A two-phase approach to solve the nurse rostering problem for large-size instances

by

Laurien Fortuin - 543813

Supervisor:	prof. dr. A.P.M. Wagelmans
Second Assessor:	dr. R.M. Badenbroek
Company Supervisor:	J. Meijer
Date final version:	Thursday 14th April, 2022

Master thesis Operations Research & Quantitative Logistics

ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Abstract

In this thesis the Nurse Rostering Problem (NRP) is considered. The problem is to assign employees to an anonymous roster. The aim is to solve the NRP for large size instances. To solve the NRP for large size instances, a decomposition of the problem is proposed. The NRP is decomposed in two subproblems. The first subproblem assigns employees to working days and the second subproblem assigns employees to shifts on their working days. To solve the decomposed NRP, three methods are proposed. The first method solves the subproblems sequentially exact. We add extensions to the first subproblem. These extensions help the first subproblem taking some constraints of the second subproblem partly into account. Method 2 is a Harmony Search Algorithm (HSA). The HSA is initialised with a pool of solutions to the first subproblem by a construction heuristic. In each iteration of the HSA, a new solution to the first subproblem is generated. The generation of a new solution is done by either the random consideration, memory consideration or pitch consideration. The quality of the new solution is determined by a fitness function based on the second subproblem. The three fitness functions are the exact, relaxation and heuristic fitness function. The last method is a Hybrid Harmony Search Algorithm (HHSA), this algorithm is similar to the HSA but Variable Neighbourhood Search (VNS) is added. After a new solution is generated, VNS is performed to further improve the solution and escape local minima. The three methods are examined against benchmark instances. Our research shows that none of the methods are capable of finding good feasible solutions to the NRP. The results give insights in the proposed decomposition and methods. The results show that it is more complex to solve the first subproblem than the second subproblem. Therefore, the structure of the HSA suits the structure of the decomposed NRP. The addition of extensions to the first subproblem is a promising method to resolve the loss of information because of the decomposition. From the second method we can derive that the relaxation fitness function gives a good indication about the exact objective value of a solution. Furthermore, the HHSA performs worse than the HSA. This is because of the inability of the VNS to find better solutions in reasonable time.

Contents

1

Introduction

The shift scheduling problem is the problem of assigning work shifts to employees for a given planning horizon. A shift is a time window of working hours of which the start time and duration are given. Every shift can be performed by multiple employees. In some cases the employee needs a specific skill to perform the shift. To assign employees to a shift, regulations, laws and preferences of the employees have to be taken into account during scheduling. Also employee specific contract agreements, like working hours are of importance during scheduling. The shift scheduling problem is very complex and therefore widely studied in the literature. In the literature two different shift scheduling problems are distinguished. The first problem is the problem we consider in this thesis: the assignment of employees to shifts. This process, rostering or sometimes called scheduling, will be referred to as rostering in this thesis. The other shift scheduling problem is the problem of determining when shifts take place and when possible break during these shifts take place. In this thesis we refer to this problem as the anonymous rostering problem. There is a wide variety of rostering problems, for example airline crew rostering and call center rostering. The rostering problem as considered in this thesis is the nurse rostering problem (NRP). The goal of rostering is to generate a roster in such a way that no regulations are violated and that the preferred cover level is met. The preferred cover level is the number of nurses that are needed at a specific time. To meet the preferred cover level at all times, an anonymous schedule is created for the scheduling period. Generating an anonymous schedule is called the anonymous rostering problem. The anonymous schedule can for example be created by an algorithm or manually. This anonymous schedule contains shifts that are not assigned to employees. Per shift the number of employees that has to work the shift is known. This number of employees is related to the demand of nurses. The demand is often forecasted by the use of historical data. In this thesis only the assignment of nurses to this anonymous roster is considered; the demand forecasting and generation of the anonymous roster are out of scope.

Some NRP's need to be solved for a long planning horizon, for example when it is stated in the law that a roster has to be provided a year in advance. Most NRPs are $\mathcal{NP}$-hard problems (Lau, 1996, Osogami and Imai, 2000). This means that the difficulty of the problem and the size of the problem are exponentially related and the problem is solvable in non-deterministic polynomial time. Therefore, the NRP is difficult to solve for large instances. We consider an instance as a large instance if the instance contains more than 100 employees and a planning horizon of more than 3 months. This thesis aims to robustly solve these large-scale NRP's. Robustly solve means it takes a reasonable amount of time to compute a solution of reasonable quality. We consider a solution to be of reasonable quality if the worst solution of a certain set of problem instances has a optimality gap of 80 percent. This means the worst case solution has an objective value that is no more than 20 percent higher than of the optimal solution. For a planning horizon of a year, a couple of hours computation time is reasonable.

The NRP we want to solve in this thesis, is the NRP of Curtois and Qu (2014). This NRP is an $\mathcal{NP}$-hard problem (Rahimian et al., 2017). In this NRP, regulations, laws and preferences are taken into account in eight different rules. One of these rules is that employees are only allowed to work one shift per day. Other rules take the minimum and maximum consecutive shifts or days-off into account. Also the minimum and maximum working hours are taken into account. In the health care sector, rest after shifts is of great importance. Therefore, there is a rule that some shifts are not allowed to be worked after a specific shift. For example, a night shift cannot be followed by a morning shift. The aim of the NRP is to maximise the allocation of employees to shifts and to take employees' preferences to work or not work a shift into account. In Chapter 2, the NRP of this thesis is explained in full detail. The aim of this thesis is to robustly solve the NRP of Curtois and Qu (2014) for large-size instances.

1.1. Literature review

The nurse rostering problem is a widely studied area in the literature. One of the first mentions of the NRP is by Warner (1976). Warner (1976) divided the NRP into three main decisions. The first being the staffing decision, this includes how many nurses with which skills are needed in the coming period. Mostly, this is over a longer period, such as a year but the number of nurses needed may fluctuate over the period. The second decision is the scheduling, so the determination who is working when. The last decision is the re-scheduling of the nurses according to last minute changes.

