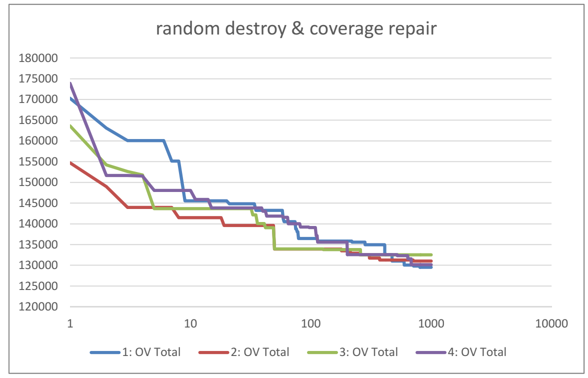
Route/Car 17	
Max # of passengers: 3	
Feasibility true	
- P H W39	@06:02 WT0
- P H W20	@06:12 WT0
- P H W17	@06:23 WT0
- D L2 W17	@07:00 WT0
- D L2 W39	@07:00 WT0
- D L2 W20	@07:00 WT0
- P H W52	@07:26 WT0
- P H W25	@07:50 WT0
- D L1 W25	@08:00 WT0
- D L1 W52	@08:00 WT0
- P H W6	@09:46 WT0
- D L1 W6	@10:00 WT0
- P L1 W13	@10:00 WT0
- P L1 W25	@10:00 WT0
- D H W13	@10:05 WT0
- D H W25	@10:12 WT0

(d) Route 17

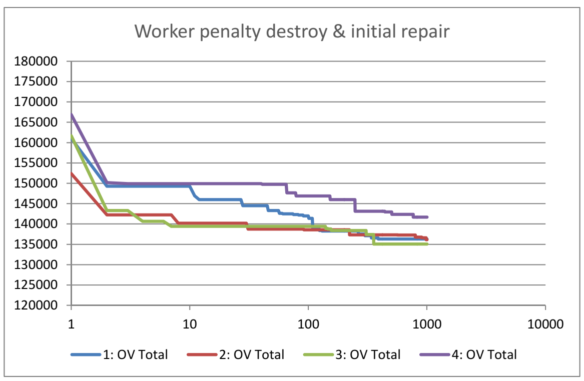
Figure 24: Sample routes, flexible instance, day 4



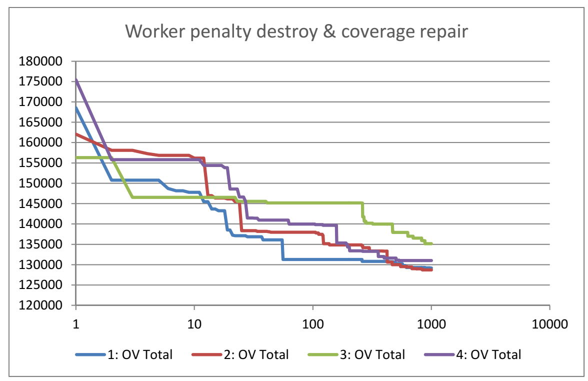
(a) Random destroy and initial repair



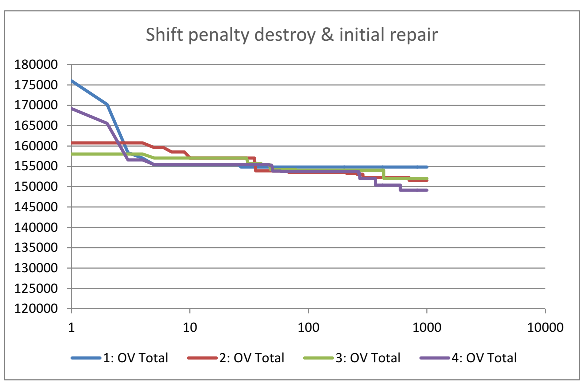
(b) Random destroy and coverage repair



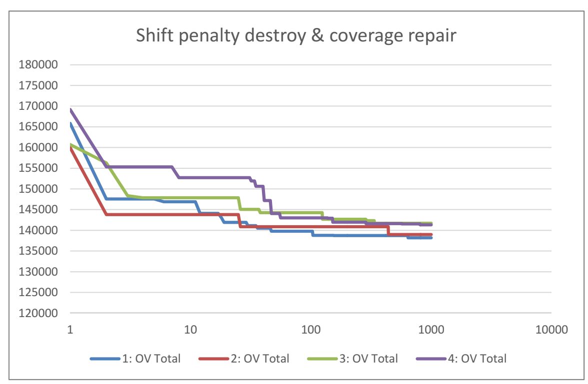
(c) Worker penalty destroy and initial repair



