Erasmus School of Economics

Master Thesis

MSc Econometrics and Management Science

Operations Research and Quantitative Logistics

A three-phase heuristic method for the Tour Scheduling Problem

A.F. Schiereck - 581266

Supervisor:
Dr. W. van den Heuvel

Company supervisor:
J. Meijer

Second assessor:
B.T.C. van Rossum

January 19, 2022

Abstract

In this thesis a heuristic algorithm for the Tour Schedueling Problem (TSP) is developed. The TSP is the problem of creating a schedule of shifts for employees of some company or institution considering required staffing levels throughout the planning horizon as well as labour laws and contractual agreements with employees. As the TSP is a very complex, NP-hard problem, decomposing the TSP into a Shift Scheduling Problem (SSP) and Nurse Rostering Problem (NRP) is among the most popular heuristic solution methods found in literature. However, decomposition methods can perform poorly when the TSP is highly constrained, including many rules pertaining to the assignment of employees to shifts. The heuristic developed in this thesis consists of three phases, two decomposition phases and an improvement phase to enhance the schedule obtained by the decomposition method. Furthermore, a Lagrangian relaxation and subgradient optimisation method is applied to the TSP to obtain a lower bound to the TSP. The three-phase heuristic seemingly yields good schedules, the improvement phase on average achieves a 60% reduction of the objective value of the TSP. The convergence of the subgradient optimisation is poor, no meaningful lower bound to the TSP is obtained.

Contents

1 Introduction **1**

2 Literature Review **3**

3 Problem Formulation **6**

3.1 Operational Tour Scheduling Problem . . . 6

3.2 Shift Scheduling Problem . . . 8

3.3 Nurse Rostering Problem . . . 9

3.4 Performance of decomposition methods . . . 10

4 Methodology **12**

4.1 Three-phase algorithm . . . 12

4.1.1 Destroy method . . . 12

4.1.2 Repair method . . . 13

4.1.3 Overview of the algorithm . . . 14

4.2 Lagrangian relaxation and subgradient optimisation . . . 15

4.2.1 Lagrangian relaxation . . . 15

4.2.2 Symmetry breaking constraints . . . 16

4.2.3 Subgradient optimisation . . . 17

5 Results **19**

5.1 Data . . . 19

5.2 Three-phase algorithm . . . 22

5.2.1 Data set 1 . . . 22

5.2.2 Data set 2 . . . 24

5.2.3 Data set 3 . . . 26

5.2.4 Data set 4 . . . 27

5.3 Lagrangian relaxation and subgradient optimisation . . . 27

5.3.1 Symmetry breaking constraints . . . 28

5.3.2 Data set 1 . . . 29

5.3.3 Data set 2 . . . 33

6 Conclusion and discussion **34**

6.1 Three-phase algorithm . . . 34

6.2 Lagrangian relaxation and subgradient optimisation . . . 35

Bibliography **38**

A Table of computation times Subgradient Optimisation **40**

B Results for modified gradient technique and average direction strategy **41**

1 Introduction

In many industries, like hospitality and retail, demand occurs in the form of customers arriving at a store or restaurant. This demand cannot be backlogged, as service should be provided to customers without delay. Furthermore, expected daily demand patterns can fluctuate drastically, in restaurants for example, peak demand occurs around lunch and dinner time, with a quiet period in between. Labour in these industries is often shift-based with highly flexible shifts, i.e. shifts can have varying duration, starting time and break placement. Moreover, employees can often take on different roles within an organisation, for example an employee in a restaurant might be able to work as a waiter or as host. This also grants flexibility to the manager creating the schedule. This flexible workforce creates an opportunity to fulfil the expected daily demand pattern while minimizing cost. As labour costs are the largest expense in many hospitality and retail companies, employee scheduling is of paramount importance.

Schedules are usually made on a weekly or monthly basis, in general we call this the planning horizon. An efficient schedule is devised such that the number of employees present matches the staffing requirements as closely as possible at all times during the planning horizon. If too many or too few employees are scheduled to work at a certain moment, we call this over or under coverage respectively. When designing a schedule, labour laws and contractual agreements with employees need to be taken into account as well. A schedule constitutes of shifts, many rules can apply that regulate these shifts. For example, in most cases the duration of the scheduled shift needs to fall between a certain minimum and maximum. Depending on the duration of the shift, it may be required to include one or more breaks. Many more restrictions are possible. Each shift has to be assigned to an employee who is qualified to work the particular shift. Usually, each employee can be assigned to at most one shift per day. If an employee is assigned to a shift on a certain day, this is called a working day, else it is a day off. In most cases the number of working days per week must be below some maximum, sometimes days off even need to be consecutive. Another common rule for shift assignment is a minimum free time period between shifts, to ensure an employee cannot work a night shift immediately followed by a morning shift the next day. A sequence of assignments of shifts and days off for one employee for the entire planning horizon that adheres to all these rules is called a tour in literature. Furthermore, in most labour contracts, it is laid down how many hours the employee can work per week. The employee is guaranteed to be payed the contracted hours, so in case the employee is scheduled for fewer hours they are payed anyway. We refer to this as under scheduling an employee. In some cases, it is allowed that the employee is scheduled for more hours than the contracted hours, then the employee is payed a (higher) overtime rate. We refer to this as over scheduling an employee. In other cases the contracted hours are a hard maximum. An optimal schedule satisfies all these rules and minimizes the cost of under and over coverage and the cost of unscheduled contracted hours and, if applicable, overtime cost. The problem of finding such a schedule can be modelled as an integer programming (IP) model, and is often referred to as the Tour Scheduling Problem (TSP).

The TSP is shown to be NP-hard by Bartholdi (1981). For large instances with a high degree of flexibility it cannot be expected that an optimal schedule can be found in reasonable time. In practical applications, the TSP is often solved using heuristic methods. Many of these methods include decomposing the TSP into two steps. In the first steps, for each day in the planning horizon, a selection of shifts is chosen that minimizes the cost of over and under coverage. In the second step, these shifts are assigned to employees, such that the costs due to over and under scheduling are minimized. The first step is often referred to as the Shift Scheduling Problem (SSP), the second step as the Nurse Rostering Problem (NRP). Decomposing the TSP significantly reduces the problem size, though both the SSP and the NRP are still NP-hard. A drawback of this method is that it does not guarantee that an optimal schedule is found. In the TSP, shifts are selected and assigned to employees simultaneously, guaranteeing that both under and over coverage is minimized as well as over and under scheduling. The weakness of the decomposition method arises because the SSP and NRP are solved sequentially, shifts are selected solely based on the staffing requirements, not taking into account whether a favourable allocation of these shifts to employees exists. Therefore, the schedule created in this manner can have unnecessarily high costs due to (over and) under scheduling. Furthermore, it is possible that labour laws and contractual agreements prevent all shifts chosen in the first step from begin assigned. Leaving these shifts unassigned causes under coverage costs that may be (partially) preventable if shifts were chosen with the employees in mind.
In this thesis a heuristic algorithm is developed to solve the TSP that includes decomposition into SSP and NRP as well as an improvement step to avoid unnecessarily high costs due to unscheduled contracted hours and under coverage.

2 Literature Review

In this section, a concise overview of previous research on the TSP is given.

Many different versions and solution approaches for labour scheduling problems are researched in literature. A review of literature in the field of staff scheduling is given by Ernst et al. (2004). In this paper, the process of staff scheduling is broken down into modules, and papers are classified based upon which modules they address. The modules considered are: demand modelling, days-off scheduling, shift scheduling, line of work or tour construction, task assignment and staff assignment. The modules demand modelling and task assignment are left outside of the scope of this thesis. The TSP always includes integrated days-off scheduling, shift scheduling and line of work or tour construction. In some cases, it does not include staff assignment. Then, the solution to the TSP represents decisions on a strategic level for managing the size of the workforce. In other cases, the workforce composition and size is fixed, and staff assignment is included. Then, the solution to the TSP represents decisions on an operational planning level. In this thesis, we introduce terminology to distinguish between these cases. When staff assignment is not included we call this the strategic TSP. When staff assignment is included we call this the operational TSP.

The TSP was first formulated by Dantzig (1954) as a set covering IP model. For this formulation a set of all feasible tours is generated. An integer decision variable per tour is created to indicate how many employees are assigned to the tour. Formulations where a set of all feasible tours is generated are referred to as explicit formulations in literature, because all feasible tours are considered explicitly. This formulation represents the strategic TSP, though it can be translated to a formulation of the operational TSP by creating a binary decision variable per employee-tour pair, indicating whether the employee is assigned to the tour. In situations with highly flexible shifts and a flexible workforce, the number of feasible tours is extremely large.

Easton and Rossin (1991) consider the strategic explicit formulation of the TSP from Dantzig (1954). To deal with the large number of feasible tours Easton and Rossin (1991) consider only a working subset of all feasible tours. This working subset is composed by taking all active tours in solution to the LP-relaxation of the TSP. The LP-relaxation of the TSP is solved using column generation.

In Restrepo et al. (2016) two formulations for the operational TSP are proposed: a tour-based formulation and a daily-based formulation. The tour-based formulation is similar to the explicit formulation of Dantzig (1954). For the daily-based formulation a set of all feasible shifts is generated for each day. Binary variables for each employee-shift pair indicate whether the shift is assigned to the employee. Rules pertaining to days off and sequences of shifts on different days are added to the model as constraints. In literature, this is called an implicit formulation, because tours are not considered explicitly, but are implicitly defined by constraints. Both formulations are solved with a branch-and-price method.

Topaloglu and Ozkarahan (2004) use an implicit goal programming formulation to solve the operational TSP. In this model, additional goals are included to satisfy employee preference for working certain shifts or having certain days off. The model is solved as an IP model with a commercial solver for small data sets.

A review of tour scheduling literature is given by Alfares (2004). In this review, papers are categorised based on the solution methods proposed. According to this paper, decomposition methods are among the most popular methods to solve the TSP. In most of the articles proposing a decomposition method a solution to the TSP is found by first solving the SSP and then the NRP, though some articles first solve a days-off scheduling problem and then a daily shift scheduling problem. Decomposing the TSP into two sub-problems decreases the problem size and complexity.

In Buffa et al. (1976) a three-phase scheduling system for operators at telephone company is proposed. In the first phase, demand is forecasted and converted to operator requirements for each half-hour interval of the day. Based on these requirements, shifts are selected by a heuristic algorithm presented by Luce (1973). Thereafter, shifts are assigned to employees by an algorithm developed by Luce (1974).

Jarrah et al. (1994) also propose a decomposition method to solve the strategic TSP. First a personnel scheduling model is formulated as an IP problem. With this model, shifts are selected. From these shifts, tours are constructed as a post-processing step.

Brusco et al. (1995) proposes a three-phase algorithm for solving the strategic TSP. In the first phase, shifts are generated by a heuristic. In the second phase an IP is formulated to assign shifts to tours. In the third phase, the solution is improved by a local search heuristic using simulated annealing. Simulated annealing is a local search method where deterioration of the candidate solution is sometimes allowed, with the aim of facilitating diversification. Adding the third improvement phase significantly improves the schedule obtained by the first two decomposition phases.

Turhan and Bilgen (2020) proposes a hybrid improvement algorithm algorithm for the NRP. The improvement algorithm includes simulated annealing and fix-&-optimize. In the fix-&-optimize step, the schedule is fixed for low-cost days. For the remaining days, the schedule is re-optimized.

In other application areas of scheduling problems, such as the scheduling of jobs in manufacturing industries or the scheduling of supply or demand of energy, decomposition methods using Lagrangian relaxation are popular. Lagrangian relaxation involves relaxing one or more constraint and adding them to the objective function. Ghaddar et al. (2015) uses Lagrangian relaxation to decompose a water pump scheduling problem, Tang et al. (2006) propose a Lagrangian relaxation algorithm for hybrid flow shop scheduling. When a problem is decomposed with Lagrangian relaxation, most commonly subradient optimisation is used to solve the Lagrangian dual problem to obtain a lower bound on the original minimisation problem. Subgradient optimisation is iterative method to find the maximum of a non-differentiable concave function. Starting from an initial point, at each iteration a step is taken in the direction of a subgradient of the function. Held et al. (1974) present computational experiments that show the potential effectiveness of the subgradient method. However, a vulnerability of the subgradient method can be encountered for piecewise linear functions. Different regions of the domain are differentiable and have gradients. The gradients of two neighbouring regions may form an obtuse angle. In this case, moving along the direction of the subgradient can form a a zig-zagging path towards the optimal solution, slowing down convergence. Adjustments to the subgradient method have been proposed to remedy this. Camerini et al. (1975) propose a modified subgradient method and Sherali and Ulular (1989) propose the average direction strategy. Both methods include determining the step direction based on the subgradient and the step direction of the previous iteration. In literature, these methods are referred to as deflected subgradient methods. In context where different subgradient methods are used, the term pure subgradient method is used to indicate subgradient optimisation where the step direction is precisely equal to the subgradient. Rong et al. (2008) applies Lagangian relaxation and a deflected subgradient optimisation method to a production planning problem for a trigeneration power plant. Lagrangian relaxation and subgradient optimisation have been applied to many different problems, however to the best of the authors' knowledge, no articles applying such methods to the TSP have been presented in literature.

The subgradient optimisation method involves repeatedly solving a relaxed version of the TSP. The solution space of the TSP is often symmetric, because in many cases all shift assignments for two employees can be swapped, resulting in a new schedule that is not substantially different in a practical sense. To reduce this symmetry, lexicographical ordering constraints can be added to the TSP formulation. The working of such constraints is described by Ostrowski et al. (2010).

In this thesis, we propose an decomposition based algorithm for the operational TSP with an improvement phase that is inspired by a fix-&-optimize heuristic. Furthermore we aim to obtain a lower bound the the TSP using Lagrangian relaxation and various subgradient methods.

3 Problem Formulation

In this section we give a formulation of the operational TSP, SSP and NRP and show how the problems are related. Finally a some observations about the performance of the decomposition method for the TSP are given.

3.1 Operational Tour Scheduling Problem

In this section the notation and formulation of the operational TSP are given. We consider a planning horizon, typically spanning one week. The planning horizon is divided into time intervals i, typically of one hour or thirty or fifteen minutes. The set of all time intervals i spanning the entire planning horizon is denoted by $\mathcal{I}$. The set $\mathcal{S}$ contains all feasible shifts s. A shift s is uniquely defined by its starting time and date and duration and, if applicable, its break placements. Any rules that apply directly to shifts, such as minimum or maximum duration or possible breaks, can be dealt with when generating this set. The set of all employees e is denoted by $\mathcal{E}$. For each employee, the set $\mathcal{S}_e$ contains all shifts that can be assigned to employee e. In principle, this set is equal to the set of all shifts $\mathcal{S}$, but rules can apply that prohibit employees from being assigned to certain shifts, e.g. the contract of an employee states that they will only work on the weekend. The staffing requirement for each time interval i is given by r_i, the cost of under staffing during interval i by c_i^- and the cost of over staffing by c_i^+. Binary parameter ϕ_{si} indicates whether time interval i is covered by shift s. The duration of shift s in working hours is given by d_s. For employee e, their number of contracted hours in the planning horizon is denoted by h_e, the cost of over scheduling by k_e^+ and the cost of under scheduling by k_e^-. Decision variable x_s indicates how many times shift s is chosen to be included in the schedule, and slack variables x_i^- and x_i^+ indicate the under and over coverage during time interval i respectively. Decision variable y_{se} indicates whether shift s is assigned to employee e, and slack variables y_e^- and y_e^+ indicate the number of under and over scheduled hours for employee e respectively. The notation for the TSP is listed in Table 1.