After the contribution of Warner (1976), more research on the NRP was carried out. A lot of different variants of the NRP were studied and solved by different methods over the past 45 years. The use of heuristics and metaheuristics to find solutions of the NRP increased in the past years since (meta)heuristics were capable of solving complex problems. E. K. Burke et al. (2008) and Tassopoulos et al. (2015) found feasible solutions of the NRP using Variable Neighbourhood Search (VNS). E. K. Burke et al. (2008) combined the VNS with a heuristic ordering to assign nurses to shifts. In this combination, an initial solution was generated first by the ordering heuristic. The ordering heuristic ordered the shifts from the highest estimated difficulty to assign to the lowest estimated difficulty to assign. The nurse were assigned at first to the hardest shifts by the heuristic. The heuristic assigned nurses to the shifts if the penalty was the lowest for the nurse to work that shift. The schedule was improved by using VNS. To further improve the schedule, the shifts of the nurses with the highest penalty were all unassigned to these nurses. The unassigned shifts were then reassigned to nurses by the ordering heuristic. This process was repeated until the VNS terminated. Tassopoulos et al. (2015) used a two-phase stochastic variable neighbourhood approach. First, random working days are assigned to employees according to the demand that day. In the second stage, employees are randomly assigned to shifts and different swaps are performed to improve the solution. The swaps are between different days in the roster and per different neighbourhood, a different kind of swap is performed. For example, for one employee a working day is changed to a day off or a random day is swapped with another random day for one employee. Also E. Burke et al. (2003), proposed a VNS approach to find a feasible solution of the NRP and states that the proposed algorithm finds solutions of good quality in a short calculation time. More local search heuristics such as Simulated Annealing (SA) (Liu et al., 2018), and Tabu search (E. Burke et al., 1998) were investigated in the literature to solve the NRP. E. Burke et al. (1998) used a hybrid Tabu search. Other proposed methods were Ejection Chain (E. K. Burke & Curtois, 2014), Hybrid Artificial Bee Colony (HABC) algorithm (Awadallah et al., 2015) and Integer programming (Santos et al., 2016). An Artificial Bee Colony (ABC) algorithm is a swarm population based metaheuristic algorithm based on the behaviour of bees. The algorithm is based on three groups of bees: the employed bees, the onlooker bees and the scout bees. The ABC algorithm is initialised with memory of random solutions. After the initialisation, employed bees modify the solutions in the memory to find solutions with more nectar. The amount of nectar is a metaphor for the quality of the solution. Now, onlooker bees select probabilistically a solution from the solutions of the employed bees. The onlooker bees modify this solution further to find a better solution. The last group of bees, the scout bees, search for a random new solution for each solution from memory that could not be improved by the employed bees. All new solutions of the onlooker bees and scout bees are memorised in the memory and the best solution until so far is saved. If the stopping criteria is not met, the process of the three groups are going to look for better solutions again. Liu et al. (2018) compared SA with HABC and the SA heuristic performed better than HABC for the large scale problems.

Genetic Algorithms (GA) are frequently used to find a solution for the NRP. One of the first who presented the GA to solve the NRP was Aickelin and Dowsland (2000). Kim et al. (2014) further improved the GA and presented a better algorithm to find a feasible solution for the NRP than with the traditional GA. Bai et al. (2010) proposed a hybrid GA. The proposed algorithm is a combination of GA and a simulated annealing hyper heuristic. The proposed algorithm was evaluated with real-life instances of a large hospital in the UK. Each instance had a planning horizon of a week and per week 30 nurses had to be assigned to shifts. The hybrid GA demonstrated a high-quality performance compared to other methods. Zhuo et al. (2015) also proposed a hybrid genetic algorithm. The research combined a Artificial Bee Colony Algorithm with a hybrid algorithm of GA and Integer Programming to obtain good results for a Chinese NRP. This Chinese NRP had to assign up to 30 nurses for a planning horizon of one month. The constraints of the Chinese NRP were similar to the constraints of the NRP of this thesis. Zhuo et al. (2015) concluded that for instances of this size, the hybrid GA outperformed other methods. Hadwan et al. (2013) proposed a Harmony Search Algorithm (HSA). HSA is based on a harmony and their possibility to perform a piece of music. Each piece of music is a solution of the NRP. The harmony has an initial memory, called the Harmony Memory, which is an initial solutions pool. The goal of the algorithm is to improve the quality of the solutions by execute one of the following two actions: generate a random new solution or combine solutions of the solution pool to a new solutions. In some case, the combined solution is adjusted. After each new solution, the quality of the new solution is determined and if this solution is better than the worst solution of the solution pool, the worst solution is replaced by the new solution. This process is repeated until the stopping criteria is met. The proposed algorithm performed better than a standard GA but the proposed algorithm was not examined against instances with more than 46 nurses for 28 days. Awadallah et al. (2017) proposed a hybrid approach of the HSA. This algorithm performed similar to other methods for large instances, such as the proposed method of Tassopoulos et al. (2015). The largest instance was an instance of 50 nurses for a planning horizon of 28 days. The rules that Tassopoulos et al. (2015) considered in the NRP are similar to the NRP as considered in this thesis. Hybrid HSAs are also used to solve other large-scale $\mathcal{NP}$-hard problems, Gong et al. (2021) combined a HSA with Tabu search to solve a large-scale dynamic parallel row ordering problem. The proposed algorithm performed similar to other methods to solve large-scale problems of the dynamic parallel row ordering problem.

Besides (meta)heuristics, column Generation is a widely used method to solve the NRP. An early contribution about column generation was made by Jaumard et al. (1998). E. K. Burke and Curtois (2014) proposed two approaches: Ejection Chain and Branch and Price. The Branch and Price algorithm is a column generation based branch and bound method. By using column generation, each node of the branch and bound tree is solved. The pricing problem of both researches was used to generate new columns. Each column represented a schedule for individual employees. The pricing problem could be considered as a resource constraint shortest path problem, which was solved by using dynamic programming. The restricted master problem takes the constraints into account that involves the whole NRP. In the restricted master problem, the violations due to combinations of all individual schedules are minimised. For example, the deviation from the preferred cover level is minimised in the restricted master problem. Strandmark et al. (2020) researched the same NRP as we consider and was able to find solutions of large-size instances of this NRP by using a column generation-based heuristic. The restricted master problem was not solved exactly. To find near-optimal solutions of the restricted master problem quickly, a depth-first strategy is performed. The pricing problem was modelled as a resource constraint shortest path and solved with a two-phase dynamic programming method.

Two international nurse rostering competitions were held and these have contributed to a large number of available instances. The first international nurse rostering competition (INRC-I) aimed to stimulate further research in staff rostering (Haspeslagh et al., 2010). The rules of the INRC-I are similar to the rules considered in this thesis. Consecutive working days, days-offs and weekend days were treated as soft constraints. This NRP had two hard constraints: 1) the regulation that a nurse can only work one shift per day, and 2) that the demand has to be covered at all times. The largest instance that was used in the competition has 50 nurses for a planning horizon of 28 days. The problem of the second international nurse rostering competition (INRC-II) differed from the problem of the INRC-I (Ceschia et al., 2015). In the INRC-II the NRP was multi-stage, where in the INRC-I the NRP had to be solved in one stage. The INRC-II had to be solved per stage where the current stage influences the next stage. Every stage was one week. The largest instance was a NRP with a planning horizon of 8 weeks for 120 nurses. The constraints were similar to the INRC-I and the NRP in this thesis, except that in INRC-II required skills of nurses were taken into account during the assignment of nurses to shifts. The cover constraint was more flexible than the INRC-I since the coverage level was formulated as a minimum level. Next to these two competitions, a set of benchmark instances were published by Curtois (n.d.). The corresponding NRP was similar to the problem of INRC-I. This NRP is considered in this thesis. The largest instance of these benchmark instances is an instance with 50 weeks for 150 nurses.