(d) Worker penalty destroy and coverage repair

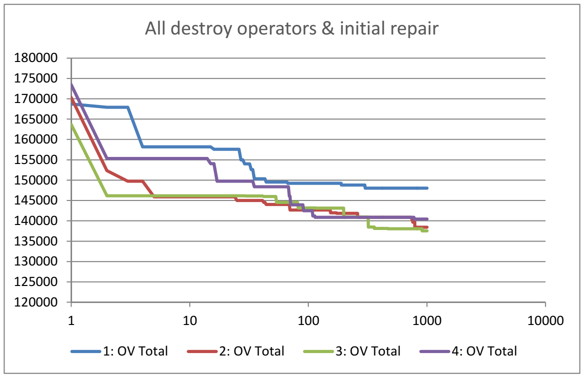


(e) Shift penalty destroy and initial repair

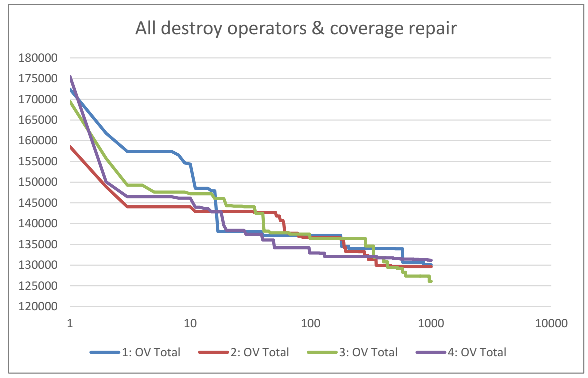


(f) Shift penalty destroy and coverage repair

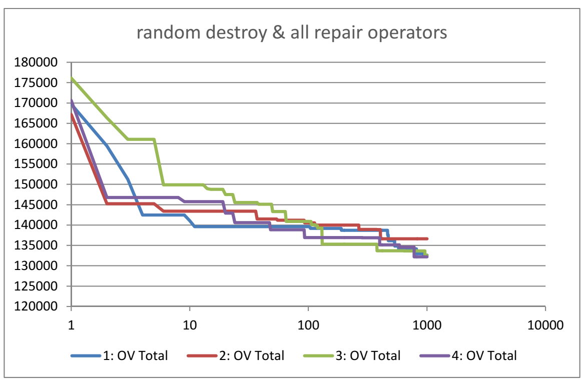
Figure 6: LNS operator performance, individual



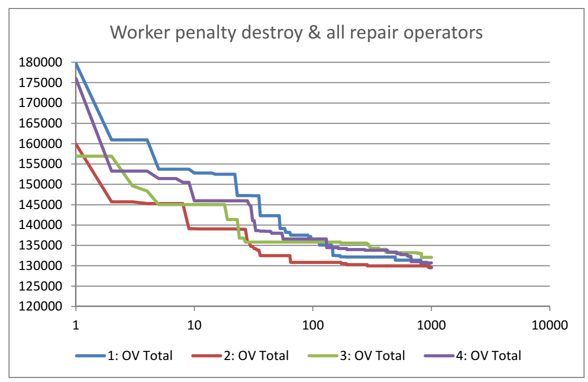
(a) All destroy operators and initial repair



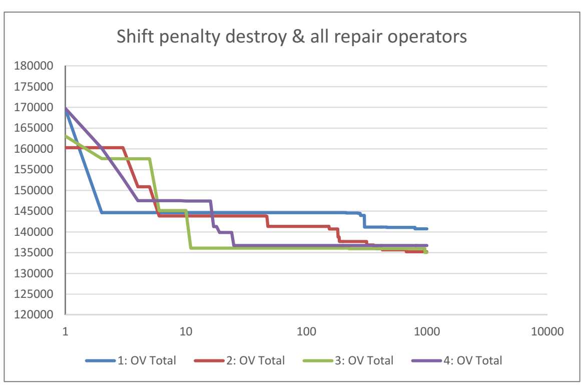
(b) All destroy operators and coverage repair