Table 1: Notation for the MIP Formulation of the operational Tour Scheduling Problem

Symbol	Definition
<u>Sets</u>	
$\mathcal{I}$	Set of all time intervals i
$\mathcal{S}$	Set of all shifts s
$\mathcal{E}$	Set of all employees e
$\mathcal{S}_e$	Set of all shifts s that can be assigned to employee e
<u>Parameters</u>	
$r_i \in \mathbb{R}_{\geq 0}$	Staffing requirement during time interval $i \in \mathcal{I}$
$c_i^- \in \mathbb{R}_{\geq 0}$	Cost of under coverage during time interval $i \in \mathcal{I}$
$c_i^+ \in \mathbb{R}_{\geq 0}$	Cost of over coverage during time interval $i \in \mathcal{I}$
$\phi_{si} \in \mathbb{B}$	Indicates whether time interval $i \in \mathcal{I}$ is is covered by shift $s \in \mathcal{S}$
$h_e \in \mathbb{R}_{\geq 0}$	Number of contracted hours in the planning horizon for employee $e \in \mathcal{E}$
$d_s \in \mathbb{R}_{\geq 0}$	Duration of shift $s \in \mathcal{S}$ in working hours
$k_e^- \in \mathbb{R}_{\geq 0}$	Cost of unscheduled contracted hours for employee $e \in \mathcal{E}$
$k_e^+ \in \mathbb{R}_{\geq 0}$	Cost of overtime for employee $e \in \mathcal{E}$
<u>Decision Variables</u>	
$x_s \in \mathbb{N}$	Number of times shift $s \in \mathcal{S}$ is chosen
$x_i^- \in \mathbb{R}_{\geq 0}$	Under coverage in time interval $i \in \mathcal{I}$
$x_i^+ \in \mathbb{R}_{\geq 0}$	Over coverage in time interval $i \in \mathcal{I}$
$y_{se} \in \mathbb{B}$	Indicates whether shift $s \in \mathcal{S}$ is assigned to employee $e \in \mathcal{E}$
$y_e^- \in \mathbb{R}_{\geq 0}$	Number of unscheduled contracted hours for employee $e \in \mathcal{E}$
$y_e^+ \in \mathbb{R}_{\geq 0}$	Number of overtime hours for employee $e \in \mathcal{E}$

The TSP is formulated as follows:

$$z^*_{TSP} = \min \sum_{i\in\mathcal{I}} c_i^- x_i^- + c_i^+ x_i^+ + \sum_{e\in\mathcal{E}} k_e^- y_e^- + k_e^+ y_e^+, \tag{1}$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{S}} \phi_{si} x_s + x_i^- - x_i^+ = r_i \qquad \forall\, i \in \mathcal{I}, \tag{2}$$

$$\sum_{e\in\mathcal{E}} y_{se} = x_s \qquad \forall\, s \in \mathcal{S}, \tag{3}$$

$$\sum_{s\in\mathcal{S}} y_{se} d_s + y_e^- - y_e^+ = h_e \qquad \forall\, e \in \mathcal{E}, \tag{4}$$

$$y_{se} = 0 \qquad \forall\, e \in \mathcal{E},\ \forall\, s \notin \mathcal{S}_e \tag{5}$$

$$x_s \in \mathbb{N} \qquad \forall\, s \in \mathcal{S}, \tag{6}$$

$$x_i^-,\ x_i^+ \in \mathbb{R}_{\geq 0} \qquad \forall\, i \in \mathcal{I}, \tag{7}$$

$$y_{se} \in \mathbb{B} \qquad \forall\, s \in \mathcal{S},\ e \in \mathcal{E}, \tag{8}$$

$$y_e^-,\ y_e^+ \in \mathbb{R}_{\geq 0} \qquad \forall\, e \in \mathcal{E}. \tag{9}$$

Constraint (2) determines the under and over coverage based on the requirements and the chosen shifts for each time interval. Constraint (3) dictates that all shifts that are chosen to be included in the schedule must be assigned to an employee. Constraint (4) determines the under and over scheduling for each employee, based on the shift assignments and contracted hours. Constraint (5) makes sure a shift is not assigned to an employee that is ineligible. Constraints (6) - (9) are included as integrality, binary and non-negativity constraints. Finally, the objective (1) minimizes the cost due to over and under coverage, and over and under scheduling.

The formulation for the TSP is very basic, in practical applications some adjustments or extensions might be needed. For example, in some cases, an employment contract does not guarantee fixed contracted hours but rather specifies that the worked hours in the planning horizon falls in a range between some minimum and maximum, denoted by h_e^{min} and h_e^{max}. In this case Constraint (4) is replaced by the following two constraints:

$$\sum_{s\in\mathcal{S}} y_{se} d_s + y_e^- \geq h_e^{min} \qquad \forall\, e \in \mathcal{E}, \tag{10}$$

$$\sum_{s\in\mathcal{S}} y_{se} d_s - y_e^+ \leq h_e^{max} \qquad \forall\, e \in \mathcal{E} \tag{11}$$

Furthermore, any rules that apply to the assignment of shifts to employees, such as maximum number of working days per week or minimum number of consecutive days off, can be added as constraints to this formulation.

3.2 Shift Scheduling Problem

In this section the formulation of the SSP is given. The formulation of the SSP is closely related to the formulation of the TSP given in section 3.1. The SSP consists of all variables and constraints of the TSP related to over and under coverage. The formulation of the SSP is given by:

$$z^*_{SSP} = \min \sum_{i \in \mathcal{I}} c_i^- x_i^- + c_i^+ x_i^+, \tag{12}$$

$$\text{s.t.} \quad \sum_{s \in \mathcal{S}} \phi_{si} x_s + x_i^- - x_i^+ = r_i \qquad \forall\, i \in \mathcal{I}, \tag{13}$$

$$x_s \in \mathbb{N} \qquad \forall\, s \in \mathcal{S}, \tag{14}$$

$$x_i^-,\ x_i^+ \in \mathbb{R}_{\geq 0} \qquad \forall\, i \in \mathcal{I}. \tag{15}$$

Constraint (13), Constraint (14) and Constraint (15) are equal to Constraint (2), Constraint (6) and Constraint (7) of the TSP. The objective of the SSP is equal to the first term of the objective of the TSP.

3.3 Nurse Rostering Problem

In this section the formulation of the NRP is given. The formulation of the NRP is closely related to the formulation of the TSP given in section 3.1, with some adjustments.

The the decision variables x_s of the TSP indicating how many times each shift s is chosen to appear in the schedule are used as parameters in the NRP. The NRP consists of the decision variables and constraints of the TSP pertaining to under and over scheduling. A basic version NRP is formulated as follows:

$$z^*_{NRP} = \min \sum_{e \in \mathcal{E}} k_e^- y_e^- + k_e^+ y_e^+ + \sum_{s \in \mathcal{S}} p_s^- z_s^-, \tag{16}$$

$$\text{s.t.} \quad \sum_{e \in \mathcal{E}} y_{se} + z_s^- = x_s \qquad \forall\, s \in \mathcal{S}, \tag{17}$$

$$\sum_{s \in \mathcal{S}} y_{se} d_s + y_e^- - y_e^+ = h_e \qquad \forall\, e \in \mathcal{E}, \tag{18}$$

$$y_{se} = 0 \qquad \forall\, e \in \mathcal{E},\ \forall\, s \notin \mathcal{S}_e \tag{19}$$

$$y_{se} \in \mathbb{B} \qquad \forall\, s \in \mathcal{S},\ e \in \mathcal{E}, \tag{20}$$

$$y_e^-,\ y_e^+ \in \mathbb{R}_{\geq 0} \qquad \forall\, e \in \mathcal{E}, \tag{21}$$

$$z_s^- \in \mathbb{R}_{\geq 0} \qquad \forall\, s \in \mathcal{S}. \tag{22}$$

This formulation can be extended and adjusted to include rules on the assignment of shifts or to replace fixed contracted hours h_e by a range of allowed working hours per planning period. In the latter case Constraint (18) is replaced by Constraints (10) and (11).

Comparing the formulations of the TSP and the NRP, it can be noted that Constraint (18), Constraint (19), Constraint (20) and Constraint (21) are equal to Constraint (4), Constraint (5), Constraint (8) and Constraint (9). Constraint (17) is similar to Constraint (3) dictating that each shift that is chosen must be assigned to an employee. However, in the NRP this is a soft constraint, with a slack variable z_s^- indicating how many of the chosen shifts s are left unassigned. Constraint (22) is added to dictate non-negativity for the slack variable z_s^-. These slack variables are added to the objective with a penalty cost p_s^-. The NRP is formulated with a soft constraint to make the formulation more generally applicable. When rules pertaining to the assignment of shifts (such as a minimum rest time between shifts, a maximum working days per week, etcetera) are added to the formulation of the NRP, it may not be possible to assign all shifts. To prevent the problem becoming infeasible and still obtain a meaningful solution, we allow shifts to be left unassigned, albeit with a high penalty. Furthermore, x_s are decision variables in the TSP but parameters in the NRP. The objective of the NRP contains the final two terms of the objective of the TSP, and the aforementioned penalty term for leaving shifts unassigned.

3.4 Performance of decomposition methods

When the operational TSP is solved with a decomposition method of first choosing shifts, then assigning those shifts to employees, the resulting schedule is generally not optimal. This is because shifts are chosen solely based on staffing requirements, whether a favourable allocation of these shifts to the employees exists is not taken into account. The resulting schedule will have low cost due to over coverage, but may have unnecessarily high costs due to under coverage, and under and over scheduling employees. When the NRP is highly constrained, i.e. many rules apply to the assignment of shifts to employees, it may not be possible to assign all shifts and under coverage can arise when shifts are left unassigned. Consider the following simple example of a decomposition method yielding a sub-optimal schedule due to the constraint that overtime is not allowed: let there be a business that is open for 8 hours, three days a week. Let the business have two employees who are contracted to work for 12 hours per week each. Solving the SSP might yield three 8-hour shifts for each of the days that the business is open. When the NRP is solved subsequently, both employees will each be assigned to one of the three 8-hour shifts. The last 8-hour shift cannot be assigned to either employee, because it would exceed their contracted hours. The last shift will be left unassigned. It is easy to see a better schedule can be obtained when the third 8-hour shift is swapped for two 4-hour shifts, but to see this the shift selection as well as the assignment of shifts to employees needs to be considered simultaneously.

Furthermore, a sub-optimal schedule can be obtained when the total staffing requirements are unequal to the total contracted hours of all employees. How this yields sub-optimal schedules can be further investigated by analizing the different costs: over and under coverage costs c^+ and c^- and over and under scheduling costs k^+ and k^-.

In most practical situations, the cost of under coverage is the highest cost. If a business is under covered, not enough employees are available to assist customers and the business misses out on revenue. The cost for over scheduling (if applicable) is generally lower than the cost for under coverage, if this were not the case, over scheduling would never be profitable and would thus never occur. In a situation where a manager must choose between under coverage or over scheduling, it is natural to prioritize avoiding under coverage. Thus it can be expected that creating a schedule by first deciding on shifts, then on the assignment to employees will yield a reasonable schedule in this case.

The cost of over coverage is approximately equal to the cost of wages in most practical situations. Having more employees present than required is generally not detrimental to business, it is just wasteful. The cost of under scheduling an employee is also approximately equal to the cost of wages, as an employee is payed wages for the contracted hours, regardless if he or she is scheduled to work. However, a manager will generally prefer to schedule an employee for all contracted hours, even if this causes over coverage. There could always be some odd jobs, or it could be unexpectedly busy. In a situation where a manger must choose between over coverage or under scheduling an employee, it is not natural to prioritize avoiding over coverage. In this case, it can be expected that creating a schedule by first deciding on shifts, then on the assignment to employees will yield an unfavourable schedule.

In conclusion, the weakness of decomposition methods for the operational TSP mainly manifests itself when the assignment of shifts to employees is highly constrained or when the size of the workforce is slightly bigger than strictly necessary. In this thesis, a heuristic three-phase algorithm is developed to find solutions to the TSP. The first two phases are decomposition phases are the two decomposition phases, i.e. the SSP and NRP are solved sequentially. The third phase of the algorithm is an improvement phase that is designed to overcome these shortcomings of the decomposition method. In section 4 the three-phased algorithm is discussed in more detail.

4 Methodology

In this section the three-phase algorithm we develop to solve the operational TSP is described. Furthermore, the Lagrangian relaxation subgradient optimisation method intended to provided a lower bound is presented.

4.1 Three-phase algorithm

In this section, a three-phase method for solving the operational TSP is described. The first two phases are the decomposition phases. In the first phase, shifts are chosen that make up the schedule by solving the SSP. In the second phase, those shifts are assigned to employees by solving the NRP. The result of these two phases is taken as the initial schedule. The third phase is the improvement phase, where the initial schedule is improved by a Large Neighbourhood Search (LNS) improvement heuristic. A LNS heuristic is chosen because the operational TSP is usually a highly constrained problem. The formulation for the operational TSP given in this thesis is a basic formulation with few constraints, but in most practical situations much more constraints apply. For example there may a maximum number of working days per week, days off may need to be consecutive, etcetera. We aim to develop a method that is easily extendable with additional constraints like these. For highly constrained problems, neighbourhoods of a feasible solution may contain many infeasible solutions. Therefore, local search methods where entire neighbourhoods are searched, may be ineffective. The LNS is an iterative method where in each iteration the schedule is repeatedly partly destroyed and repaired, so that a feasible solution is obtained. In the following sections, the destroy and repair methods are presented. Finally an overview of the three-phase algorithm is given.

4.1.1 Destroy method

The destroy method used in this algorithm is selected based on the reflections on the performance of decomposition methods presented in section 3.4. As discussed, decomposition methods can be expected to yield poor schedules when the NRP is highly constrained, when the workforce is larger than necessary or both. In the first case, shifts may be left unassigned when constrains prevent a feasible allocation of all shifts to employees. The schedule obtained by a decomposition method may have under scheduled employees for whom the available shifts to not fit in their schedule, as well as under coverage due to shifts being left unassigned. In the second case, all shifts may be assigned but some employees may still be under scheduled. The schedule obtained may have under scheduled employees and no or little under coverage. In both cases, the schedule can be improved by adjusting the schedule for the employees who are under scheduled. Therefore, the schedule is destroyed in each iteration by removing all shifts from the schedule for the employee who is the most under scheduled. This yields a destroyed schedule which likely has some under coverage due to shifts being removed, possibly in addition to some preexisting under coverage. Moreover it has one employee to whom no shifts are assigned.

4.1.2 Repair method

Subsequently the schedule is repaired by a repair method. This method is designed based on the reflections of the performance of decomposition methods as presented in section 3.4 and on the destroy method presented in section 4.1.1. As discussed in section 3.4 the destroyed schedule can be expected to include some periods of under coverage and one employee to whom no shifts are assigned. The schedule is repaired in two steps. In the first step a set of candidate shifts is compiled. The second step is inspired by a fix-&-optimize heuristic, in this step an adjusted version of the NRP is solved to assign shifts from the set of candidate shifts to the employee without shifts. The shift assignments for all other employees are fixed. In the next paragraphs the procedure for compiling the candidate shift set is described and the fix-&-optimize inspired step is discussed in greater detail.