To reduce the complexity of solving the NRP, decomposition of the NRP into subproblems is researched. An early multi-phase algorithm approach was proposed by Baxter and Mosby (1988). The heuristic first assigns persons to working days in one week. On the assigned working days, the persons are assigned to shifts per week. In the last stage, a desired rotation of weekly rosters is generated and assessed. The rotational schedule is a schedule in which for example, employee A works roster 1 in the first week and employee B works roster 2. The next week, employee A works roster 2 and employee B works roster 3.

Brucker et al. (2010) decomposed the NRP into a phase where sequential constraints are considered and a phase where the other constraints are considered. Sequential constraints are constraints that impose rules about sequential shifts, for example the number of consecutive shifts. By only considering the sequential constraints, individual sequences of shifts were constructed by a heuristic. The second phase constructed the schedules of the nurses and the total roster based on the individual sequence of shifts and the schedule and roster constraints. The solution of the two-stage construction algorithm was improved by using local search. The research showed that this proposed algorithm obtained reasonable results for the instances.

Valouxis et al. (2012) also decomposed the NRP into two different phases. In the first phase the working and free days per week were determined. During the second stage, nurses were assigned to shifts on the determined working days. Both subproblems were solved by solving an integer programming problem. An additional local search for the first phase was introduced to improve the solution. The proposed algorithm performed well on the instances of the INRC-I, the approach outperformed all other approaches.

A two-phase algorithm was also proposed by Solos et al. (2013). They proposed a two-phase variable neighbourhood search algorithm. The first phase randomly assigned nurses to working days and improved this solution with swaps in the roster. The same procedure was followed for the assignment of shifts. For the largest instances (26 and 29 days, 46 employees), the proposed algorithm outperformed the algorithm of Brucker et al. (2010). Tassopoulos et al. (2015) extended this two-phase algorithm by introducing more swap techniques. The results showed that the proposed two-phase algorithm outperformed the same one-phase algorithm and the approach presented by Solos et al. (2013).

Abdelghany et al. (2021) proposed a new two-stage variable neighbourhood search algorithm for the same NRP as the NRP of Curtois and Qu (2014). In the first phase only the most important soft constraints were considered and in the second stage the NRP was optimised by taking all the soft constraints into account. The problem was examined with a runtime of one hour for large-size instances but performed poorly due to long computation times in the first phase. When the runtime was extended, better results were obtained.

Most of the decompositions of the NRP are by the types of the constraints. As described above, the constraints were decomposed by importance, type or nature. The proposed decompositions were examined to different instances but only the instances examined by Abdelghany et al. (2021) had the size we want to solve in this thesis. The other decomposition methods were examined for smaller instances. Also, no complete research on the effect of the proposed decomposition was carried out. Smet (2018) investigated which constraints had the most significant influence on the complexity and hardness of the solvability of the NRP. This research showed that constraints about consecutive working shifts had a significant influence on the solvability of the NRP. The research of Smet (2018) and the research of Valouxis et al. (2012), Solos et al. (2013) and Tassopoulos et al. (2015) show that decomposing the NRP by type of the constraints is promising to reduce the complexity of solving the NRP. In the literature, almost no comprehensive studies on solving the NRP for large-size instances have been carried out. Strandmark et al. (2020) found a feasible solution of good quality for large-size instances of the NRP but the implementation of this research is very complex. Therefore, this method is not warranted for more general NRP's.

Both hybrid genetic algorithm (HGA) and hybrid harmony search algorithm (HHSA) showed promising results for the NRP. Both methods have not been not examined for the large instances that are examined in this thesis. The HHSA proposed by Awadallah et al. (2017) showed promising results for the largest instance of the INRC-I (50 nurses, 28 days). Also, the HGA of Zhuo et al. (2015) showed good results for approximately the same instance size as Awadallah et al. (2017) with similar constraints. Hadwan et al. (2013) stated that the HSA outperformed the GA for instance of 46 nurses for 28 days. Furthermore, the research of Gong et al. (2021) shows that a HHSA is promising for a large-scale $\mathcal{NP}$-hard problem.

The literature shows that both decomposing of the NRP and a HHSA are promising methods to solve the NRP. In literature not all of these methods have been examined for large-size instances of the NRP but they look promising for smaller size instances. Furthermore, HHSA is promising for another large-scale $\mathcal{NP}$-hard problem. Since the decomposing methods are promising, we decompose the NRP into two phase. The aim of this decomposition is to reduce the complexity of solving the NRP. The first phase assigns the employees to working days and days-off. The second phase assigns employees to shifts on working days. By decomposing the problem, we have two subproblems that are each less difficult than solving the problem as a whole. If each subproblem can be solved easily, the running time of solving the NRP is reduced. In this thesis, the first phase is called the days-on/off assignment subproblem and the second phase the shift assignment subproblem. The solution of the days-on/off assignment problem is the input for the shift assignment subproblem. The solution of the second subproblem is also the solution for the entire NRP. Different combinations of methods are used to find a solution for both subproblems. In this thesis we search for solutions of the NRP by using three different methods:

1. Both subproblems are solved exactly sequentially by using a MIP-solver. The MIP-solver is a python package for modelling and solving Mixed-Integer Linear Programs. To reduce the symmetry in the first subproblem, some extensions are added to the first subproblem.
2. The NRP is solved by using a HSA. Feasible solutions for the first subproblem are generated randomly by a construction heuristic. For each solution of the first subproblem, the quality of each solution is determined by solving the second subproblem with the first subproblem as input. The solution of the second subproblem is also the solution to the entire NRP. The quality of the second subproblem is determined by solving the second subproblem exactly, by solving a relaxation of the second subproblem or by solving the second subproblem using a construction heuristic.
3. We look for a solution of the NRP by using a HHSA. We find a solution in the same way as the HSA but we also perform a VNS to improve the solutions of the second subproblem. In the HHSA, we determine the quality of each solution by solving the second subproblem exactly, by solving a relaxation of the second subproblem or by solving the second subproblem using a construction heuristic.

1.2. Research questions

The goal of this thesis is to robustly solve large-size instances of the NRP. To research this goal, the main research question is:

Which method to solve the decomposed NRP gives the best trade-off between the solving time and the quality of the solution?