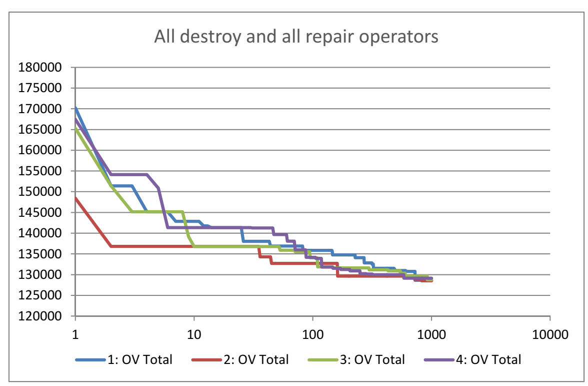
(c) Random destroy and all repair operators



(d) Worker penalty destroy and all repair operators

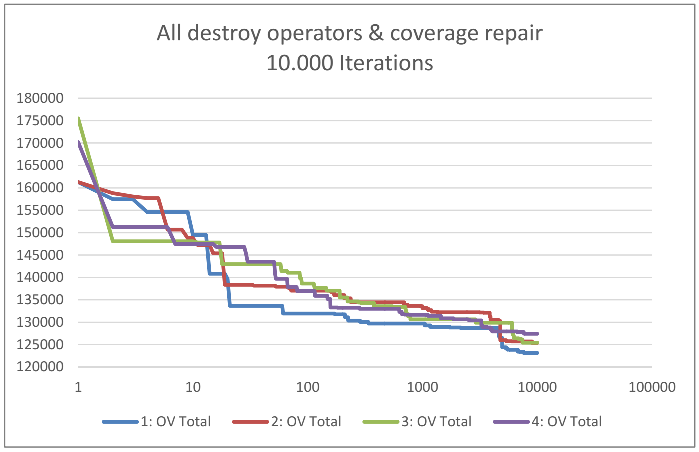


(e) Shift penalty destroy and all repair operators

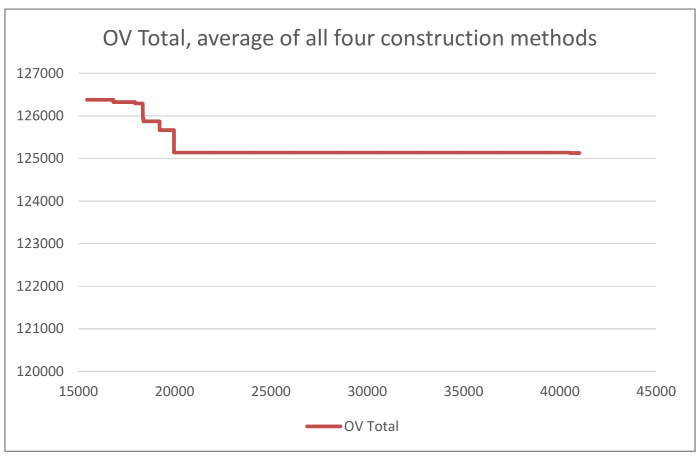


(f) All destroy and all repair operators

Figure 7: LNS operator performance, combined



(a) All operators, 10 000 iterations



(b) Average objective value up to 40 000 iterations

Figure 8: Optimal number of iterations for LNS

Shift	Loc	Day	Skill	Min P	Fulfilled hours		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	0	4	1	14	/10-15									52,15.	52,15,33.	52,15,48.	52,15,48.	52,15,48.											
2	0	0	4	0	15	/10-15														7,27,8.	7,8.	27,7,8.	27,7,15.	7,15.	7.	7.				
5	0	1	4	1	15	/10-15								15,7.	15,7.	15,42,6.	15,8,6.	15,48,51.	15,42.											
6	0	1	4	0	15	/10-15														7,48,25.	7,48,25.	48,8,7.	7.	15,7.	15,7.	15.				
9	0	2	4	1	15	/10-15								7.	25,7.	25,45,42.	25,45,48.	25,45,48.	25,45,48.											
10	0	2	4	0	15	/10-15														7,25,8.	7,25.	7,51.	7,51.	7,15.	7,15.	7,15.				
13	0	3	4	1	14	/10-15								7.	7.	7,37,42.	45,7,8.	45,7,8.	7,8,25.											
14	0	3	4	0	15	/10-15														45,7,8.	45,7,51.	7,8,51.	7,51.	7.	15,7.	15.				
17	0	4	4	1	15	/10-15								15,7.	15,7.	15,7,53.	15,48.	15,27,48.	15,27,48.											
18	0	4	4	0	13	/10-15														15,45,13.	15,45,13.	15,27,13.	15,27.	15.	15.					
3	0	0	7	0	15	/10-15							17.	20,17.	17.	20,17.	17.	20,17.	17.	20,17.	17.	17.	17.							
7	0	1	7	0	15	/10-15							17.	39,20.	17,39.	39,20.	17,39.	20.	17.				17.	17.	17.	17.				
11	0	2	7	0	15	/10-15							17.	39.	39.	19,39,41.	39,41.	41.	41.	20,41,45.	41,45.									
15	0	3	7	0	15	/10-15							17.	20,39,17.	20,17.	45,20,39,17.	20,17.	20.	20.	20.										
19	0	4	7	0	15	/10-15									26.	19,29,26,40.	29.	19,29.	29,17.	19,17.	19,17.	17.								