Candidate shift set To repair the schedule a set of suitable candidate shifts is selected from the set of all possible shifts. All possible shifts s are given a score, denoted by α_s. The score of a shift is based on the reduction of under coverage that is attained when the shift is added to the destroyed schedule. A shift that reduced the under coverage by one hour will obtain a score of -1. In this context, a lower (more negative) score means a shift resolves more under coverage and is thus considered 'better'. This way, shifts that precisely cover all periods of under coverage on a certain day are awarded the best score, as well as shifts that cover and extend beyond the periods of under coverage. Furthermore, shifts that cover most of the periods of under coverage will be awarded a good score. The n shifts with the best score are added to the set of candidate shifts. The size of the candidate shift set n is chosen based on the total number of possible shifts. The aim of this scoring method is to provide for the fix-&optimize inspired step a set of shifts of varying length that cover all or most of the under coverage of the destroyed schedule. This allows in the fix-&-optimize inspired step the selecting of shifts that add up precisely to the contracted hours of the employee.

Fix-&-optimize inspired step In the fix-&-optimize inspired step, the schedule is repaired by solving an adjusted version of the NRP. The schedule is fixed for all employees except the employee who was selected in the destroy method. The formulation of the adjusted version of the NRP is given:

$$\min \sum_{s\in\mathcal{S}_c} \alpha_s \sum_{e\in\mathcal{E}} y_{se} + \sum_{e\in\mathcal{E}} k_e^- y_e^- + k_e^+ y_e^+, \tag{23}$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{S}_c} y_{se} d_s + y_e^- - y_e^+ = h_e \qquad \forall\ e \in \mathcal{E}, \tag{24}$$

$$y_{se} = 0 \qquad \forall\ e \in \mathcal{E},\ \forall\ s \notin \mathcal{S}_e, \tag{25}$$

$$y_{se} \in \mathbb{B} \qquad \forall\ s \in \mathcal{S}_c,\ e \in \mathcal{E}, \tag{26}$$

$$y_e^-,\ y_e^+ \in \mathbb{R}_{\geq 0} \qquad \forall\ e \in \mathcal{E}, \tag{27}$$

Here the candidate shift set is denoted by $\mathcal{S}_c$. This formulation of the adjusted NRP is similar to the formulation given in 3.3, Constraints (24), (25), (26), (27) are almost equal to Constraints (18), (19), (20), (21). In the adjusted version, Constraint (17) is dropped, and correspondingly the penalty term in the objective for leaving shifts unassigned. This adjustment is made because the candidate shift set is intended to provide a variety of shift options, not a set of shifts that should all be assigned. Furthermore, the shift score that is determined when compiling the candidate shift set is added to the objective. Note that, since the NRP is a minimization problem, shifts with lower scores are more likely part of an optimal solution. As discussed in the previous paragraph, the 'best' shifts are given the lowest scores.

4.1.3 Overview of the algorithm

The algorithm starts with the SSP-phase, then the NRP-phase to obtain an initial solution. Then the iterative improvement phase starts. In each iteration the schedule is destroyed by removing all shifts assigned to the most under scheduled employee. If there are no more under scheduled employees, the algorithm is terminated. Subsequently the schedule is repaired and a candidate solution is obtained. The candidate solution is accepted in favour of the incumbent solution if the under coverage has reduced or remained the same and the under scheduling has reduced or remained the same. Finally, at the end of each iteration the employee for whom the schedule was destroyed is added to a tabu-list to prevent the algorithm from repeating the same iteration. Whenever the new candidate solution is accepted, the tabu-list is cleared. An overview of the three-phase algorithm is given in Table 1.

Algorithm 1: Heuristic three-phase algorithm for TSP with LNS

```
Create a schedule x with unassigned shifts by solving the SSP;
Assign the shifts in schedule x to employees by solving the NRP;
Set best know solution x* ← x;
while stopping criterion not met do
    Select the most under scheduled employee that is not on the tabu-list ê;
    Remove all shifts assigned to employee ê for the schedule to obtain destroyed schedule x^d;
    Compile a candidate shift set by assigning a score to each possible shift s;
    Solve the adjusted version of the NRP while fixing the shift assignments for all employees
      except ê to obtain candidate solution x';
    if acceptance criterion for candidate x' with incumbent solution x* is met then
        Set x* ← x';
        Clear tabu-list
    end
    Add employee ê to tabu-list
end
Return x*
```

4.2 Lagrangian relaxation and subgradient optimisation

In this section, a Lagrangian relaxation and subgradient optimisation procedure is described which is aimed at providing a lower bound for the operational TSP.

4.2.1 Lagrangian relaxation

One set of constraints of the operational TSP is relaxed and incorporated in the objective. The set of constraints that is relaxed are Constraints (3), which dictate that each shift that is chosen to be included in the schedule should be assigned to an employee. Such a constraint exists for each shift $s \in \mathcal{S}$. Each of these constraints is relaxed and incorporated into the objective, multiplied by Lagrangian multiplier $\lambda_s \in \mathbb{R}$. The Lagrangian multiplier is chosen to be a real number, because the relaxed constraints are equality constraints. $\boldsymbol{\lambda} \in \mathbb{R}^m$ denotes the vector of all Lagrangian multipliers, where m is the number of shifts, i.e. the cardinalty of set $\mathcal{S}$. The relaxed TSP (RTSP) yields the following Lagrangian function:

$$w(\boldsymbol{\lambda}) = z^*_{RTSP} = \min \sum_{i \in \mathcal{I}} c_i^- x_i^- + c_i^+ x_i^+ + \sum_{e \in \mathcal{E}} k_e^- y_e^- + k_e^+ y_e^+ + \sum_{s \in \mathcal{S}} \lambda_s \Big(x_s - \sum_{e \in \mathcal{E}} y_{se} \Big), \tag{28}$$

$$\text{s.t.} \quad \sum_{s \in \mathcal{S}} \phi_{si} x_s + x_i^- - x_i^+ = r_i \qquad \forall\, i \in \mathcal{I}, \tag{29}$$

$$\sum_{s \in \mathcal{S}} y_{se} d_s + y_e^- - y_e^+ = h_e \qquad \forall\, e \in \mathcal{E}, \tag{30}$$

$$y_{se} = 0 \qquad \forall\, e \in \mathcal{E},\ \forall\, s \notin \mathcal{S}_e, \tag{31}$$

$$x_s \in \mathbb{N} \qquad \forall\, s \in \mathcal{S}, \tag{32}$$

$$x_i^-,\ x_i^+ \in \mathbb{R}_{\geq 0} \qquad \forall\, i \in \mathcal{I}, \tag{33}$$

$$y_{se} \in \mathbb{B} \qquad \forall\, s \in \mathcal{S},\ e \in \mathcal{E}, \tag{34}$$

$$y_e^-,\ y_e^+ \in \mathbb{R}_{\geq 0} \qquad \forall\, e \in \mathcal{E}. \tag{35}$$

In this relaxed formulation, each constraint only contains x-variables (related to choosing shifts) or y-variables (related to assigning shifts), no constraint contains both. Therefore the RTSP decomposes into the following two subproblems resembling the SSP and the NRP.

$$w(\boldsymbol{\lambda}) = z^*_{RTSP} = \min \sum_{i\in\mathcal{I}} c_i^- x_i^- + c_i^+ x_i^+ + \sum_{s\in\mathcal{S}} \lambda_s x_s, \tag{36}$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{S}} \phi_{si} x_s + x_i^- - x_i^+ = r_i \qquad \forall\, i \in \mathcal{I}, \tag{37}$$

$$x_s \in \mathbb{N} \qquad \forall\, s \in \mathcal{S}, \tag{38}$$

$$x_i^-,\ x_i^+ \in \mathbb{R}_{\geq 0} \qquad \forall\, i \in \mathcal{I}. \tag{39}$$

$$+ \min \sum_{e\in\mathcal{E}} k_e^- y_e^- + k_e^+ y_e^+ - \sum_{s\in\mathcal{S}} \lambda_s \sum_{e\in\mathcal{E}} y_{se}, \tag{40}$$

$$\text{s.t.} \quad \sum_{s\in\mathcal{S}} y_{se} d_s + y_e^- - y_e^+ = h_e \qquad \forall\, e \in \mathcal{E}, \tag{41}$$

$$y_{se} = 0 \qquad \forall\, e \in \mathcal{E},\ \forall\, s \notin \mathcal{S}_e, \tag{42}$$

$$y_{se} \in \mathbb{B} \qquad \forall\, s \in \mathcal{S},\ e \in \mathcal{E}, \tag{43}$$

$$y_e^-,\ y_e^+ \in \mathbb{R}_{\geq 0} \qquad \forall\, e \in \mathcal{E}. \tag{44}$$

The value of the Lagrangian function is a lower bound to the objective value of the TSP, i.e. $w(\boldsymbol{\lambda}) \leq z^*_{TSP}\ \forall \boldsymbol{\lambda} \in \mathbb{R}^m$, where m is the number of shifts. A proof is sketched here. Let $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})$ denote the optimal solution to the RTSP for some $\tilde{\boldsymbol{\lambda}}$, such that $z_{RTSP}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) = z^*_{RTSP}$ and let $\mathcal{F}_{RTSP}$ denote the feasible region of the RTSP. Let $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ denote an optimal solution to the TSP, such that $z_{TSP}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) = z^*_{TSP}$ and let $\mathcal{F}_{TSP}$ denote the feasible region of the TSP. Since the RTSP is obtained by relaxing the TSP, $\mathcal{F}_{TSP} \subseteq \mathcal{F}_{RTSP}$. Furthermore, for any solution $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{F}_{TSP}$ we have $z_{TSP}(\boldsymbol{x}, \boldsymbol{y}) = z_{RTSP}(\boldsymbol{x}, \boldsymbol{y})$ because the Lagrangian term in the objective of the RTSP equals 0 for a solution $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{F}_{TSP}$. If the optimal solution to the RTSP $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \in \mathcal{F}_{TSP}$, then $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) = (\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ and $w(\tilde{\boldsymbol{\lambda}}) = z^*_{RSTP} = z^*_{TSP}$. If the optimal solution to the RTSP $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \notin \mathcal{F}_{TSP}$, then $w(\tilde{\boldsymbol{\lambda}}) = z^*_{RTSP} = z_{RTSP}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \leq z_{RTSP}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) = z_{TSP}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) = z^*_{TSP}$.

4.2.2 Symmetry breaking constraints

In the subgradient optimisation procedure, the RTSP is solved repeatedly. With the aim of reducing computation time, lexicographical ordering constraints are added to the RTSP to eliminate symmetric solutions. In order for these constraints to be added, ordered sets of 'identical' employees are formulated. In this context, employees e_1 and e_2 are considered to be 'identical' if their contract hours are the same and their sets of feasible shifts $\mathcal{S}_{e_1}$ and $\mathcal{S}_{e_2}$ are the same. For each cluster of identical employees that can be found, an ordered set $\mathcal{E}^{\text{identical}}$ is formulated. Such a set always contains 2 or more employees. The following notation is introduced: for each employee $e \in \mathcal{E}^{\text{identical}}$ except the first employee in the order, the employee that comes before e in the order is denoted by ϵ_e.

Furthermore, the set of shifts $\mathcal{S}$ is ordered, and the following notation is introduced: for each shift $s \in \mathcal{S}$ we introduce the set $\mathcal{S}_s^{\text{prior}}$ that contains shift s and all shifts that come before s in the order.

In case one or more sets of identical employees $\mathcal{E}^{\text{identical}}$ exist the following constraints are added to the formulation of the RTSP for each ordered set of identical employees:

$$y_{s,e} \leq \sum_{\sigma \in \mathcal{S}_s^{\text{prior}}} y_{\sigma,\epsilon_e} \qquad \forall s,\ \forall e \in \mathcal{E}^{\text{identical}} \setminus \{\text{first employee of set } \mathcal{E}^{\text{identical}}\}. \tag{45}$$

This constraint ensures that an employee $e \in \mathcal{E}^{\text{identical}}$ can only be assigned to shift s if the employee ϵ that comes before e in the order of $\mathcal{E}^{\text{identical}}$ is assigned to at least one prior shift $\sigma \in \mathcal{S}_s^{\text{prior}}$. For any two equivalent solutions that are equal except that the shift assignment for two identical employees has been swapped, one of the two solutions is eliminated by these constrains.

4.2.3 Subgradient optimisation

The best lower bound to the TSP is obtained by solving the following maximisation problem by subgradient optimisation: $w^*(\boldsymbol{\lambda}) = \max\{w(\boldsymbol{\lambda}) : \boldsymbol{\lambda} \in \mathbb{R}^m\}$. In the next paragraph, the procedure of the pure subgradient method used to solve this problem is outlined. In the following paragraph, the adjustments made to this procedure in deflected subgradient methods is outlined.

Pure subgradient optimisation The following iterative steps describe the pure subgradient optimisation method.

Step 1: Initialize $k = 0$, $\boldsymbol{\lambda}^{(0)} \in \mathbb{R}^m$.

Step 2: Define step size sequence $\{\mu_k\}_{k=0}^{\infty} = \rho_k \frac{UB - w(\boldsymbol{\lambda}^{(k)})}{||\boldsymbol{d}^{(k)}||^2}$. Here $\{\rho_k\}_{k=0}^{\infty}$ is a sequence with $\rho_k > 0$ and $\lim_{k\to\infty} \rho_k = 0$. UB is some upper bound of $w(\boldsymbol{\lambda})$. For this upper bound we use the objective value of the solution found by the three-phase algorithm. $\boldsymbol{d}^{(k)}$ is the step direction.

Step 3: Solve the relaxed problem to find $w(\boldsymbol{\lambda}^{(k)})$ and corresponding solution $(\boldsymbol{x}, \boldsymbol{y})^{(k)}$.

Step 4: Determine step direction $\boldsymbol{d}^{(k)}$. For pure subgradient optimisation, the step direction $\boldsymbol{d}^{(k)}$ is equal to subgradient $\boldsymbol{s}^{(k)} = \boldsymbol{x}^{(k)} - \sum_{e \in \mathcal{E}} \boldsymbol{y}_e^{(k)}$, where $\boldsymbol{y}_e$ is the column of matrix $\boldsymbol{y}$ corresponding to employee e.

Step 5: Update values for $\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + \mu_k \boldsymbol{d}^{(k)}$. Unless $\boldsymbol{s}^{(k)} = 0$, return to Step 3.

It can be noted that no Lagrangian heuristic is used in this procedure. A Lagrangian heuristic is a heuristic procedure to obtain a solution to the TSP from the solution to the relaxed problem found in step 3 of the procedure. When a Lagrangian heuristic is included in the subgradient optimisation procedure, not only can a lower bound to the TSP be found but also a feasible solution to the TSP. However, preliminary results have shown that the solution to the relaxed problem often include cases of some shifts being assigned to all employees. This occurs when the multiplier value of one shift makes it more favourable to be assigned than other shifts. Generally, these solutions to the relaxed problem unfortunately do not resemble meaningful solutions to the TSP. Because of this no effective Lagrangian heuristic is found, and this step is thus emitted.