Besides this question, we answer the following two questions in this thesis:

1. Is the solving time of the NRP reduced by decomposing the NRP into a days-on/off assignment subproblem and shift assignment subproblem?
2. When solving the decomposed NRP, is the value of the solution of the NRP not more than 20 percent above the best known objective value?

In this thesis, the NRP according to Curtois and Qu (2014) is used. The regulations and preferences that are taken into account in the NRP are explained in Chapter 2. Also a MIP-formulation of the problem is given there. In Chapter 3, the MIP-formulations of the subproblems are given and the three methods to find solutions to the NRP are explained. Chapter 4 describes the results for the three different methods that have been used. The conclusion and discussion can be found in Chapter 5.

2
Problem Description

In this thesis the NRP according to Curtois and Qu (2014) is used. The aim is to assign employees to the anonymous roster of shifts such that the preferred cover level is satisfied. An employee can be assigned to a shift if the regulations, laws, the preferences and the characteristics of the employee are met. Employees characteristics are for example the maximum and minimum working time. Employee preference can be a request to work or to not work on a specific day or shift. In some cases, the employee can only be assigned to a shift when the skills of the employee meet the skills that are necessary for executing the shift. In this thesis, we do not take the skill requirements into account. So every employee can work on every shift if the regulations, laws and preferences are met. There are different shift types; for example a night shift, morning shift and afternoon shift. The preferred cover level cannot always be met, for example when not enough employees are available. When fewer employees are assigned than desired, it is called undercoverage. In the health sector undercoverage is detrimental. The opposite, overcoverage, is achieved when more than necessary employees are assigned. This is less unfavourable than undercoverage. In this thesis, under- and overcoverage are both accepted, however undercoverage is highly undesirable. We stimulate to have no over- and undercoverage by penalising when this happens. When there are too few or too many employees, we apply a penalty to the objective function of the problem. When assigning employees to the roster, regulations and laws have to be taken into account. The regulations, laws and preferences that are taken into account in this thesis are:

1. Every employee can work a maximum of one shift per day.
2. Some shifts are not allowed to be followed after another shift. This is because of a compulsory rest time between shifts. For example, a night shift cannot be followed by a morning shift.
3. The employee works a maximum number of the same shift types per planning period. For example, a nurse works at most 5 night shifts per planning period.
4. The minimum and maximum working hours of an employee are defined per planning period. This is usually defined in employee contracts or labour legislation.
5. The employee has a maximum and minimum of consecutive shifts per planning horizon. The minimum of consecutive days prevents rapidly switching patterns like work-rest-work-rest. The maximum prevents the employee from working too many consecutive shifts.
6. The employee has at least a minimum number of consecutive days-off per planning horizon.
7. To minimise the work during a weekend, a maximum per planning horizon is set for work during the weekend. Either working at one or both days of the weekend counts as working during the weekend.
8. When a day-off is requested, the employee cannot work that day.

Rules 1 and rule 5 can be combined to a rule that we use in the days-on/off assignment subproblem. Since every employee can work a maximum of one shift per day and the maximum and minimum number of consecutive shifts are known, we can derive the minimum and maximum consecutive working days.

2.1. The MIP-formulation

The problem as described above can be formulated as a Mixed Integer Problem (MIP). The objective of this MIP is to allocate employees as much as possible to the shifts and to minimise the under- and overcoverage. The parameters of the formulation are given in Table 2.1. The decision variables of the MIP formulation are given in Table 2.2. Two important notes are that the planning horizon always starts on Monday and a shift is counted to take place on a day if it starts in that day. For example, a shift at day 3 starts at 23:00 and has a duration of three hours, we count this shift to day 3 although it also takes place during day 4.

Table 2.1: Parameters of the MIP formulation

Parameters	Explanation
I	set of employees.
h	number of days in the planning horizon.
D	set of days in the planning horizon $\{1, ..., h\}$.
W	set of weekends in the planning horizon $\{1, ..., \lfloor h/7 \rfloor\}$.
T	set of shift types.
R_t	set of shift types that cannot be assigned immediately after shift type t.
N_i	set of days that employee i cannot be assigned to a shift on.
l_t	length of the shift type t in minutes.
m_{it}^{max}	maximum number of shifts of type t that can be assigned to employee i.
b_i^{min}	minimum number of minutes that employee i must be assigned.
b_i^{max}	maximum number of minutes that employee i must be assigned.
c_i^{min}	minimum number of consecutive shifts that employee i can work.
c_i^{max}	maximum number of consecutive shifts that employee i can work.
o_i^{min}	minimum number of consecutive days off that employee i can be assigned.
a_i^{max}	maximum number of weekends that employee i can work.
q_{idt}	penalty if shift type t is not assigned to employee i on day d.
p_{idt}	penalty if shift type t is assigned to employee i on day d.
$u_d t$	preferred cover level of shift type t on day d.
v_{dt}^{min}	weight if below preferred cover level for shift type t on day d.
v_{dt}^{max}	weight if exceeding preferred cover level for shift type t on day d.

Table 2.2: Decision variables of the MIP formulation

Decision variable	Explanation
x_{idt}	1 if employee i is assigned to shift type t on day d ($i \in I, d \in D, t \in T$), 0 otherwise.
k_{iw}	1 if employee i works on weekend w ($i \in I, w \in W$), 0 otherwise.
y_{dt}	total number below the preferred cover level for shift type t on day d ($d \in D, t \in T$).
z_{dt}	total number above the preferred cover level for shift type t on day d ($d \in D, t \in T$).

The MIP formulation of Curtois and Qu (2014) is:

$$\text{min} \sum_{i\in I}\sum_{d\in D}\sum_{t\in T} q_{idt}\left(1-x_{idt}\right)+\sum_{i\in I}\sum_{d\in D}\sum_{t\in T} p_{idt}x_{idt}+\sum_{d\in D}\sum_{t\in T} y_{dt}v_{dt}^{\text{min}}+\sum_{d\in D}\sum_{t\in T} z_{dt}v_{dt}^{\text{max}}, \tag{2.1}$$

$$\text{s.t.} \sum_{t\in T} x_{idt} \leq 1, \qquad \forall\, i\in I, d\in D \tag{2.2}$$

$$x_{idt}+x_{i(d+1)u} \leq 1, \qquad \forall\, i\in I, d\in\{1\cdots h-1\}, t\in T, u\in R_t \tag{2.3}$$

$$\sum_{d\in D} x_{idt} \leq m_{it}^{max}, \qquad \forall\, i\in I, t\in T \tag{2.4}$$