0	0	0	8	0	15	/8-15								9.	43,9.	43,9.	43,9.	9.	9.	9.	9.	9.	9.	9.	9.					
4	0	1	8	0	15	/8-15								34.	34.	34.	34,48.	34,8.	34,8.	34,8.	34,8.	34.	34.							
8	0	2	8	0	15	/8-15								17,34.	17,34.	17,34.	17.	17.	17.	17.	17.	17.	17.	17.	17.					
12	0	3	8	0	15	/8-15								30.	43,30.	43,30.	43,30.	48,30.	48,30.			14.	14.	14.	14.					
16	0	4	8	0	15	/8-15								9.	43,9.	43,9.	43,9.	9.	9.	9.	9.	9.	9.	9.	9.					

DL	Loca	Day	Skill	Min N	Fulfilled hours		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
23	1	0	1	0	6	/3-6										32.	32,12.	32.	12.		12.									
27	1	1	1	0	6	/3-6											44,12.	44.	44.	44.	44.									
31	1	2	1	0	6	/3-6										32.	46,32.	46,32.	46.											
35	1	3	1	0	6	/3-6											12.		12.		12.		12.		12.		12.			
39	1	4	1	0	1	/3-6										27.														
20	1	0	4	1	6	/2-8									7,25.	7,13,6,37.														
21	1	0	4	0	20	/10-20											5,13,6.	5,13,6.	5,13,6.	5,13,6.	5,13,6.	5,13,6.	5,51.							
24	1	1	4	1	6	/2-8									52,25.	13,33,37,25.														
25	1	1	4	0	20	/10-20											13,33,5.	13,37,5.	13,37,5.	13,5,6.	13,5,6.	13,51,5.	51,5.							
28	1	2	4	1	6	/2-8									15,52.	13,15,52,37.														
29	1	2	4	0	20	/10-20											5,13,6.	5,13,6.	5,13,6.	5,13,52.	5,13,27.	5,13,27.	5,27.							
32	1	3	4	1	7	/2-8									15,52,25.	33,6,53,25.														
33	1	3	4	0	20	/10-20											6,15,13.	6,15,13.	6,15,13.	6,15,13.	6,15,13.	6,15,13.	15,27.							
36	1	4	4	1	6	/2-8									52,25.	37,42,13,25.														
37	1	4	4	0	19	/10-20											8,42,6.	37,8,6.	37,8,6.	8,42,6.	8,6,48.	8,6,48.	51.							
22	1	0	8	0	12	/8-12								30,18,34.	18,30,34.	18,30.	18.	18.	30.	30.										
26	1	1	8	0	12	/8-12								18,9.	18,9.	18,9.	18,9.	9.	9.	9.	9.									
30	1	2	8	0	12	/8-12								30.	30.	14,30.	14,30.	14,30.	14,30.	14.	14.									
34	1	3	8	0	12	/8-12								34,9.	34,9.	34,9.	34,9.	34.	34.	34.	34.									
38	1	4	8	0	12	/8-12								30,38.	30,38.	30,38.	30,38.	30.	30.	30.	30.									