As a subgradient, we use the vector $\boldsymbol{s}^{(k)} = \boldsymbol{x}^{(k)} - \sum_{e \in \mathcal{E}} \boldsymbol{y}_e^{(k)}$, we will show that this is a subgradient for the function $w(\boldsymbol{\lambda})$. It should be noted that the problem at hand is a maximisation problem, and therefore the subgradient $\boldsymbol{s}^{(k)}$ should satisfy $w(\boldsymbol{\lambda}) - w(\boldsymbol{\lambda}^{(k)}) \leq \boldsymbol{s}^{(k)\intercal}(\boldsymbol{\lambda} - \boldsymbol{\lambda}^{(k)})$.

$$
\begin{aligned}
w(\boldsymbol{\lambda}) - w(\boldsymbol{\lambda}^k) \\
&= \min \Big\{ \sum_{i \in \mathcal{I}} c_i^- x_i^- + c_i^+ x_i^+ + \sum_{e \in \mathcal{E}} k_e^- y_e^- + k_e^+ y_e^+ + \sum_{s \in \mathcal{S}} \lambda_s \Big(x_s - \sum_{e \in \mathcal{E}} y_{se} \Big) : \ (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{F}_{RTSP} \Big\} \\
&\quad - \sum_{i \in \mathcal{I}} c_i^- x_i^{-(k)} + c_i^+ x_i^{+(k)} - \sum_{e \in \mathcal{E}} k_e^- y_e^{-(k)} + k_e^+ y_e^{+(k)} - \sum_{s \in \mathcal{S}} \lambda_s^{(k)} \Big(x_s^{(k)} - \sum_{e \in \mathcal{E}} y_{se}^{(k)} \Big) \\
&\leq \sum_{i \in \mathcal{I}} c_i^- x_i^{-(k)} + c_i^+ x_i^{+(k)} + \sum_{e \in \mathcal{E}} k_e^- y_e^{-(k)} + k_e^+ y_e^{+(k)} + \sum_{s \in \mathcal{S}} \lambda_s \Big(x_s^{(k)} - \sum_{e \in \mathcal{E}} y_{se}^{(k)} \Big) \\
&\quad - \sum_{i \in \mathcal{I}} c_i^- x_i^{-(k)} + c_i^+ x_i^{+(k)} - \sum_{e \in \mathcal{E}} k_e^- y_e^{-(k)} + k_e^+ y_e^{+(k)} - \sum_{s \in \mathcal{S}} \lambda_s^{(k)} \Big(x_s^{(k)} - \sum_{e \in \mathcal{E}} y_{se}^{(k)} \Big) \\
&= \sum_{s \in \mathcal{S}} (\lambda_s - \lambda_s^{(k)}) \Big(x_s^{(k)} - \sum_{e \in \mathcal{E}} y_{se}^{(k)} \Big) = \Big(\boldsymbol{x}^{(k)} - \sum_{e \in \mathcal{E}} \boldsymbol{y}_e^{(k)} \Big)^\intercal (\boldsymbol{\lambda} - \boldsymbol{\lambda}^{(k)})
\end{aligned}
$$

Deflected subgradient methods In the pure subgradient method, the step direction $\boldsymbol{d}^{(k)}$ is simply equal to the subgradient $\boldsymbol{s}^{(k)}$. In deflected subgradient methods, the step direction is a function of the subgradient and the step direction of the previous iteration. For these methods a deflection parameter is determined in every iteration, denoted by Ψ_k. The step direction is then given by $\boldsymbol{d}^{(k)} = \boldsymbol{s}^{(k)} + \Psi_k \boldsymbol{d}^{(k-1)}$. If the subgradient forms an obtuse angle with the previous step direction, i.e. if $\boldsymbol{s}^{(k)} \boldsymbol{d}^{(k-1)} < 0$, the deflection parameter is non-zero and the step direction is partly determined by the previous step direction. The formulas for the deflection parameters for the modified gradient technique (MGT) and average direction strategy (ADS) are given:

$$
\text{Modified gradient technique: } \Psi_k = \begin{cases} -\eta \frac{\boldsymbol{s}^{(k)} \boldsymbol{d}^{(k-1)}}{||\boldsymbol{d}^{(k-1)}||^2}, & \boldsymbol{s}^{(k)} \boldsymbol{d}^{(k-1)} < 0 \\ 0, & \text{otherwise} \end{cases} \tag{46}
$$

$$
\text{Average direction strategy: } \Psi_k = \begin{cases} \frac{||\boldsymbol{s}^{(k)}||}{||\boldsymbol{d}^{(k-1)}||}, & \boldsymbol{s}^{(k)} \boldsymbol{d}^{(k-1)} < 0 \\ 0, & \text{otherwise} \end{cases} \tag{47}
$$

For the modified gradient technique $\eta \in (0, 2]$. In this thesis the value for η is chosen to be 1.5, as recommended by Guta (2003).

5 Results

In this section results illustrating the performance of the methods presented in section 4 are shown. First a description of the data used to generate the results is given. Next some results from the three-phase algorithm are presented. Finally results from the Lagrangian relaxation and subgradient optimisation methods to obtain a lower bound are given.

5.1 Data

Several real life data sets of different sizes are available to assess the performance of the algorithm designed in this thesis. These data sets are described in the following paragraphs.

Data set 1 The first data set is small. It contains 3 employes and 42 different possible shifts in a one week planning horizon. Each employee is contracted to work 38 hours per week. The penalty costs for under scheduling are set to 600 per hour. Shifts can have a duration of 6 hours, 8 hours or 10 hours. For this data set, 3 shifts are added to the candidate shift set for each day, totalling 21. The staffing requirements are one employee each day from 7:00 until 23:00. These requirements are depicted in Figure 2. The penalty costs for under coverage are set to 1000 per hour and the penalty cost for over coverage to 100 per hour. In this data set, the total number of contracted hours of all employees is slightly larger than the total requirements throughout the week. Constraints are added to the NRP such that the following rules with regards to assigning shifts are adhered to:

- An employee can be assigned to at most one shift per day.
- An employee can have at most 5 working days per week.
- Any two shifts assigned to one employee must be at least 11 hours apart in time.
- An employee cannot work more that their contracted hours.
- If an employee has a rest day preceded and followed by working days, the rest period between shifts must be at least 36 hours.

Data set 2 The second data set is somewhat larger than the first, it contains 8 employees and 4624 different possible shifts in a one week planning horizon. The duration of possible shifts is a multiple of half hours and ranges between 4 and 12 hours. For this data set, 50 shifts are added to the candidate shift set for each day, totalling 350. Most employees have a contract to work for 38 hours, the average contract is to work for 32 hours. The penalty costs for under scheduling are set to 600 per hour. Notably, the contracted hours of some employees is not a multiple of half hours, thus scheduling precisely for contracted hours is not always possible. The staffing requirements vary between 1 and 4 employees required, between 7:00 and 23:00. The requirements are depicted in Figure 3. The penalty costs for under coverage are 2000 per hour and the penalty costs for over coverage to 200 per hour. In this data set the total number

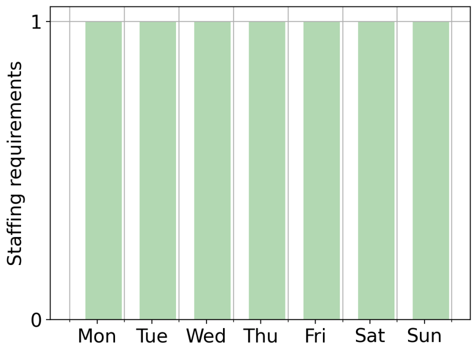

Figure 2: Staffing requirements from data set 1

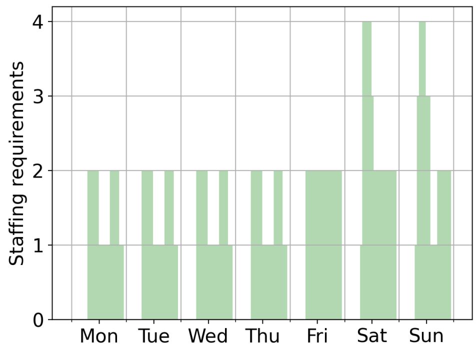

Figure 3: Staffing requirements from data set 2

of contracted hours of all employees is larger than the total requirements throughout the week. Constraints are added to the NRP such that the following rules are adhered to:

- An employee can be assigned to at most one shift per day.
- An employee can have at most 5 working days per week.
- Any two shifts assigned to one employee must be at least 11 hours apart in time.
- An employee cannot work more that their contracted hours.
- If an employee has a day off, the preceding or following day must also be a day off.
- If an employee has a working day, the preceding or following day must also be a working day.

Data set 3 The third data set is a larger data set. It contains 65 employees and 26901 different possible shifts in a one week period. The duration of possible shifts is a multiple of half hours and ranges between 4 and 13 hours. Shifts can have one or multiple breaks, depending on their duration. For this data set, 70 shifts are added to the candidate shift set for each day, totalling 490. The staffing requirements vary drastically between 2 and 69 between 10:00 and 24:00. The requirements are depicted in Figure 4. The penalty costs for under coverage are 2000 per hour and the penalty costs for over coverage are 200 per hour. Notably, in this data set, at peak hour the staffing requirements are higher than the number of employees available. Some under coverage is thus unavoidable. In this data set, all employees have contracts that specify a range of allowed working hours per planning period. The average maximum working hours is 24.4 hours, the average minimum working hours 21.9. The total requirements throughout the week lies between the total minimum and maximum working hours of all employees. The penalty costs for under scheduling are 1200 per hour. Constraints are added to the NRP such that the following rules are adhered to:

- An employee can be assigned to at most one shift per day.

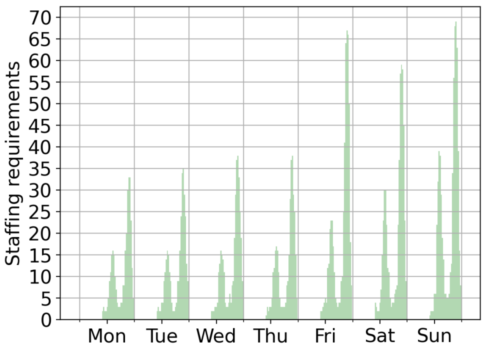

Figure 4: Staffing requirements from data set 3

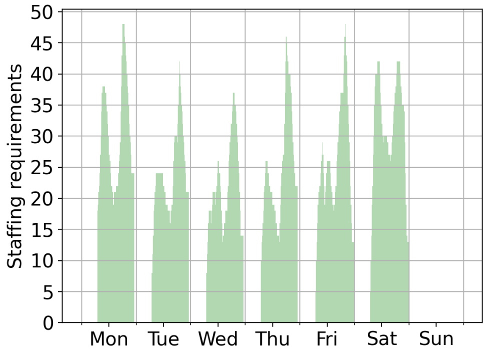

Figure 5: Staffing requirements from data set 4

- Any two shifts assigned to one employee must be at least 12 hours apart in time.
- An employee cannot work more that their contracted hours.
- Employees have periods of unavailability during which they cannot be assigned to work. The employees have 2.7 periods of unavailability during the one week planning horizon on average. Each period of unavailability lasts 7 hours and 44 minutes on average.

Data set 4 The last data set is the largest data set. It contains 108 employees and 53148 different possible shifts. The duration of possible shifts is a multiple of quarter hours and ranges between 4 and 8.5 hours. Shifts can have a break, depending on the duration. For this data set, 500 shifts are added to the candidate shift set for each day, totalling 1800. Employees are contracted to work a fixed number of hours per week, on average 26.5 hours per week. Contracted hours range between 10 and 40. The penalty costs for under scheduling are 600 per hour. The staffing requirements vary between 5 and 48, between 7:15 and 24:00. The requirements are depicted in Figure 5. The penalty costs for under coverage are 800 per hour and the penalty costs for over coverage are 400 per hour. In this data set the total number of contracted hours of all employees is larger than the total requirements throughout the week. Constraints are added to the NRP such that the following rules are adhered to:

- An employee can be assigned to at most one shift per day.
- Any two shifts assigned to one employee must be at least 11 hours apart in time.
- An employee cannot work more that their contracted hours.
- An employee can have at most 5 working days per week.
- Employees have periods of unavailability during which they cannot be assigned to work. The employees have 0.64 periods of unavailability during the one week planning horizon on average. Each period of unavailability lasts 38 hours and 45 minutes on average.

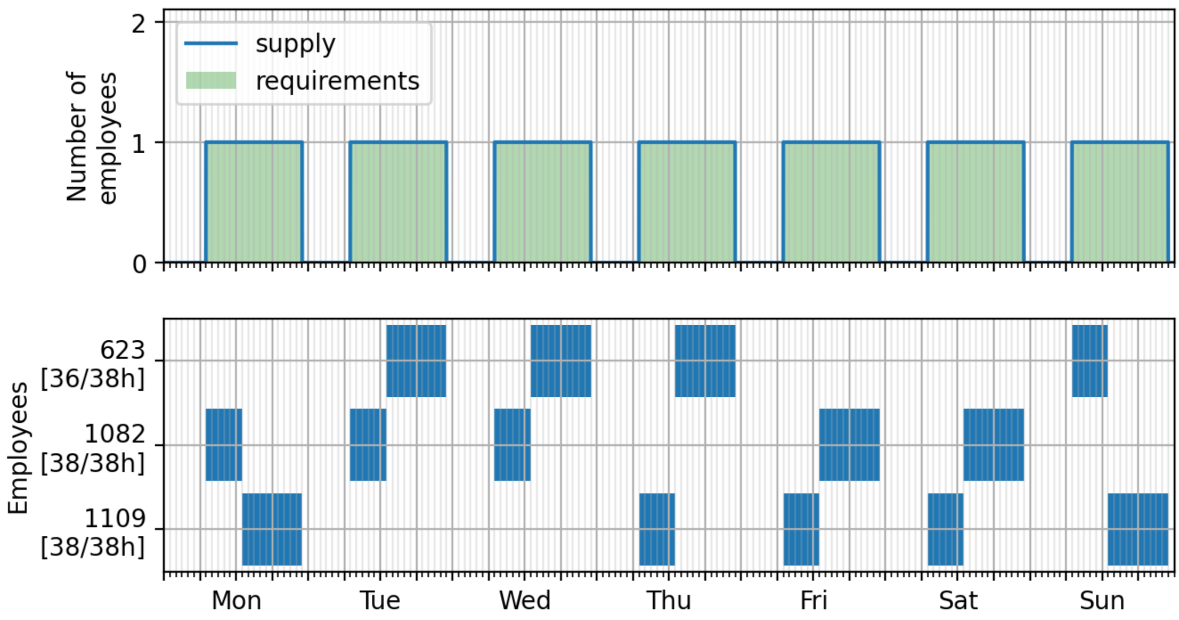

Figure 6: Depiction of the initial schedule for data set 1. In the top graph, the staffing requirements and the supply of employees scheduled are plotted. In the bottom graph, the schedule for each employee is depicted, where a blue bar indicates a shift.

5.2 Three-phase algorithm

In the following sections, the results of the three-phase algorithm with each of the available data sets are presented.