$$b_i^{\text{min}} \leq \sum_{d\in D}\sum_{t\in T} l_t x_{idt} \leq b_i^{\text{max}}, \qquad \forall\, i\in I \tag{2.5}$$

$$\sum_{j=d}^{d+c_i^{\text{max}}}\sum_{t\in T} x_{ijt} \leq c_i^{\text{max}}, \qquad \forall\, i\in I, d\in\{1\cdots h-c_i^{\text{max}}\} \tag{2.6}$$

$$\sum_{t\in T} x_{idt}+\left(s-\sum_{j=d+1}^{d+s}\sum_{t\in T} x_{ijt}\right)+\sum_{t\in T} x_{i(d+s+1)t} \geq 1, \tag{2.7}$$

$$\forall\, i\in I, s\in\left\{1\cdots c_i^{\text{min}}-1\right\}, d\in\{1\cdots h-(s+1)\}$$

$$\left(1-\sum_{t\in T} x_{idt}\right)+\sum_{j=d+1}^{d+s}\sum_{t\in T} x_{ijt}+\left(1-\sum_{t\in T} x_{i(d+s+1)t}\right) \geq 1, \tag{2.8}$$

$$\forall\, i\in I, s\in\left\{1\cdots o_i^{\text{min}}-1\right\}, d\in\{1\cdots h-(s+1)\}$$

$$k_{iw} \leq \sum_{t\in T} x_{i(7w-1)t}+\sum_{t\in T} x_{i(7w)t} \leq 2k_{iw}, \quad \forall\, i\in I, w\in W \tag{2.9}$$

$$\sum_{w\in W} k_{iw} \leq a_i^{\text{max}}, \qquad \forall\, i\in I \tag{2.10}$$

$$x_{idt}=0, \qquad \forall\, i\in I, d\in N_i, t\in T \tag{2.11}$$

$$\sum_{i\in I} x_{idt}-z_{dt}+y_{dt}=u_{dt}, \qquad \forall\, d\in D, t\in T, \tag{2.12}$$

$$x_{idt}\in\mathbb{B}, \qquad \forall\, i\in I, d\in D, t\in T \tag{2.13}$$

$$k_{iw}\in\mathbb{B}, \qquad \forall\, i\in I, w\in W \tag{2.14}$$

$$y_{dt}\in\mathbb{N}_0, \qquad \forall\, d\in D, t\in T \tag{2.15}$$

$$z_{dt}\in\mathbb{N}_0. \qquad \forall\, d\in D, t\in T \tag{2.16}$$

The first constraint 2.2 makes sure each employee works at most one shift per day. Since rest time is necessary after some shifts, the second constraint 2.3 makes sure employees are assigned to a shift that is allowed to follow the previous shift. Some employees can or prefer to not work more than a given number of a specific shift type. This is ensured in constraint 2.4. To make sure an employee does not work more than the maximum or less than the minimum agreed working time, the constraint 2.5 exists. Constraints 2.6 and 2.7 make sure the maximum and minimum number of consecutive shifts are taken into account. The minimum number of consecutive shifts constraint (2.7) consists of three parts. The middle part of the constraint checks if employee i works s sequential shifts from day $d+1$ to day $d+s$, where s is 1 to the minimum number of consecutive shifts minus one. If so, the employee also has to work a shift at day d or day $d+s+1$ to ensure the employee works the minimum of consecutive shifts. It is important to notice that the constraint does not restrict to have a sequence of shifts that is more than c_i^{min} in the first $c_i^{\text{min}}-1$ days and last $c_i^{\text{min}}-1$ days of the planning horizon. This means that if c_i^{min} is two, we can start with one shifts and have some days off after this shift, so the first sequence of working is shorter than c_i^{min}. Constraint 2.8 ensures every employee has a minimum of consecutive days off. This constraint works the same as constraint 2.7 but then for days-off. This constraint also does not force to have more than o_i^{min} days off in the first and last days of the planning horizon. To ensure the number of weekends in which the employee has to work is set to a limit, the constraints 2.9 and 2.10 are included. In the problem we examine in this thesis, we count that an employee works the whole weekend even if he/she only works on Saturday or Sunday. When an employee requests a day off, that has to be granted, which is executed by constraint 2.11. The last constraint is the cover requirement constraint: in this constraint 2.12, the preferred cover level is met by the assignment of employees to shifts and under- and overcoverage. The objective (2.1) consists of two parts, the first two terms minimise the allocation of shifts to employees who requested days off or to work a certain shift. The third and fourth terms minimises under- and overcoverage.

2.2. Description of the data

To test the solution of the NRP, the solution can be compared with the results from benchmark instances. These benchmark instances are public and can be obtained from Curtois (n.d.). There are 24 benchmark instances of different sizes available. The smallest instance has a planning horizon of two weeks, eight employees and one shift type. For the largest instance, the NRP has to be solved for 52 weeks, 150 employees and 32 different shift types. In Table 2.3 the specifications of all benchmark instances are given. Also the best known objective value and lower bound are given. These solutions and lower bounds were found by other research. When the best known value is equal to the best known lower bound, the solution is optimal and in bold in the table. For the four largest instances and instance 15, no optimal solution has yet been found.

Every shift holds information about the shift ID, the length of the shift and the shifts that cannot take place after this shift. An anonymous shift roster is given for each day in the planning horizon. In this roster multiple shifts per day are scheduled. For every shift the unique shift IDs, the preferred cover level and the weights for over- and undercoverage are given. In this thesis the penalty for undercoverage is 100 and overcoverage is 1. For each of the employees their ID, maximum number of shifts per shift type and their maximum and minimum working minutes are given. Furthermore, for each employee the maximum and minimum consecutive shifts and maximum number of weekends are given. The employee can request to work a specific shift or to not work on a specific day or shift. The employee's ID, day, shift ID and weight to (not) assign this shift are given for each request. The preference weights have a value between 1 and 10. In Appendix A, benchmark instance 1 is included, to show how the data is presented.

Table 2.3: The specifications of the benchmark instances

Instance	Weeks	Employees	Shift types	Best known lower bound	Best known solution
1	2	8	1	607	**607**
2	2	14	2	828	**828**
3	2	20	3	1001	**1001**
4	4	10	2	1716	**1716**
5	4	16	2	1143	**1143**
6	4	18	3	1950	**1950**
7	4	20	3	1056	**1056**
8	4	30	4	1300	**1300**
9	4	36	4	439	**439**
10	4	40	5	4631	**4631**
11	4	50	6	3443	**3443**
12	4	60	10	4040	**4040**
13	4	120	18	1348	**1348**
14	6	32	4	1278	**1278**
15	6	45	6	3829	3832
16	8	20	3	3225	**3225**
17	8	32	4	5746	**5746**
18	12	22	3	4459	**4459**
19	12	40	5	3149	**3149**
20	26	50	6	4769	**4769**
21	26	100	8	21124	21159
22	52	50	10	28997	31279
23	52	100	16	16990	17428
24	52	150	32	26571	42463

3

Methodology

The NRP is an $\mathcal{NP}$-hard problem. To obtain a robust feasible solution to this problem for large-size instances, we decompose the NRP into two subproblems. The first subproblem assigns employees to either working days or days-off. The second subproblem assigns employees to shifts on their working days. We call the first subproblem the days-on/off assignment subproblem and the second subproblem the shift assignment subproblem. The purpose is to reduce the solution space of the second subproblem by determining the working days of each employee in the first subproblem. The aim is to reduce the running time of finding a feasible solution to the NRP.