Figure 20: Shift schedule for location 0 and 1, flexible

DL	Loca	Day	Skill	Min N	Fulfilled hours		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
41	2	0	1	0	15	/10-15										22,27.	44,22.	22,44.	44,22.	44,22.	44,22.	22.	12.		12.					
44	2	1	1	0	15	/10-15										22,32.	22,32.	22,32.	12.	27,22.	12,27.	22.	12,27.		12.					
47	2	2	1	0	14	/10-15									46.	22.	44,22.	22,44.	44,22.	44,22.	44,22.	22.				46.				
50	2	3	1	0	14	/10-15										22,27.	44,22.	22,27.	44,22.	22,27.	44,22.	22.				46.				
53	2	4	1	0	13	/10-15									46.	32,22.	32,22.	32,22.	46,22.	22.	22.	22.				46.				
42	2	0	5	1	20	/15-20									46.	46,35.	27,46,35.	27,46.	46.	47.	47.	47.	47.	47,12.	47.	12,46.	12,46.	~		
45	2	1	5	1	20	/15-20									46.	27,35,46.	52,27,46.	27,46.	52,27.	47,12.	~	27,47,12.	~	47,12.	~	12.	12.	~		
48	2	2	5	1	20	/15-20									~	27.	27,12.	27,52,12.	27,12.	47,12.	47,12.	47,12.	47,12.	47,12.	47,12.	~	~	~		
51	2	3	5	1	19	/15-20									46.	46.	52,27,46.	52,46.	52,27,46.	47,52.	27,47.	47.	47.	47,35.	47.	~	~	~		
54	2	4	5	1	20	/15-20									~	46.	35,46,52,12.	46,52,12.	52,12.	52,12.	35,12.	35,12.	35,12.	35.	~	~	46.	~		
40	2	0	7	0	20	/15-20									26.	41,26,40,45,29.	26,41,45.	41,26,45.	26,45.	41,26,45.	26,41,45.									
43	2	1	7	0	20	/15-20									26.	41,45,29,26.	41,45,29,26.	41,45,29.	41,45,29.	41,45.	41,17,45.									
46	2	2	7	0	18	/15-20								20.	20,26.	20,40,33,29.	20,40,33,29.	20,19,29.	19,29.	19.	19.									
49	2	3	7	0	20	/15-20									26.	26,19,41.	26,19,41.	26,19,41.	26,19,41,17.	26,19,41.	26,19,17.									
52	2	4	7	0	20	/15-20								20,39,17.	20,39,17.	20,33,39,45,17.	20,33,39,17.	20.	20.	20,41.	41.									

DL	Loca	Day	Skill	Min N	Fulfilled hours		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
55	3	0	3	0	10	/5-10									10.	10.	36,49,10.	21,49,10.	21,49.											
56	3	1	3	0	10	/5-10									10.	10.	10,49.	10,49,21.	10,49,21.											
57	3	2	3	0	10	/5-10									10.	10.	49,10,36.	21,10.	21,49,10.											
58	3	3	3	0	10	/5-10									10.	10.	10,49,36.	10,21.	10,49,21.											
59	3	4	3	0	10	/5-10									10.	10.	36,10,49.	21,10.	21,10,49.											

DL	Loca	Day	Skill	Min N	Fulfilled hours		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
60	4	0	6	0	15	/9-15								31,50.	24.	24,31.	24,50.	24,31.	24,50.	24.	24.	36,24.								
61	4	1	6	0	15	/9-15								31,50.	50,24.	31,50.	50.	31,50.	31,50.	50.	50.	36,50.								
62	4	2	6	0	15	/9-15								50.	24,50.	24,50.	24,50.	24,50.	24,50.	24,50.	24.	24.								
63	4	3	6	0	15	/9-15								50.	50,24.	50,24.	50,24.	50,24.	50,24.	50,24.	24.	24.								
64	4	4	6	0	15	/9-15								50.	24,50.	24,50.	24,50.	24,50.	24,50.	24,50.	24.	24.								