5.2.1 Data set 1

In the initial solution of the SSP, the shifts perfectly cover the demand, i.e. there is no over or under coverage. An optimal solution to the SSP is found within 0.7 seconds. In the initial solution to the NRP, no shifts are left unassigned so there is no over or under coverage. Two of the three employees are scheduled to work precisely for their contracted hours, one employee is under scheduled by 2 hours. An optimal solution to the NRP is found within 0.3 seconds.

For this data set, intermediate steps of the algorithm are displayed to illustrate the working of the algorithm. In Figure 6 the schedule obtained by solving the SSP and the NRP consecutively. The shifts perfectly cover the staffing requirements. Employees are anonymized and identified by an ID number. Employee 1082 and 1109 are scheduled to work for a total of 38 hours, which is in line with their contracts. Employee 623 is scheduled to work for 36 hours, and is thus 2 hours under scheduled. Employee 623 is identified by the algorithm as the most under scheduled employee, and the destroyed schedule is obtained by removing all shifts assigned to employee 623 from the schedule. The destroyed schedule is depicted in Figure 7. Now the candidate shift set is obtained by scoring all possible shifts. On Monday, Friday and Saturday, there is no under coverage so no shifts on those days reduce any under coverage. All shifts on those days

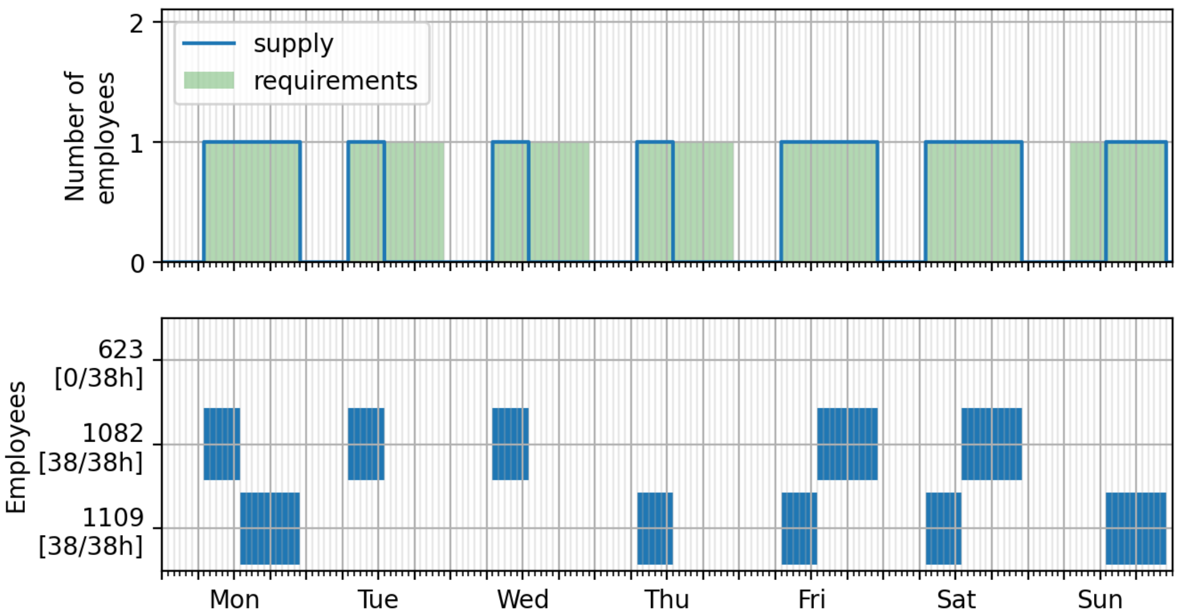

Figure 7: Depiction of the destroyed schedule in iteration 1 for data set 1. In the top graph, the staffing requirements and the supply of employees scheduled are plotted. In the bottom graph, the schedule for each employee is depicted, where a blue bar indicates a shift.

obtain the score 0, and two random shifts are added to the candidate set. Shifts on Tuesday, Wednesday, Thursday and Sunday obtain a score equal to the reduction of under coverage. The candidate shifts set obtained is listed in Table 2.

Table 2: Candidate shift set

Shift ID number	Day	Start time	End time	Shift duration (hours)	Score
1	Monday	13:00	23:00	10	0
2	Monday	7:00	17:00	10	0
3	Monday	7:00	15:00	8	0
4	Tuesday	13:00	23:00	10	-10
5	Tuesday	15:00	23:00	8	-8
6	Tuesday	17:00	23:00	6	-6
7	Wednesday	13:00	23:00	10	-10
8	Wednesday	15:00	23:00	8	-8
9	Wednesday	17:00	23:00	6	-6
10	Thursday	13:00	23:00	10	-10
11	Thursday	15:00	23:00	8	-8
12	Thursday	17:00	23:00	6	-6
13	Friday	13:00	23:00	10	0
14	Friday	7:00	17:00	10	0
15	Friday	15:00	23:00	8	0
16	Saturday	13:00	23:00	10	0
17	Saturday	7:00	15:00	8	0
18	Saturday	7:00	13:00	6	0
19	Sunday	7:00	17:00	10	-6
20	Sunday	7:00	13:00	6	-6
21	Sunday	7:00	15:00	8	-6

Finally the adjusted version of the NRP is solved to obtain the final schedule depicted in Figure 8. In the final schedule, there are two hours of over coverage on Sunday between 13:00 and 15:00. All employees are scheduled for precisely their contracted hours, this causes the algorithm to terminate.

In Figure 9 the progress of the objective, under coverage, over coverage and under scheduled hours are plotted. Overall, the objective decreased from 1200 to 200, a reduction of 83%.

For data set 1, the algorithm terminated after one iteration. The computation time for the algorithm to terminate was 0.2 seconds, thus taking 0.2 seconds per iteration.

5.2.2 Data set 2

For the second data set, an optimal initial solution to the SSP is obtained after 7.8 seconds. The shifts selected by the SSP perfectly cover the staffing requirements. An optimal initial solution to the NRP is obtained after 2.7 seconds. All shifts from the SSP are assigned, so there is no under or over coverage in the initial solution. All eight employees are under scheduled in the initial solution, with a total of 49.4 under scheduled hours.

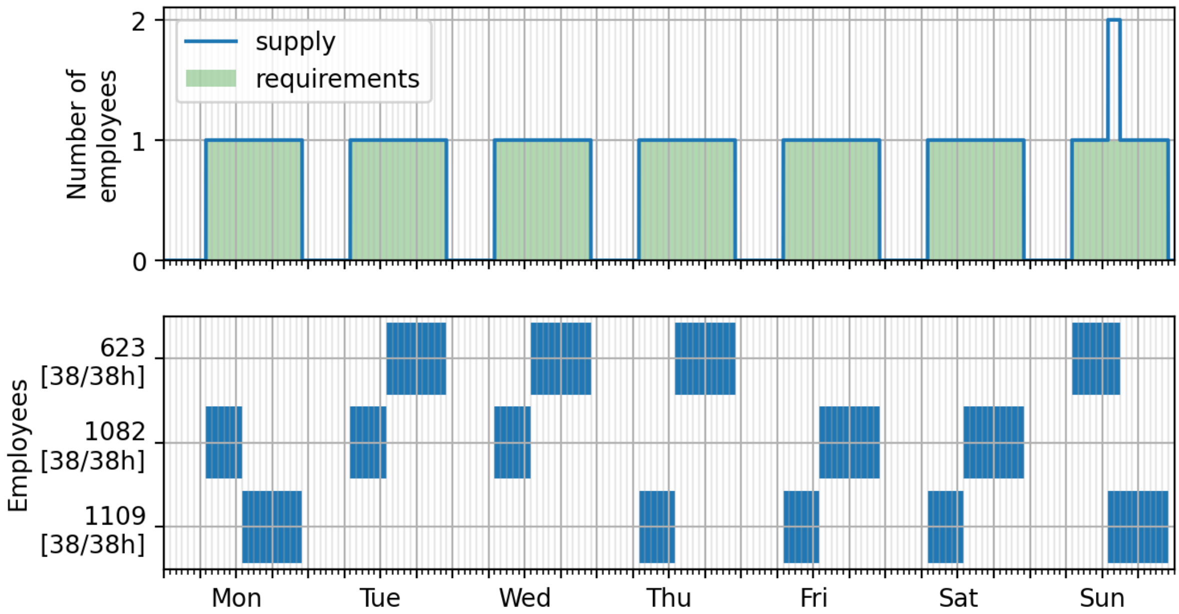

Figure 8: Depiction of the final schedule for data set 1. In the top graph, the staffing requirements and the supply of employees scheduled are plotted. In the bottom graph, the schedule for each employee is depicted, where a blue bar indicates a shift.

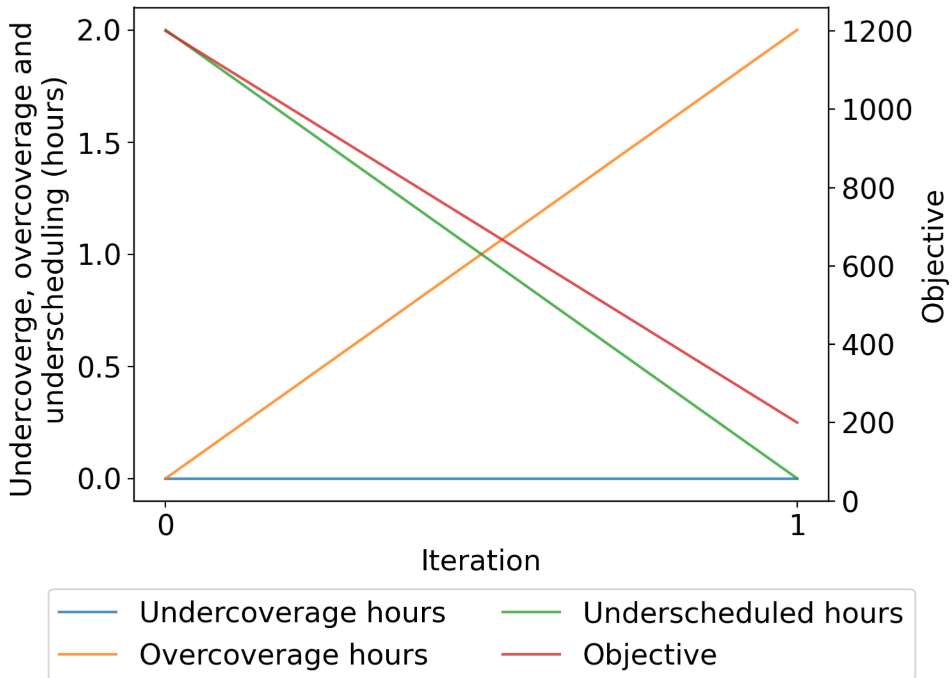

Figure 9: Progress of under coverage, over coverage, under scheduling (on the left axis) and the objective (on the right axis) throughout the iterations of the algorithm for data set 1.

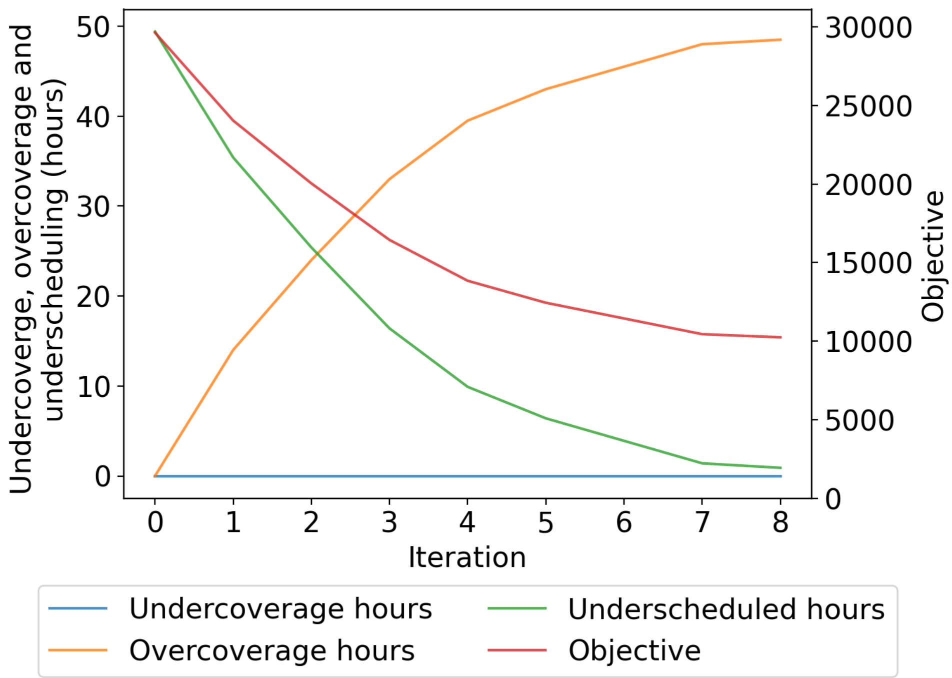

Figure 10: Progress of under coverage, over coverage, under scheduling (on the left axis) and the objective (on the right axis) throughout the iterations of the algorithm for data set 2.

The progress of the objective, under coverage, over coverage and under scheduling are plotted in Figure 10. Only those iterations in which the candidate solution was accepted are included in the graph.

In the final solution, there is no under coverage, there are 48.5 hours of over coverage, and 0.9 hours of under scheduling. Overall, the objective decreased from 29640 to 10240, a reduction of 65%.

The algorithm terminated after 10 iterations and 128.5 seconds, thus taking 12.9 seconds per iteration.

5.2.3 Data set 3

For the third data set, an optimal initial solution to the SSP is found after 47.6 seconds. In the initial solution, there is 0.5 hour of over coverage, and 18.5 hours of under coverage. After 946 seconds, the NRP algorithm was terminated and the best known solution was used as initial solution. The objective gap at this point was 15%. In the initial solution, 15 shifts from the SSP solution are left unassigned, therefore the initial solution has 81.5 hours of under coverage. The initial solution has no over coverage and 16 employees are under scheduled, with a total of 98.7 under scheduled hours.

The progress of the objective, under coverage, over coverage and under scheduling are plotted in Figure 11. In the final solution, there are 46 hours of under coverage, there are 71 hours of

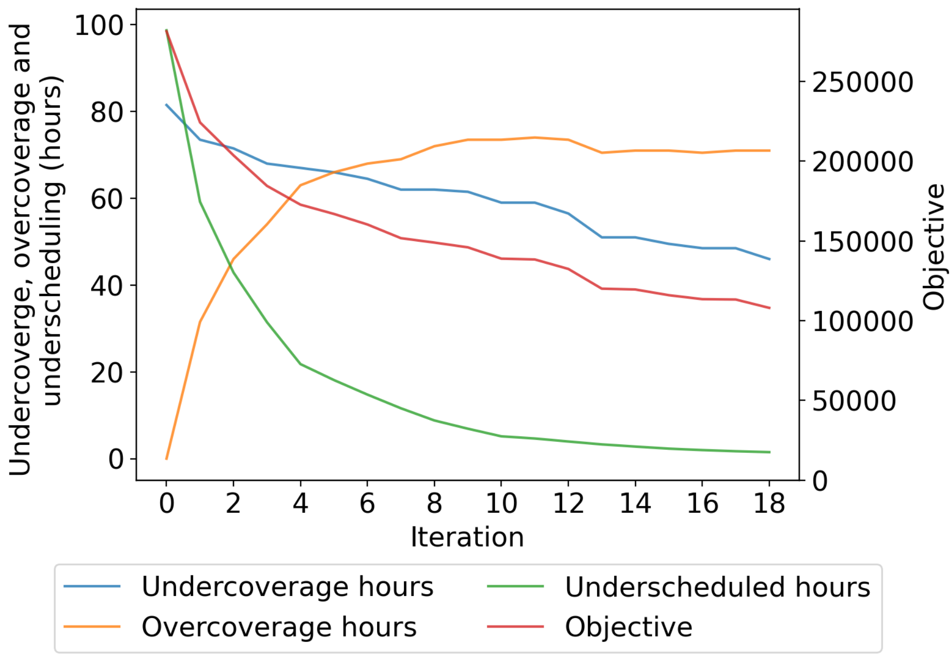

Figure 11: Progress of under coverage, over coverage, under scheduling (on the left axis) and the objective (on the right axis) throughout the iterations of the algorithm for data set 3.

over coverage, and 1.5 hours of under scheduling. Overall, the objective decreased from 281440 to 108000, a reduction of 62%.