In the first subproblem all the constraints that have influence on the working days and days-off are considered. For example, the number of consecutive working days. In the second subproblem, the shift constraints and the other constraints are handled. The rules of the NRP, as explained in Chapter 2, are divided into the two subproblems. Table 3.1 explains which rule is taken into account in which subproblem.

Table 3.1: The partitioning of the constraints per subproblem

Rule	Subproblem
The employee has a maximum and minimum of consecutive shifts	Days-on/off assignment
The minimum number of consecutive days-off per employee	Days-on/off assignment
The maximum number working at weekends	Days-on/off assignment
The assignment of requested days-off	Days-on/off assignment
The employee can work at most one shift per day	Shift assignment
The shift sequence constraint	Shift assignment
The maximum number of the same shift type per employee	Shift assignment
The minimum and maximum working time per employee	Shift assignment

The objective of the first subproblem is to assign working days to employees such that the deviations from the daily preferred cover level are minimised. The objective of the second subproblem is identical to the objective of the NRP: to minimise non-preferred allocation and minimise under- and overcoverage. The objective value of the shift assignment subproblem indicates the quality of the solution of the NRP. The MIPs of both subproblems are derived from the MIP of the NRP as explained in Chapter 2, see Section 3.1 and 3.2.

To find feasible solutions for the NRP, the different instances are solved using different methods. We use three different methods to find solutions for the NRP. Method 1 finds solutions by solving the first and the second subproblem sequentially using the MIP-solver. The second method is a Harmony Search Algorithm in which the solutions are solutions of the first subproblem and the quality of each solution is (an indication of the) objective value of the second subproblem. The (indication of) objective value is determined in three different ways: by solving the second subproblem exactly using the MIP-solver, by solving a relaxation of the second subproblem, or by solving the second subproblem by using a construction heuristic. The structure of the third method is the same as the second method only the third method is a Hybrid Harmony Search Algorithm (HHSA). In the HHSA we perform a VNS to escape the local minima that is why this is a Hybrid HSA. To compare the proposed methods, we can look at the best known objective values of the benchmark instances. Next to this, we solve the complete NRP problem with a maximum solving time using the MIP-solver. The results of this method, the compare method, can be compared with the results of the three methods. In Figure 3.1, a short overview of the three methods and the compare method are given. To explain the methods, we first discuss the both subproblems by their MIP-formulations. In Sections 3.3, 3.4 and 3.5 we elaborate on the three methods. In Section 3.6 the compare method is explained.

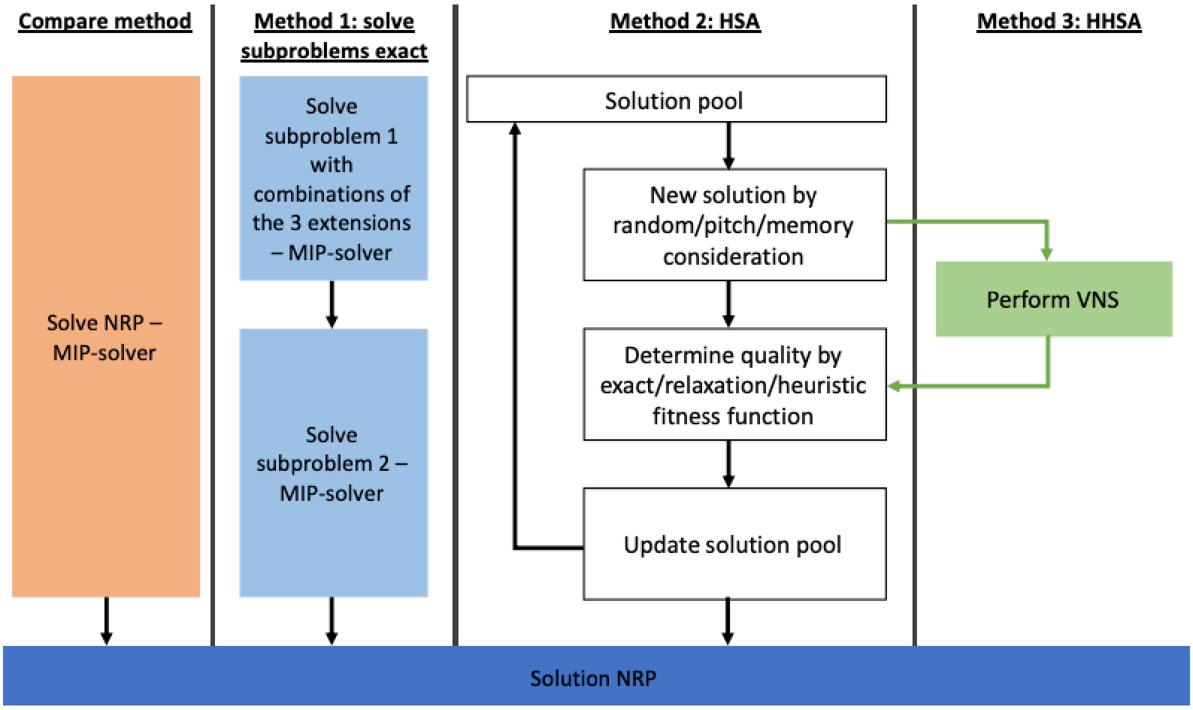


Figure 3.1: Overview of the methods

3.1. The MIP-formulation of the days-on/off assignment subproblem

The decomposition of the NRP into two subproblems is carried out by introducing the decision variable g_{id}. The decision variable g_{id} is a binary variable and has the following values:

$$g_{id} = \begin{cases} 1, & \text{if employee } i \ (i \in I) \text{ is assigned to work on day } d \ (d \in D). \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

The MIP-formulation of the days-off scheduling can be formulated as follows, with the parameters and decision variables given in Table 3.2 and 3.3 respectively:

Table 3.2: Parameters of the days-off MIP formulation

Parameters	Explanation
I	set of employees.
h	number of days in the planning horizon.
D	set of days in the planning horizon $\{1, ..., h\}$.
W	set of weekends in the planning horizon $\{1, ..., \lfloor h/7 \rfloor\}$.
N_i	set of days that employee i cannot be assigned to work.
c_i^{min}	minimum number of consecutive days that employee i can work.
c_i^{max}	maximum number of consecutive days that employee i can work.
o_i^{min}	minimum number of consecutive days off that employee i can be assigned.
a_i^{max}	maximum number of weekends that employee i can work.
f_d	preferred cover level of day d.
v_d^{min}	weight if below preferred cover level on day d.
v_d^{max}	weight if exceeding preferred cover level on day d.