Figure 21: Shift schedule for location 2, 3 and 4, flexible

Shift	Loc	Day	Skill	Min P	Fulfilled hours		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	0	0	4	2	20	/20-20									15,52.	27,15.	27,7.	27,7.	27,7.	27,7.	27,7.	27,5.	27,5.	15,7.						
4	0	1	4	2	20	/20-20									25,52.	6,25.	6,5.	6,5.	6,5.	6,5.	6,5.	6,5.	27,5.	15,7.						
7	0	2	4	2	19	/20-20									7~	6,7.	5,6.	5,6.	5,6.	5,6.	5,6.	5,6.	5,7.	15,7.						
10	0	3	4	2	20	/20-20									15,7.	15,7.	15,7.	15,7.	15,7.	15,7.	15,7.	15,7.	15,7.	15,7.						
13	0	4	4	2	20	/20-20									7,15.	7,15.	7,15.	7,15.	7,15.	45,7.	45,7.	7,15.	7,15.	7,15.						
2	0	0	7	1	10	/10-10									17.	40.	40.	20.	17.	20.	17.	17.	17.	17.						
5	0	1	7	1	10	/10-10									39.	41.	45.	29.	29.	41.	41.	17.	17.	17.						
8	0	2	7	1	10	/10-10									17.	41.	41.	41.	41.	26.	41.	17.	17.	17.						
11	0	3	7	1	10	/10-10									17.	17.	17.	17.	17.	17.	17.	17.	17.	17.						
14	0	4	7	1	9	/10-10									17.	45.	45.	45.	45.	20.	~	17.	17.	17.						
0	0	0	8	1	10	/10-10									30.	30.	30.	30.	30.	30.	30.	30.	30.	30.						
3	0	1	8	1	10	/10-10									34.	34.	34.	34.	34.	34.	34.	34.	34.	34.						
6	0	2	8	1	10	/10-10									34.	17.	17.	17.	30.	30.	30.	30.	34.	34.						
9	0	3	8	1	10	/10-10									9.	9.	9.	9.	9.	9.	9.	9.	9.	9.						
12	0	4	8	1	10	/10-10									43.	43.	9.	9.	9.	9.	9.	14.	14.	14.						

Shift	Loc	Day	Skill	Min P	Fulfilled hours		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
17	1	0	1	1	5	/5-5									46.	22.	12.	22.	12.											
20	1	1	1	1	4	/5-5									~	22.	12.	22.	22.											
23	1	2	1	1	5	/5-5									46.	32.	27.	27.	27.											
26	1	3	1	1	4	/5-5									~	27.	27.	27.	27.											
29	1	4	1	1	5	/5-5									46.	22.	12.	12.	12.											
15	1	0	4	2	17	/18-18									7,25.	6,13.	5,13.	5,13.	5,15.	5,6.	6,13.	6,13.	51~							
18	1	1	4	2	17	/18-18									7,15.	7,15.	7,15.	7,15.	7,15.	7,15.	7,15.	7,15.	51~							
21	1	2	4	2	18	/18-18									15,25.	15,27.	15,13.	15,13.	15,13.	15,13.	15,8.	15,8.	27,51.							
24	1	3	4	2	18	/18-18									25,52.	13,25.	13,25.	13,25.	13,25.	8,13.	8,13.	8,13.	27,51.							
27	1	4	4	2	17	/18-18									25~	6,13.	6,13.	6,13.	6,13.	6,13.	6,13.	6,13.	27,51.							
16	1	0	8	2	10	/10-10										14,43.	14,43.	14,9.	14,8.	14,9.										
19	1	1	8	2	10	/10-10										9,14.	9,8.	9,8.	9,8.	9,8.										
22	1	2	8	2	10	/10-10										9,14.	8,9.	8,9.	8,9.	8,9.										
25	1	3	8	2	10	/10-10										14,18.	14,8.	14,8.	14,8.	14,18.										
28	1	4	8	2	10	/10-10										14,30.	14,48.	14,8.	14,8.	14,8.										