The algorithm terminated after 25 iterations or 372.2 seconds, thus taking 14.9 seconds per iteration.

5.2.4 Data set 4

For the final data set, an optimal solution to the SSP is found within 51.6 seconds. In the initial solution, there are 14 hours and 45 minutes of under coverage, and 15 minutes of over coverage. An optimal solution to the NRP was found after 935.7 seconds. In the initial solution, all shifts are assigned and 89 employees are under scheduled, for a total of 171 hours and 45 minutes.

The progress of the objective, under coverage, over coverage and under scheduling are plotted in Figure 12. In the final solution, there are 13 hours and 30 minutes of under coverage, 170 hours and 45 minutes of over coverage and no under scheduling. The algorithm terminated after 89 iterations or 4552 seconds, thus taking 51.1 seconds per iteration. Overall, the objective decreased from 114950 to 79100, a reduction of 31%.

5.3 Lagrangian relaxation and subgradient optimisation

In this section, some results from the Lagrangian relaxation and subgradient optimisation method are presented. First some results of the symmetry breaking constraints are shown.

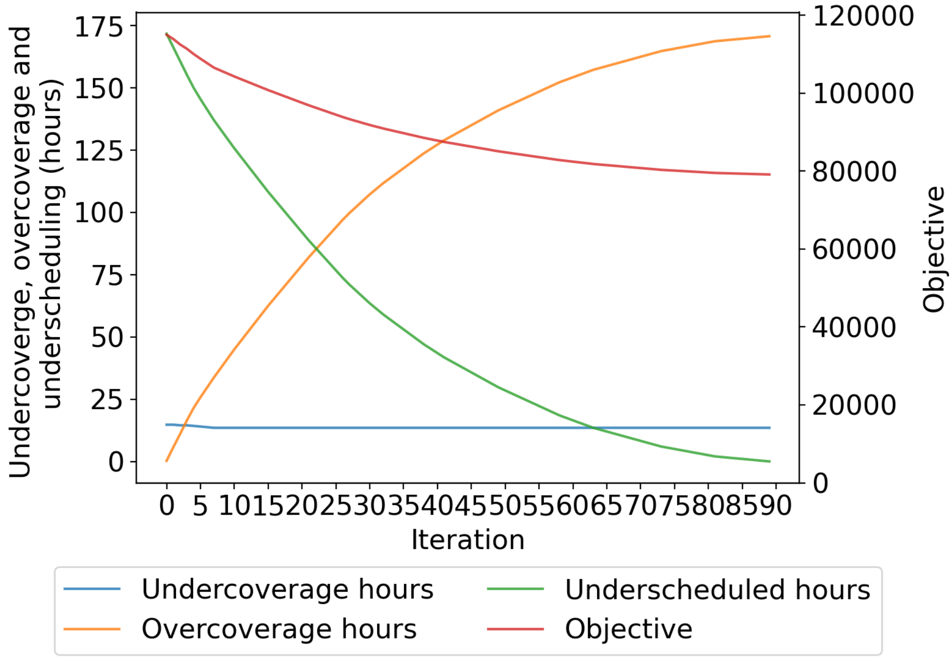

Figure 12: Progress of under coverage, over coverage, under scheduling (on the left axis) and the objective (on the right axis) throughout the iterations of the algorithm for data set 4.

Next results from the subgradient optimisation for data set 1 are presented. Finally some results for data set 2 are shown.

5.3.1 Symmetry breaking constraints

Symmetry breaking constraints are added to the formulation of the NRP-subproblem of the Lagrangian Relaxation for each set of identical employees, as described in section 4.2.2. In Table 3 is listed how many sets of identical employees are detected in each of the data sets, as well as the number of symmetry breaking constraints that arise from these sets. To investigate the effect on the computation time of the symmetry breaking constraints, 50 iterations of the pure

Table 3: The sets of identical employees identified in each data set is listed. Also the number of symmetry breaking constraints that arise from these sets. Data sets 1 and 2 contain one set of identical employees each, data set 3 contains no sets of identical employees. Data set 4 contains several sets of identical employees

Data set	Number of employees	Cardinality of sets of identical employees	Number of symmetry breaking constraints
1	3	[3]	84
2	8	[5]	18496
3	65	[]	0
4	108	[9, 21, 18, 28, 24]	5049060

subgradient optimisation method are executed for data set 1, with and without the symmetry breaking constraints. Without the symmetry breaking constraints, the total computation time spent on solving the NRP-subproblem of the Lagrangian relaxation is 1 minute and 19 seconds. With the symmetry breaking constraints the total computation time spent on solving the NRP-subproblem of the Lagrangian relaxation is 1 minute and 33 seconds. In data set 1, the number of shifts is 14 times as large as the number of employees, for data set 2, 3 and 4 it is 578, 414 and 492 times as large respectively. Furthermore, in data set 1 all employees are 'identical' whereas data sets 2, 3 and 4 contain a more diverse employee force. From these observations it is concluded that lexicographical symmetry constrains will eliminate the a higher fraction of solutions for data set 1 compared to data sets 2, 3 and 4, and thus lexicographical symmetry breaking constraints are expected to reach the most computation time reduction in data set 1. Due to these results, symmetry breaking constraints are omitted from the formulation of the Lagrangian relaxation in all further experiments in this thesis.

5.3.2 Data set 1

Experiments are conducted for the pure subgradient method, modified gradient technique and average direction strategy. Furthermore different definitions of the sequence $\{\rho_k\}_{k=0}^{\infty}$ are used. Finally different initial values for $\boldsymbol{\lambda}^{(0)} \in \mathbb{R}^m$ are taken, where m is the number of possible shifts $s \in \mathcal{S}$. On average, one iteration of subgradient optimisation for data set 1 takes 4.86 seconds, 0.34 seconds to solve the SSP-subproblem and 4.52 seconds to solve the NRP-subproblem. More detailed records of the computation time for each of the different experiments conducted are listed in the appendix in Table 5.

In Figure 13, some results for the pure subgradient optimisation method are presented. In each of the experiments, the initial value of the multipliers are set to $\boldsymbol{\lambda}^{(0)} = 0$, different definitions of the sequence $\{\rho_k\}_{k=0}^{\infty}$ are used. The graph shows the lower bound obtained in each iteration. Similar experiments have been conducted for the modified gradient technique and average direction strategy. The results from these experiments are similar to the results for the pure subgradient method. The results from these experiments are depicted in Figures 16 and 17 in the appendix. For all three different subgradient methods, the quickest convergence is observed for sequence $\rho_k = \frac{0.25}{k+1.5}$ and $\rho_k = \frac{0.25}{k+3}$. For all different definitions of sequence $\{\rho_k\}_{k=0}^{\infty}$ the best bound is obtained in the first iteration and is equal to $z^*_{RTSP} = 0$. This is a trivial lower bound.

In Figure 14 some results obtained with each of the three subgradient optimisation methods are depicted in the same graph. The initial values of the multipliers are set to $\boldsymbol{\lambda}^{(0)} = 0$, the sequence $\{\rho_k\}_{k=0}^{\infty}$ is defined as $\rho_k = \frac{0.25}{k+3}$, because this is one of the best performing definitions for each of the methods. The best bound obtained in each of these experiments is obtained in the first iteration and is equal to $z^*_{RTSP} = 0$.

In Figure 15 some results obtained with different starting values for $\boldsymbol{\lambda}^{(0)}$ are depicted. All three different subgradient optimisation methods are used, as well as different definitions for sequence $\{\rho_k\}_{k=0}^{\infty}$. For sequence $\{\rho_k\}_{k=0}^{\infty}$, one of the best performing definitions is chosen ($\rho_k = \frac{0.25}{k+3}$), as

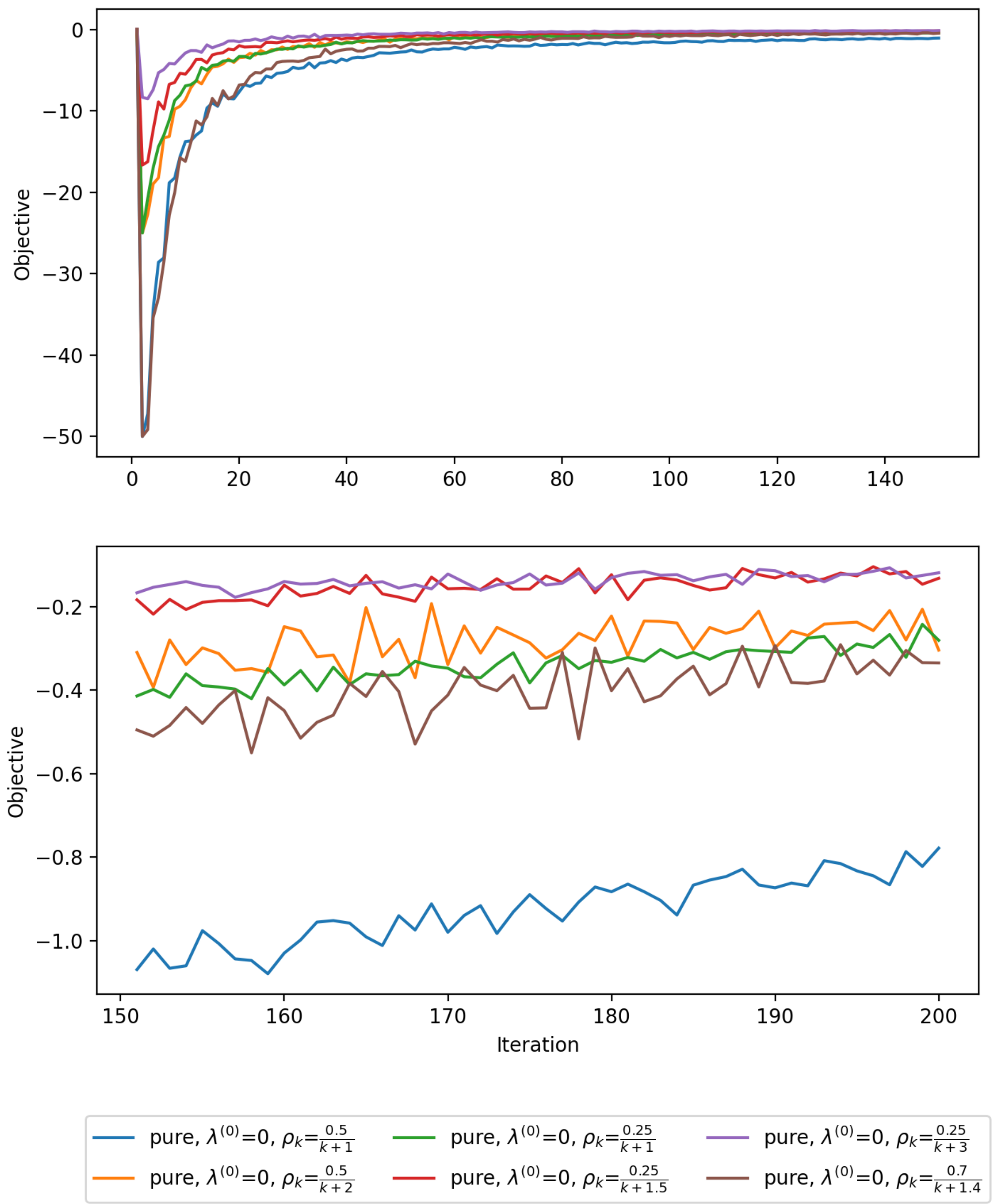

Figure 13: Lower bound to the objective of the TSP of data set 1 after each iteration. The pure subgradient method is used and the multipliers are initialized at $\boldsymbol{\lambda}^{(0)} = 0$. Different definitions of sequence $\{\rho_k\}_{k=0}^{\infty}$ are used. For readability, separate graphs are made for the first 150 iterations and the final 50.

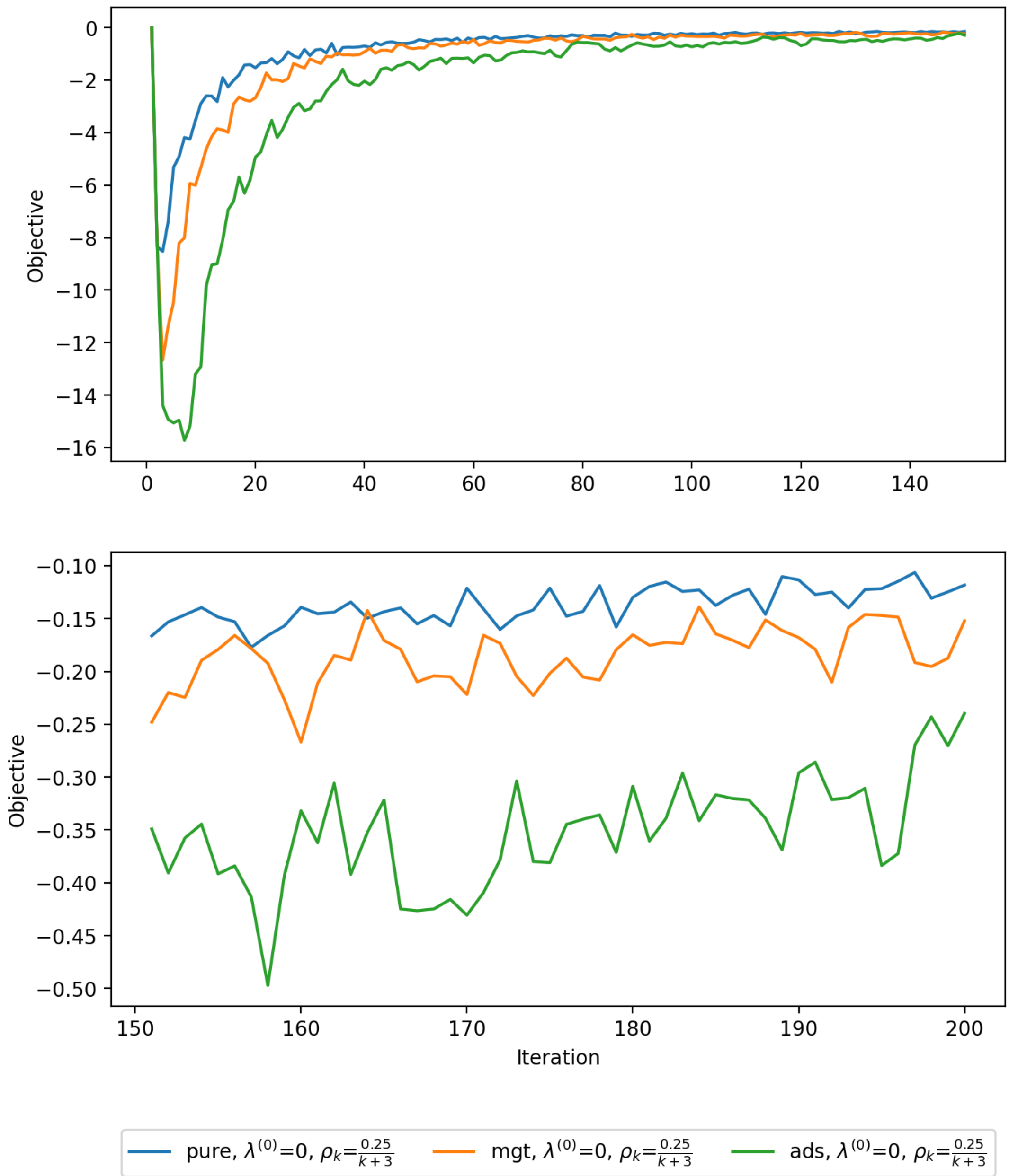

Figure 14: Lower bound to the objective of the TSP of data set 1 after each iteration. Different subgradient optimisation methods are used. The multipliers are initialized at $\boldsymbol{\lambda}^{(0)} = 0$. Sequence $\{\rho_k\}_{k=0}^{\infty}$ is defined as $\rho_k = \frac{0.25}{k+3}$. For readability, separate graphs are made for the first 150 iterations and the final 50.