Table 3.3: Decision variables of the MIP formulation

Decision variable	Explanation
g_{id}	1 if employee i is assigned to work on day d ($i \in I, d \in D$), 0 otherwise.
k_{iw}	1 if employee i works on weekend w ($i \in I, w \in W$), 0 otherwise.
y_d	total number below the preferred cover level on day d ($d \in D$).
z_d	total number above the preferred cover level on day d ($d \in D$).

$$\min \sum_{d \in D} y_d v_d^{\text{min}} + \sum_{d \in D} z_d v_d^{\text{max}}, \tag{3.2}$$

$$\text{s.t.} \sum_{j=d}^{d+c_i^{\text{max}}} g_{ij} \leq c_i^{\text{max}}, \qquad \forall i \in I, d \in \{1, ..., h - c_i^{\text{max}}\} \tag{3.3}$$

$$g_{id} + \left(s - \sum_{j=d+1}^{d+s} g_{ij} \right) + g_{i(d+s+1)} \geq 1, \quad \forall i \in I, s \in \{1, ..., c_i^{\text{min}} - 1\}, d \in \{1, ..., h - (s+1)\} \tag{3.4}$$

$$(1 - g_{id}) + \left(s - \sum_{j=d+1}^{d+s} g_{ij} \right) + (1 - g_{i(d+s+1)}) \geq 1, \tag{3.5}$$
$$\forall i \in I, s \in \{1, ..., o_i^{\text{min}} - 1\}, d \in \{1, ..., h - (s+1)\}$$

$$k_{iw} \leq g_{i(7w-1)} + g_{i(7w)} \leq 2k_{iw}, \qquad \forall\, i \in I, w \in W \tag{3.6}$$

$$\sum_{w \in W} k_{iw} \leq a_i^{\text{max}}, \qquad \forall\, i \in I \tag{3.7}$$

$$\sum_{i \in I} g_{id} - z_d + y_d = f_d, \qquad \forall\, d \in D \tag{3.8}$$

$$g_{id} \in \mathbb{B}. \qquad \forall\, i \in I, d \in D \setminus N_i \tag{3.9}$$

$$k_{iw} \in \mathbb{B}, \qquad \forall\, i \in I, w \in W \tag{3.10}$$

$$z_d \in \mathbb{N}_0. \qquad \forall\, d \in D \tag{3.11}$$

$$y_d \in \mathbb{N}_0. \qquad \forall\, d \in D \tag{3.12}$$

The objective of the days-on/off assignment is to minimise the under- and overcoverage on day-level. The first constraint 3.3 ensures that an employee can never work more than the maximum consecutive working days. The second constraint 3.4 ensures that an employee can never work less than the minimum consecutive working days. These constraints are derived from constraints 2.6 and 2.7. Since the employee can only work one shift per day, the maximum or minimum number of consecutive shifts is equal to the maximum or minimum number of consecutive working days. Similar to the minimum number of consecutive shifts and days off constraints of the complete NRP, the constraints do not force that the first $c_i^{\text{min}} - 1$ days and the first $o_i^{\text{min}} - 1$ days of the planning horizon are part of a sequence of working days or days off that is minimal c_i^{min} or o_i^{min} long. Constraint 3.5 makes sure that employee i has at least o_i^{min} consecutive days off. Constraint 3.6 checks whether the employee works on one of or both weekend days, whereas constraint 3.7 guarantees the employee does not work more than a_i^{max} weekends. The last constraint (3.8) defines the deviations of the preferred cover level of each day. The deviation of the preferred cover level is the level of under- and overcoverage. The preferred cover level of day d, f_d, is calculated by adding up the cover level of all the shifts of any type that starts during that day, d. So $f_d = \sum_{t \in T} u_{dt}$, where T is the set of shift types and u_{dt} is the preferred cover level of shift type t on day d. The preferred cover level can be added up easily because we do not take skills of employees into account under the assumption that every employee can work every shift. To ensure that requested days off are granted, variable g_{id} only exists at day d if employee i did not requested a day off on day d.

3.2. The MIP-formulation of the shift assignment subproblem

The MIP of the shift assignment subproblem is formulated below and the parameters and decision variables are explained in Tables 3.4 and 3.5.

Table 3.4: Parameters of the assignment MIP formulation

Parameters	Explanation
I	set of employees.
h	number of days in the planning horizon.
D	set of days in the planning horizon $\{1, ..., h\}$.
G_i	set of working days of employee i in the planning horizon.
DG_i	set of days d for which holds that d and $d+1$ are a working day of employee i; $d \in G_i$ and $d+1 \in G_i$.
T	set of shift types.
R_t	set of shift types that cannot be assigned immediately after shift type t.
l_t	length of the shift type t in minutes.
m_{it}^{max}	maximum number of shifts of type t that can be assigned to employee i.
b_i^{min}	minimum number of minutes that employee i must be assigned.
b_i^{max}	maximum number of minutes that employee i must be assigned.
q_{idt}	penalty if shift type t is not assigned to employee i on day d.
p_{idt}	penalty if shift type t is assigned to employee i on day d.
$u_d t$	preferred cover level of shift type t on day d.
v_{dt}^{min}	weight if below preferred cover level for shift type t on day d.
v_{dt}^{max}	weight if exceeding preferred cover level for shift type t on day d.
g_{id}	indicator if employee i works a day d.

Table 3.5: Decision variables of the assignment MIP formulation

Decision variable	Explanation
x_{idt}	1 if employee i is assigned to shift type t on day d ($i \in I, d \in G_i$), 0 otherwise.
y_{dt}	total number below the preferred cover level for shift type t on day d ($d \in D, t \in T$).
z_{dt}	total number above the preferred cover level for shift type t on day d ($d \in D, t \in T$).