Figure 22: Shift schedule for location 0 and 1, restrictive

Shift	Loc	Day	Skill	Min P	Fulfilled hours		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
31	2	0	1	1	8	/9-9									~	32.	46.	32.	46.	44.	22.	22.	12.							
34	2	1	1	1	8	/9-9									~	27.	44.	44.	44.	44.	44.	22.	12.							
37	2	2	1	1	8	/9-9									~	22.	12.	22.	12.	22.	12.	22.	12.							
40	2	3	1	1	8	/9-9									~	22.	12.	12.	12.	12.	12.	22.	12.							
43	2	4	1	1	8	/9-9									~	32.	32.	32.	22.	22.	12.	22.	12.							
32	2	0	5	1	10	/12-12									~	~	52.	46.	52.	47.	12.	47.	47.	47.	12.	46.				
35	2	1	5	1	12	/12-12									46.	46.	46.	46.	12.	47.	47.	12.	47.	12.	47.	46.				
38	2	2	5	1	12	/12-12									52.	52.	52.	52.	52.	47.	27.	47.	47.	35.	47.	12.				
41	2	3	5	1	12	/12-12									46.	46.	46.	52.	46.	52.	47.	47.	47.	47.	47.	46.				
44	2	4	5	1	12	/12-12									52.	27.	27.	46.	27.	27.	27.	12.	35.	12.	12.	12.				
30	2	0	7	3	18	/18-18									26,20,39.	29,26,33.	29,26,17.	29,26,41.	29,26,41.	26,41,45.										
33	2	1	7	3	18	/18-18									26,20,17.	40,20,29.	40,26,20.	26,41,20.	26,20,46.	26,20,17.										
36	2	2	7	3	18	/18-18									20,39,26.	19,20,39.	19,20,39.	19,20,29.	19,20,45.	19,17,20.										
39	2	3	7	3	18	/18-18									26,20,39.	26,19,29.	26,20,33.	26,46,19.	26,20,29.	26,19,41.										
42	2	4	7	3	18	/18-18									20,39,26.	33,19,20.	19,20,40.	19,20,29.	19,29,17.	19,17,41.										

Shift	Loc	Day	Skill	Min P	Fulfilled hours		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
45	3	0	3	2	6	/6-6										10,35.	10,35.	10,44.												
46	3	1	3	2	6	/6-6										10,35.	36,10.	10,49.												
47	3	2	3	2	5	/6-6										10~	36,44.	10,49.												
48	3	3	3	2	5	/6-6										10~	10,44.	10,44.												
49	3	4	3	2	5	/6-6										10~	10,35.	49,10.												

Shift	Loc	Day	Skill	Min P	Fulfilled hours		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
50	4	0	6	1	9	/9-9								50.	24.	24.	24.	24.	24.	24.	24.	24.								
51	4	1	6	1	9	/9-9								50.	50.	50.	50.	50.	50.	50.	50.	50.								
52	4	2	6	1	9	/9-9								50.	24.	24.	24.	24.	24.	24.	24.	24.								
53	4	3	6	1	9	/9-9								31.	50.	24.	36.	24.	24.	50.	50.	36.								
54	4	4	6	1	9	/9-9								31.	24.	31.	24.	31.	24.	50.	24.	24.								

Figure 23: Shift schedule for location 2, 3 and 4, restrictive

C Zusammenfassung

Diese Arbeit stellt ein Lösungsverfahren für ein reales Tourenplanungsproblem vor. Das Problem entstand durch die Schwierigkeiten in der österreichischen Tourismusbranche. In unserer Lösung werden die Mitarbeiter flexibel unterschiedlichen Standorten zugeteilt und in verschiedene Schichten mit unterschiedlichen Qualifikationen eingeteilt. Jeglicher Transport der dadurch entsteht muss durch Shuttlebusse sichergestellt werden. Alle Dienstpläne erfüllen strikte zeitliche Auflagen bezüglich der Verfügbarkeit der Arbeitnehmer und der Erfordernisse der Arbeitgeber. Arbeitgeber legen zusätzlich ein Personenlimit für jede Schicht fest. Die Tourenplanung wird als Dial-a-Ride-Problem modelliert. Dieses beinhaltet Nebenbedingungen bezogen auf Unannehmlichkeiten die den transportierten Personen durch lange Wartezeiten entstehen, Zeitfenster und eine Beschränkung der maximalen Routendauer. Die Transportaufträge werden von den erzeugten Dienstplänen vorgegeben. Der Transport selbst erfolgt mit einer Vielzahl von homogenen Fahrzeugen die in einem gemeinsamen Depot stationiert sind. Das übergeordnete Ziel des Problems besteht darin, geeignete Dienstpläne für die Mitarbeiter zu erstellen, die eine effiziente Tourenplanung ermöglichen, während die Erfüllung der Anforderungen der Arbeitgeber sichergestellt ist.

Das Problem wird mithilfe von Metaheuristiken gelöst. Zwei unterschiedliche Planungsansätze werden gelöst und verglichen. Der restriktive Ansatz entspricht einem starren Arbeitszeitenmodell mit fixen Zeiten für den Arbeitsbeginn und das Arbeitsende. Der flexible Ansatz entspricht dem moderneren Gleitzeitmodell, dass eine individuellere Arbeitszeiteinteilung ermöglicht. Die Ergebnisse beider Ansätze werden präsentiert und gegenübergestellt.