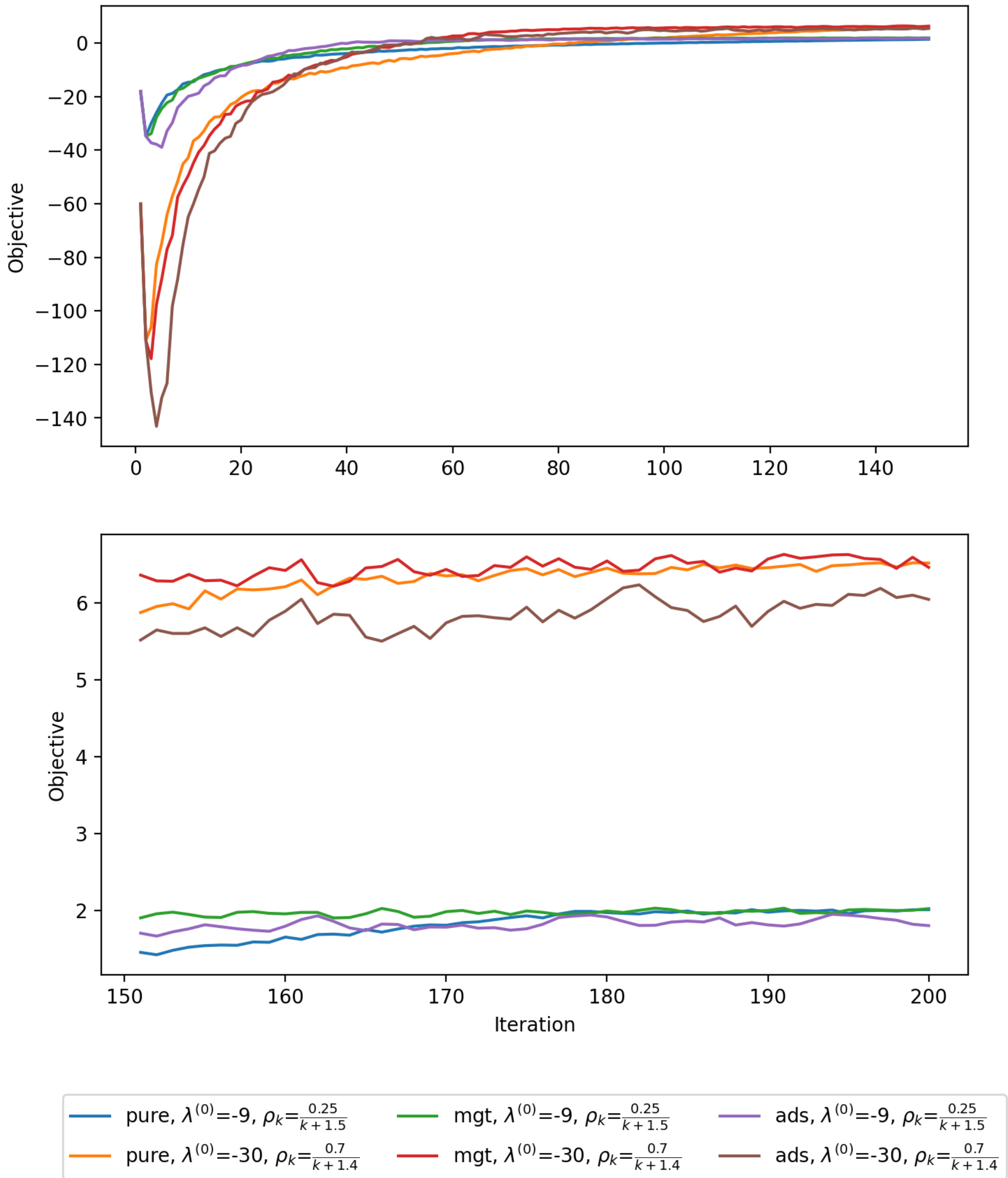

Figure 15: Lower bound to the objective of the TSP of data set 1 after each iteration. Different subgradient optimisation methods are used. The multipliers $\boldsymbol{\lambda}^{(0)}$ are initialized at different values. Different definitions for sequence $\{\rho_k\}_{k=0}^{\infty}$ are used. For readability, separate graphs are made for the first 150 iterations and the final 50.

well as a definition that is seen to yield slower convergence for each of the methods ($\rho_k = \frac{0.7}{k+1.4}$). Different starting values $\boldsymbol{\lambda}^{(0)}$ are used for both of these definitions for $\{\rho_k\}_{k=0}^{\infty}$. The best performing starting values $\boldsymbol{\lambda}^{(0)}$ that have been found are plotted in Figure 15. The convergence is very sensitive to different starting values $\boldsymbol{\lambda}^{(0)}$. In Table 4, the best bound obtained in each experiment is listed, as well as the iteration in which this best bound was obtained.

Table 4: Best bound lower bound for the TSP of data set 1 obtained in different experiments of subgradient optimisation. The last column indicates in which iteration the best bound was obtained

Subgradient optimisation method	$\boldsymbol{\lambda}^{(0)}$	$\{\rho_k\}_{k=0}^{\infty}$	**Best bound**	**Obtained in iteration**
pure	-9	$\rho_k = \frac{0.25}{k+1.5}$	2.008	189
pure	-30	$\rho_k = \frac{0.7}{k+1.4}$	6.520	199
MGT	-9	$\rho_k = \frac{0.25}{k+1.5}$	2.027	191
MGT	-30	$\rho_k = \frac{0.7}{k+1.4}$	6.629	191
ADS	-9	$\rho_k = \frac{0.25}{k+1.5}$	1.948	194
ADS	-30	$\rho_k = \frac{0.7}{k+1.4}$	6.232	182

5.3.3 Data set 2

For data set 2, one iteration of subgradient optimisation is attempted. While the number of employees in data set two is not much larger than in data set 1, the number of shifts is significantly bigger. For the SSP-subproblem, an optimal solution is found within 9.46 seconds. The objective of this subproblem is 0. For the NPR-subproblem however, after 1 hour and 30 minutes, no optimal solution has been found. The upper and lower bound to the objective are 540 and 144 respectively. Due to these results and considering data sets 3 and 4 not only have a lager number of employees but also a larger number of shifts, no experiments with subgradient optimisation are conducted for data sets 3 and 4.

6 Conclusion and discussion

In this section some conclusions about the methods developed in this thesis are presented, furthermore some possible extensions and improvements are discussed. First the three-phase algorithm is discussed, subsequently the Lagrangian relaxation and subgradient optimisation method.

6.1 Three-phase algorithm

In this thesis, a heuristic algorithm is presented to obtain solutions to the operational Tour Scheduling Problem (TSP), i.e. the problem of creating a schedule of shifts to cover staffing requirements of some business and assigning those shifts to employees. Heuristic methods to solve the TSP that include decomposition steps where a Shift Scheduling Problem (SSP) and a Nurse Rostering Problem (NRP) are solved are popular in literature because the TSP is a complex, NP-hard problem. The drawback of decomposing the TSP in a SSP and a NRP is that shifts are chosen without considering if a favourable allocation of those shifts to the employees exists. Whenever the available employee force is not flexible, i.e. the number of employees is fixed and employees' contracts state how many hours an employee must work per week precisely, decomposition methods may not perform well. Decomposition methods can be expected to yield bad results whenever contracted hours of employees exceed the required workforce over a planning horizon, or whenever many rules dictate how shifts must be assigned to employees. The heuristic algorithm designed in this thesis is developed specifically to overcome these shortcomings of the decomposition method. The heuristic algorithm consists of three phases. The first two phases are the decomposition phases where the SSP and the NRP are solved respectively. The third phase is an improvement phase designed to improve under scheduling and under coverage. Experiments to determine the effectiveness of the algorithm are conducted using four real-world data sets of different sizes.

For all data sets, the three-phase algorithm, the initial solution found after the decomposition phases was improved upon in the improvement phase. On average, 99% of under scheduling present in the initial solution is remedied by the improvement phase. In case the initial solution contains under coverage, this is reduced on average 26.1%. Overall, the objective was reduced on average by 60%. The computation time for the algorithm increases significantly with the number of employees and shifts. In the decomposition phases the NP-hard SSP and NRP are solved. In the improvement phase, the computation time per iteration depends on the number of shifts, the number of iterations needed depends on the number of employees that are under scheduled in the initial solution. The number of iterations could be limited if the initial solution contains as many as possible properly scheduled employees, leaving a few severely under scheduled. This could be obtained adding constraints and objective terms to the NRP, which determine that the penalty cost per hour of under scheduling for each employee decreases step-wise. Furthermore, the improvement to the objective per iteration of the improvement phase decreases. Whenever computation time is of the essence it can be considered to terminate the algorithm after a set number of iterations.

As mentioned, the three phase algorithm is designed with employees with fixed contracted hours in mind, as this is a situation where decomposition methods can be expected to perform poorly. However, the algorithm was also tested with a data set where employees do not have fixed contracted hours, but are allowed to be scheduled for some range of hours (data set 3). In this data set, the algorithm performed well, decreasing both under scheduling and under coverage. The algorithm could be made more generally applicable by changing the way employees are selected in the destroy method of the improvement phase. The following is suggested for choosing en employee: choose the employee who, in the current schedule, is scheduled during time intervals where there is over coverage, and is not scheduled during time intervals during which there is under coverage. The purpose of selecting employees this way is that, hopefully, the algorithm can change the working hours of the employee from periods of over coverage to periods of under coverage.

Finally some comments are made with regard to the practicality of the three-phase algorithm. In the three-phase algorithm, rules pertaining to the assignment of shifts to employees are considered in the both the second (NRP phase) and third (improvement phase) phase. These rules may be complex and may require quite some complicated, error prone code. A well-known principle of software development is DRY: Don't Repeat Yourself. The three-phase algorithm nicely complies with this principle as both the NRP phase and the improvement phase use a formulation of the NRP. When using the three-phase algorithm, one implementation of the NRP can be used for both steps, reducing the work in coding and code upkeep.
Finally, in the repair step of the improvement phase a scoring method is used to determine which shifts are the 'best'. The main component of this score is the reduction in under coverage that the shift achieves, however this scoring method can be extended with custom criteria. For example, if shifts of a certain length are know to be preferred by employees, the score can incorporate a bonus for shifts with this length, yielding a schedule more in line with employee preference.

6.2 Lagrangian relaxation and subgradient optimisation

The second algorithm that is investigated in this thesis is a Lagrangian relaxation and subgradient optimisation method. The aim of this method is to provide a lower bound to the TSP. In literature, some examples are found of such methods used to solve scheduling problems, but not the TSP. Furthermore, in literature, some adjusted subgradient optimisation methods are described. In this thesis, besides the pure subgradient method, the Modified Gradient Technique (MGT) and the Average Direction Strategy (ADS) are applied to the operational TSP.
Whenever a Lagrangian relaxation and subgradient optimisation is used in literate, often a Lagrangian heuristic is used to generate feasible solutions to the original problem in each

iteration. When applying this method to the TSP in this thesis, preliminary results showed that a Lagrangian heuristic did not yield meaningful feasible solutions to the TSP. Therefore, a Lagrangian heuristic is omitted form the subgradient optimisation method.

Furthermore, in literature research is found where lexicographical ordering constraints are used to eliminate symmetry in solution spaces to reduce computation time. In this thesis, the formulation of the Lagrangian relaxation is extended with lexicographical ordering constraints. Preliminary experiments showed that this did not decrease computation time in the subgradient optimisation. Therefore, lexicographical ordering constraints are omitted from the formulation of the Lagrangian relaxation.

Experiments are conducted for each of the three different subgradient optimisation methods using data set 1. Different starting values for the Lagrangian multipliers $\boldsymbol{\lambda}$ are used, as well as different definitions are used for sequence $\{\rho_k\}_{k=0}^{\infty}$, which is used to determine the step size for updating the values of the Lagrangian multipliers. In the first experiments, the effect of different definitions for sequence $\{\rho_k\}_{k=0}^{\infty}$ is investigated for each of the subgradient optimisation methods. Overall, no meaningful lower bound is obtained in these experiments. Different definitions of sequence $\{\rho_k\}_{k=0}^{\infty}$ do influence the convergence, but in all cases the best lower bound obtained is 0, which is a trivial lower bound. Comparing the convergence of the three subgradient methods used, it is observed that the pure subgradient method and MGT perform similarly. ADS performs slightly worse. Varying the starting value for Lagrangian multipliers $\boldsymbol{\lambda}$ a non-trivial lower bound is obtained. The best bound obtained for data set 1 is 6.629. While this bound is non-trivial, it is still well below the objective value of 200 of the solution for data set 1 found by the three-phase algorithm. The author believes this solution to be optimal, based on inspection. The convergence is very sensitive to the starting value of $\boldsymbol{\lambda}$, and specific combinations of values for $\boldsymbol{\lambda}$ and $\{\rho_k\}_{k=0}^{\infty}$ yield better bounds. An extensive grid search is needed to find such a combination, sequentially finding a 'good' starting value $\boldsymbol{\lambda}$ and 'good' definition for $\{\rho_k\}_{k=0}^{\infty}$ is observed to not be a effective strategy. For the larger data sets 2, 3 and 4, which contain more shifts, the computation time of the NRP-subproblem increases drastically. No optimal solution is found in one iteration for the NRP-subproblem for data set 2 in 1.5 hour.

In short, the Lagrangian relaxation and subgradient optimisation method only yield a trivial lower bound in many cases, to obtain a non-trivial lower bound an extensive grid search for appropriate starting values $\boldsymbol{\lambda}$ and definition of sequence $\{\rho_k\}_{k=0}^{\infty}$ is needed. For larger data sets with a higher number of shifts, the computation time per iteration increases. Considering this, it can be concluded that the Lagrangian relaxation and subgradient optimisation method as investigated in this thesis is not appropriate to be applied to the TSP.

The author conjectures that the slow convergence of this method might be caused by the a typical characteristic of the TSP, i.e. that it has many different optimal solutions, and the RTSP having even more optimal solutions. The following reasoning is proposed to explain this conjecture: In the subgradient optimisation method, multiplier values are updated until all ‘good’ solutions to the RTSP are penalized such that they are no longer ‘better’ than any solution that is also feasible to the TSP. Whenever many ‘good’ solutions to the RTSP exist, this may take many iterations. However, further research is needed to substantiate this conjecture.

Bibliography

Alfares, H. K. (2004). Survey, categorization, and comparison of recent tour scheduling literature. *Annals of Operations Research*, 127(1):145–175.

Bartholdi, J. J. (1981). A guaranteed-accuracy round-off algorithm for cyclic scheduling and set covering. *Operations Research*, 29(3):501–510.

Brusco, M. J., Jacobs, L. W., Bongiorno, R. J., Lyons, D. V., and Tang, B. (1995). Improving personnel scheduling at airline stations. *Operations Research*, 43(5):741–751.

Buffa, E. S., Cosgrove, M. J., and Luce, B. J. (1976). An integrated work shift scheduling system. *Decision Sciences*, 7(4):620–630.

Camerini, P. M., Fratta, L., and Maffioli, F. (1975). On improving relaxation methods by modified gradient techniques. *Nondifferentiable Optimization Mathematical Programming Studies*, page 26–34.

Dantzig, G. B. (1954). Letter to the editor—a comment on edie's "traffic delays at toll booths". *Journal of the Operations Research Society of America*, 2(3):339–341.

Easton, F. F. and Rossin (1991). Sufficient working subsets for the tour scheduling problem. *Management Science*, 37(11):1441–1451.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., and Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European journal of operational research*, 153(1):3–27.

Ghaddar, B., Naoum-Sawaya, J., Kishimoto, A., Taheri, N., and Eck, B. (2015). A lagrangian decomposition approach for the pump scheduling problem in water networks. *European Journal of Operational Research*, 241(2):490–501.

Guta, B. (2003). *Subgradient Optimization Methods in Integer Programming with an Application to a Radiation Therapy Problem.* doctoralthesis, Technische Universität Kaiserslautern.

Held, M., Wolfe, P., and Crowder, H. P. (1974). Validation of subgradient optimization. *Mathematical Programming*, 6(1):62–88.

Jarrah, A. I. Z., Bard, J. F., and deSilva, A. H. (1994). Solving large-scale tour scheduling problems. *Management Science*, 40(9):1124–1144.

Luce, B. J. (1973). A shift scheduling algorithm. In *44th National Meeting of ORSA*.

Luce, B. J. (1974). Employee assignment system. In *Joint National Meeting of ORSA/TIMS*.

Ostrowski, J., Anjos, M. F., and Vannelli, A. (2010). *Symmetry in scheduling problems*. Citeseer.

Restrepo, M. I., Gendron, B., and Rousseau, L.-M. (2016). Branch-and-price for personalized multiactivity tour scheduling. *INFORMS Journal on Computing*, 28(2):334–350.

Rong, A., Lahdelma, R., and Luh, P. B. (2008). Lagrangian relaxation based algorithm for trigeneration planning with storages. *European Journal of Operational Research*, 188(1):240–257.

Sherali, H. D. and Ulular, O. (1989). A primal-dual conjugate subgradient algorithm for specially structured linear and convex programming problems. *Applied Mathematics & Optimization*, 20(1):193–221.

Tang, L., Xuan, H., and Liu, J. (2006). A new lagrangian relaxation algorithm for hybrid flowshop scheduling to minimize total weighted completion time. *Computers & Operations Research*, 33(11):3344–3359. Part Special Issue: Operations Research and Data Mining.

Topaloglu, S. and Ozkarahan, I. (2004). An implicit goal programming model for the tour scheduling problem considering the employee work preferences. *Annals of Operations Research*, 128(1):135–158.

Turhan, A. M. and Bilgen, B. (2020). A hybrid fix-and-optimize and simulated annealing approaches for nurse rostering problem. *Computers & Industrial Engineering*, 145:106531.

A Table of computation times Subgradient Optimisation

Table 5: Computation time for different experiments for obtaining the lower bound to the TSP of data set 1 with Subgradient Optimisation.

Subgradient Optimisation method	$\boldsymbol{\lambda}^{(0)}$	$\{\rho_k\}_{k=0}^{\infty}$	Average computation time per iteration (sec)		
			Total	**For solving SSP**	**For solving NRP**
pure	0	$\rho_k = \frac{0.5}{k+1}$	5.42	4.98	0.44
pure	0	$\rho_k = \frac{0.5}{k+2}$	4.63	4.25	0.38
pure	0	$\rho_k = \frac{0.25}{k+1}$	4.69	4.32	0.37
pure	0	$\rho_k = \frac{0.25}{k+1.5}$	4.62	4.25	0.37
pure	0	$\rho_k = \frac{0.25}{k+3}$	2.54	2.14	0.40
pure	0	$\rho_k = \frac{0.7}{k+1.4}$	5.13	4.73	0.40
MGT	0	$\rho_k = \frac{0.5}{k+1}$	15.91	15.57	0.34
MGT	0	$\rho_k = \frac{0.5}{k+2}$	2.94	2.56	0.39
MGT	0	$\rho_k = \frac{0.25}{k+1}$	5.44	5.10	0.34
MGT	0	$\rho_k = \frac{0.25}{k+1.5}$	4.07	3.70	0.37
MGT	0	$\rho_k = \frac{0.25}{k+3}$	4.39	4.03	0.36
MGT	0	$\rho_k = \frac{0.7}{k+1.4}$	6.36	6.00	0.36
ADS	0	$\rho_k = \frac{0.5}{k+1}$	11.12	10.76	0.36
ADS	0	$\rho_k = \frac{0.5}{k+2}$	2.49	2.19	0.30
ADS	0	$\rho_k = \frac{0.25}{k+1}$	7.27	6.99	0.28
ADS	0	$\rho_k = \frac{0.25}{k+1.5}$	3.69	3.43	0.26
ADS	0	$\rho_k = \frac{0.25}{k+3}$	2.84	2.56	0.28
ADS	0	$\rho_k = \frac{0.7}{k+1.4}$	4.10	3.83	0.27
pure	-9	$\rho_k = \frac{0.25}{k+1.5}$	1.23	0.91	0.32
pure	-30	$\rho_k = \frac{0.7}{k+1.4}$	1.35	0.96	0.39
MGT	-9	$\rho_k = \frac{0.25}{k+1.5}$	3.48	3.13	0.35
MGT	-30	$\rho_k = \frac{0.7}{k+1.4}$	4.10	3.79	0.31
ADS	-9	$\rho_k = \frac{0.25}{k+1.5}$	2.03	1.72	0.31
ADS	-30	$\rho_k = \frac{0.7}{k+1.4}$	6.88	6.61	0.27

B Results for modified gradient technique and average direction strategy

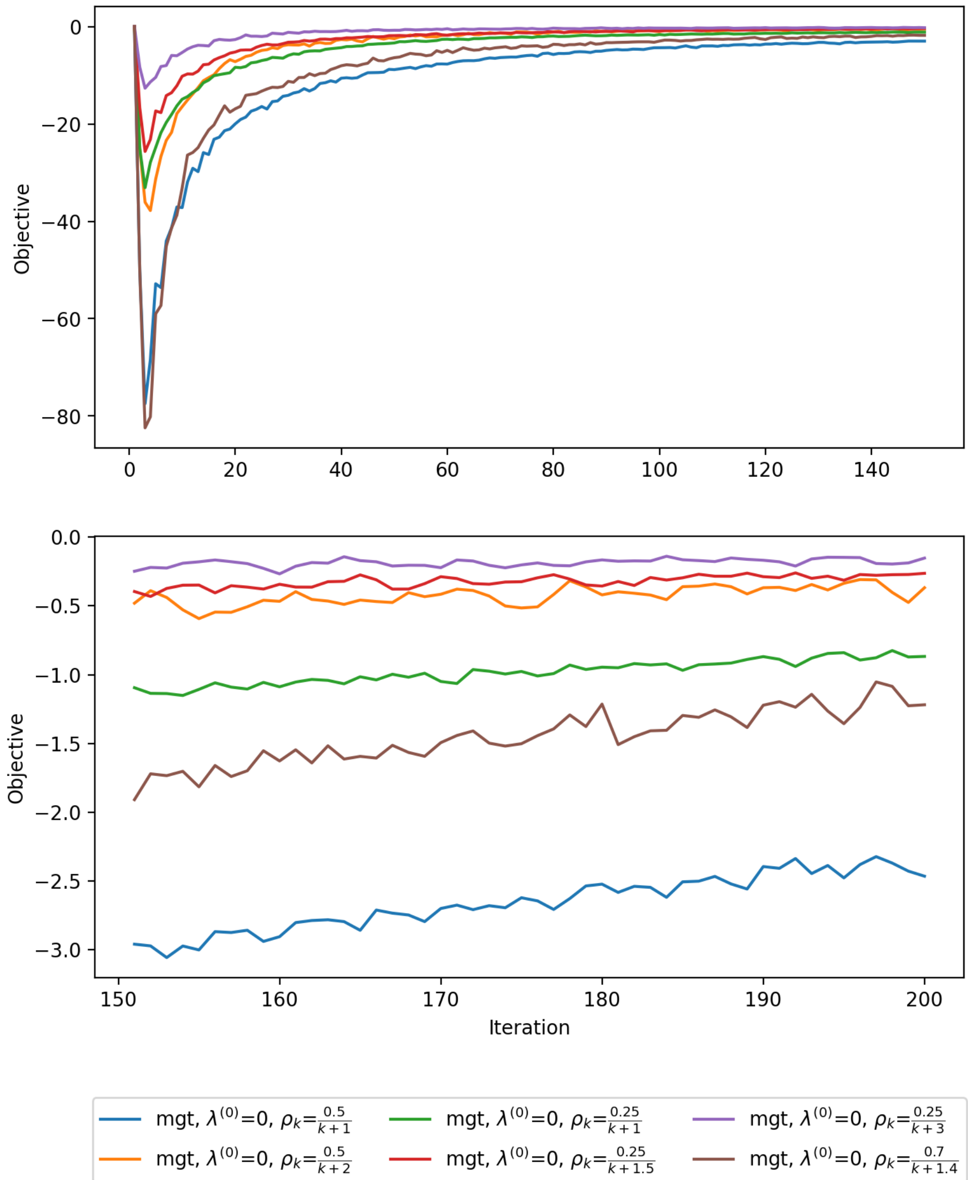

Figure 16: Lower bound to the objective of the TSP of data set 1 after each iteration. The MGT method is used and the multipliers are initialized at $\boldsymbol{\lambda}^{(0)} = 0$. Different definitions of sequence $\{\rho_k\}_{k=0}^{\infty}$ are used. For readability, separate graphs are made for the first 150 iterations and the final 50.

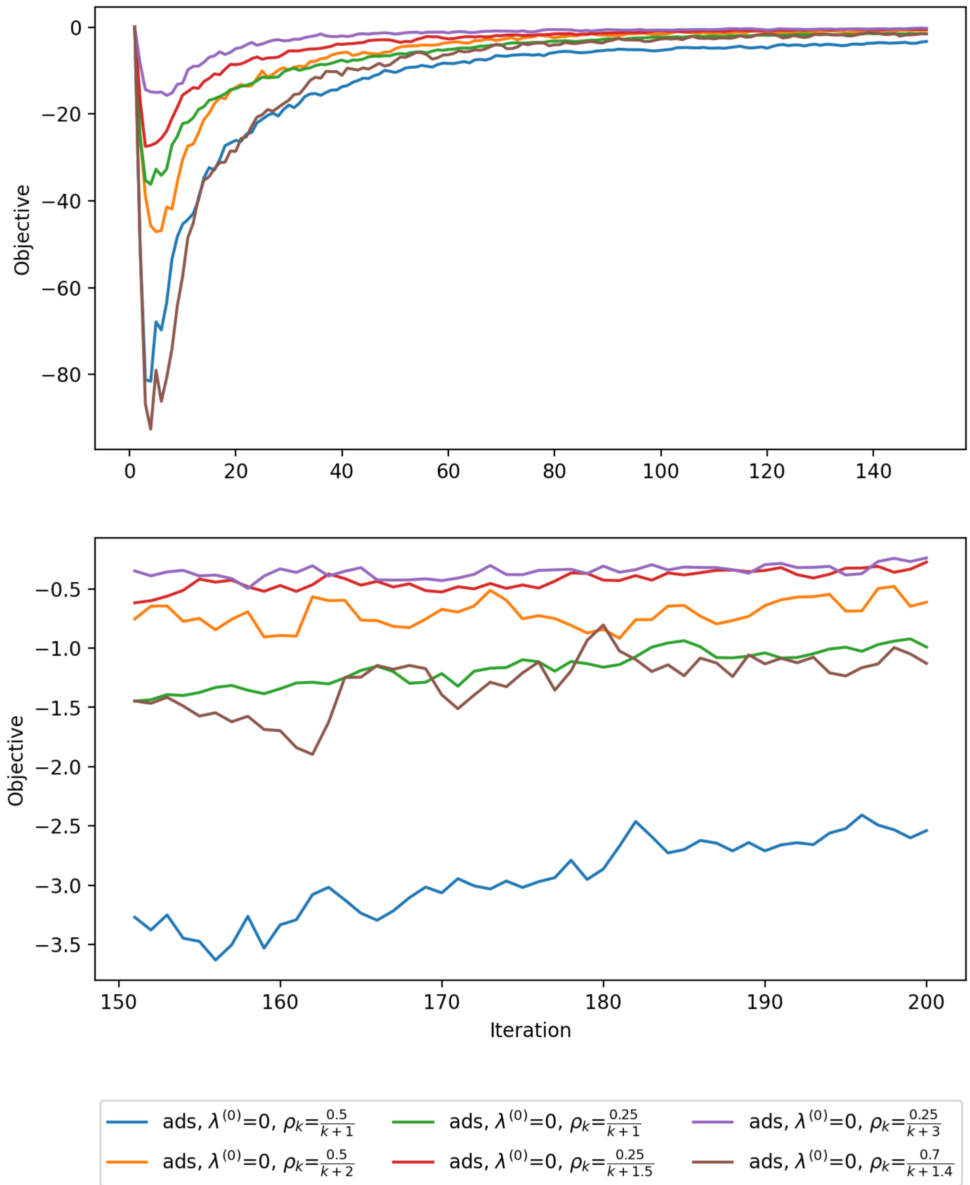

Figure 17: Lower bound to the objective of the TSP of data set 1 after each iteration. The ADS method is used and the multipliers are initialized at $\boldsymbol{\lambda}^{(0)} = 0$. Different definitions of sequence $\{\rho_k\}_{k=0}^{\infty}$ are used. For readability, separate graphs are made for the first 150 iterations and the final 50.