Antonie-Alt-Gasse 12/8/11
1100 Vienna
Austria

Alina G. Dragomir

+43 680 3318896
alina-gabriela.dragomir@univie.ac.at

Personal Information

Surname	Dragomir
Forename	Alina-Gabriela
Nationality	Austria
Date of birth	7th June 1989
Place of birth	Bucharest, Romania

Education

10.2013–01.2016 (estimated)	**Master studies in Business Administration**, *University of Vienna*, Vienna, Austria.
21.08.2013	**Bachelor exam**, *University of Vienna*, Vienna, Austria.
10.2009–02.2013	**Bachelor studies in International Business Administration**, *University of Vienna*, Vienna, Austria.
04.2009	**Matura (A-levels)**, *Vienna Business School, school of economics*, Vienna, Austria. Passed with distinction
2004–2009	**School of economics**, *Vienna Business School*, 1010 Vienna, Austria.
1999–2003	**Secondary school**, *Akademisches Gymnasium Wien*, 1010 Vienna, Austria.
1995–1999	**Primary school**, *Evangelische Privatvolksschule*, 1060 Vienna, Austria.

Work experience

Since September 2015	**University of Vienna**, *Project assistent, FEAT (FWF project number P 27858 Einzelprojekte).*
02.2015–06.2015	**University of Vienna**, *Student assistant at the department of business administration, Chair for production and operations management with international focus .*
03.2013–06.2015	**Robert Bosch AG**, *In the department ECU development for BMW as a part-time intern.*
08.2012	**Robert Bosch AG**, *Internship in the department Sales Original Equipment.*
09.2009–08.2012	**Wiener Schokoladenkönig Leschanz Ges.m.b.H**, *Marginally employed salesclerk.*
07.2012	**Amway Gesellschaft m.b.H.**, *Internship in the Back Office.*
07.2010	**Gewista GmbH**, *Internship in Financial Accounting.*
07.2008	**Gewista GmbH**, *Internship in Financial Accounting.*
07.2007	**Gewista GmbH**, *Internship in Financial Accounting.*
07.2005	**Amway Gesellschaft m.b.H.**, *Internship in Customer Service.*

Student projects

02.2015–12.2015 (estimated)	**Master thesis at the chair of production and operations management with international focus**, *University of Vienna.* Working title: *Routing and scheduling of employees with different skills in multiple working locations"* Advisor: Univ.-Prof. Mag. Dr. Karl Franz Doerner
10.201–01.2015	**Term paper for the seminar Neuere Entwicklungen in eBusiness and eLogistics**, *University of Vienna.* Project title: *"Problems related to Scheduling TV Commercials"* Advisor: ao. Univ.-Prof. Dr. Christine Strauß
10.2013–01.2014	**Term paper for the seminar A in collaboration with Marina Ivanova**, *University of Vienna.* Project title: *"Order Sharing through Joint Route Planning in Carrier Collaboration: Two-phase Heuristic Algorithms for Full Truckloads Multi-depot Capacitated Vehicle Routing Problem in Carrier Collaboration by Liu et al. (2010)"* Advisor: o.Univ.Prof. Dr. Richard F. Hartl
10.2013–01.2014	**Student project work in metaheuristics for transportation logistics (TL)**, *University of Vienna.* Project title: *"Implementation of a Metaheuristic for a Capacitated Vehicle Routing Problem"* Advisor: Mag. Stefanie Kritzinger, PhD
12.2012–03.2014	**Bachelor thesis as part of International Strategy & Organization**, *University of Vienna.* Thesis title: *"Evolution of Corporate Social Responsibility and Codes of Ethics"* Advisor: MMag. Christina Keinert-Kisin
06.2012–12.2012	**Bachelor thesis as part of Risk and Insurance 1**, *University of Vienna.* Thesis title: *"An Overview of Natural Disasters and the Risk They Pose"* Advisor: Ao. Univ.-Prof. Dr. Jörg Borrmann

Languages

German	Native
Romanian	Mother tongue
English	Proficient
French	Basic

Computer skills

Programming	C++ (basic), Java (advanced)
General	Windows XP, Windows 7, Lyx, Office, SAP, LaTeX