The MIP of the shift assignment subproblem is:

$$\min \sum_{i\in I}\sum_{d\in G_i}\sum_{t\in T} p_{idt}x_{idt} + \sum_{d\in D}\sum_{t\in T} y_{dt}v_{dt}^{\text{min}} + \sum_{d\in D}\sum_{t\in T} z_{dt}v_{dt}^{\text{max}}, \tag{3.13}$$

$$x_{idt} + x_{i(d+1)u} \leq 1, \qquad \forall\, i \in I, d \in DG_i, t \in T, u \in R_t \tag{3.14}$$

$$\sum_{d\in G_i} x_{idt} \leq m_{it}^{max}, \qquad \forall\, i \in I, t \in T \tag{3.15}$$

$$b_i^{\text{min}} \leq \sum_{d\in G_i}\sum_{t\in T} l_t x_{idt} \leq b_i^{\text{max}}, \qquad \forall\, i \in I \tag{3.16}$$

$$\sum_{i\in I: d\in G_i} x_{idt} - z_{dt} + y_{dt} = u_{dt}, \qquad \forall\, d \in D, t \in T, \tag{3.17}$$

$$\sum_{t\in T} x_{idt} = 1, \qquad \forall\, i \in I, d \in G_i \tag{3.18}$$

$$x_{idt} \in \mathbb{B}, \qquad \forall\, i \in I, d \in G_i, t \in T \tag{3.19}$$

$$y_{dt} \in \mathbb{N}_0, \qquad \forall\, d \in D, t \in T \tag{3.20}$$

$$z_{dt} \in \mathbb{N}_0. \qquad \forall\, d \in D, t \in T \tag{3.21}$$

The formulation of the shift assignment is very similar to the MIP of the nurse rostering problem as given by Equations 2.1 to 2.16. The constraints that are taken into account during the days-on/off assignment are removed and the objective is adjusted compared to the objective of the whole NRP formulation (2.1). The removed constraints are the constraints of the minimum (2.7) and maximum (2.6) consecutive working days, the minimum consecutive days-off (2.8), working on weekend days (2.9 and 2.10) and about requested days-off (2.11). Constraint 3.18 is derived from constraint 2.2 such that at most one shift is worked per day. Furthermore, this constraint is necessary to not violate the constraints of the first subproblem. The constraint ensures that the employee can only work one shift per day on the day that the employee is assigned to have a working day. This constraint makes sure that the second subproblem uses the solution of the first subproblem as input and does not change the working days or days-off. The objective of the second subproblem differs from the objective of the whole NRP formulation in two ways. First, the first term of the whole NRP formulation is removed. The first term determines the penalty of not working shifts that have the employees' preference. Since the first subproblem determines if the employee works on a day, we can determine the penalty of not working preferred shifts from the solution of the first subproblem. The term is neglected in the objective function of second subproblem since the term cannot be minimised in this subproblem. To compare the obtained objective value of the second subproblem with the objective values of the solutions of the NRP, the outcome of $\sum_{i\in I}\sum_{d\in O_i}\sum_{t\in T} q_{idt}$ is added to the objective value of the second subproblem. In this term O_i is the set of days-off of employee i and q_{idt} is the penalty of not working shift t on day d by employee i. The other adjustment in the objective term is the change from $\sum_{i\in I}\sum_{d\in D}\sum_{t\in T} p_{idt}x_{idt}$ to $\sum_{i\in I}\sum_{d\in G_i}\sum_{t\in T} p_{idt}x_{idt}$. We sum $\sum_{t\in T} p_{idt}x_{idt}$ over the working days of employee i (G_i) instead of all the days as we do in the whole NRP formulation.

Now, the two subproblems and their MIP-formulations are known. With this two subproblems in mind, we develop the following three methods to find reasonable solutions to the NRP. First, the three methods are explained. Subsequently, the compare method is explained, see Section 3.6.

3.3. Method 1: Solving the subproblems sequentially exactly

To evaluate the quality and solving time of the decomposed NRP, the two subproblems are sequentially solved exactly by using a MIP-solver. This method, method 1, uses the optimal solution of the first subproblem as input in the second subproblem. The second subproblem is also solved to optimality. An overview of this method is given in Figure 3.1. The results of method 1 can be compared to the results of the compare method and the benchmark instances. When an optimal solution of the days-on/off assignment is found, it is possible that the optimal solution of the second subproblem does not yield an optimal solution to the complete NRP due to the days-on/off assignment. A reason for this is that the solution of the days-on/off assignment has impact on the solution of the second subproblem but in the objective of first subproblem, no evaluation of the quality of the solution to the second phase is included. To improve the solution of the first subproblem such that the solution of the second subproblem is improved, several extension can be applied. We used three extension to take the second subproblem partly into account in the first subproblem. The first extension includes a number of desired working days into the objective function by taking the contract hours into account during the assignment of working days. The second extension takes the preferences to work or not work a shift on a given day into account into the objective function. The first two extensions are soft constraints. The last extension is a hard constraint and also takes the contract hours into account. The last extension adds an extra constraint to the first subproblem to obligate each employee a minimum and maximum of working days. Another reason to add the extensions to the first subproblem is to remove symmetry of the solution space. Margot (2010) describes that when variables can be permuted without changing the problem, symmetry occurs. Ostrowski et al. (2010) explains symmetry in the job scheduling problem. Ostrowski et al. (2010) also describes that the variables in this problem can be permuted and therefore symmetry occurs in the job scheduling problem. The NRP shows some overlap with the job scheduling problem and in the first subproblem we can permute the variable g_{id} at a day d without consequences between employees for which the following characteristics are equal: $c_i^{\text{max}}, c_i^{\text{max}}, c_i^{\text{min}}, a_i^{\text{max}}$. Therefore, when employees have the same characteristics, symmetry occurs. By adding the extensions, less variables can by permuted without consequences, which removes a part of the symmetry. Due to the extensions, more characteristics of the employees are taken into account during the first subproblem, which causes that fewer employees are the same. Furthermore, an employee can prefer to work a certain day instead of another employee, which make the corresponding variable not permuted. The extensions are explained in more detail in Section 3.3.1, 3.3.2 and 3.3.3

3.3.1. First extension: stimulate to work the number of required working days

We introduce a balancing element in the MIP formulation to stimulate a better solution of the first subproblem for the second subproblem. The balancing element ensures that employees are assigned to a number of days, proportional to their contract hours. For example, employee A has to work 40 hours per week according his/her contract and will be assigned to work more days than another employee who has an agreed working time of 10 hours per week. To stimulate a weighted assignment of the number of working days, we penalise the relative deviation of the assigned working days to the average desired number of working days. First, we introduce the parameter which denotes the average desired working time per employee, γ_i , this the average of the maximum contracted working time (b_i^{max}) and minimum contracted working time (b_i^{min}) in minutes. For each employee i the average desired working time is calculated by:

$$\gamma_i = \frac{1}{2}(b_i^{\text{max}} + b_i^{\text{min}}). \tag{3.22}$$

To determine the average desired number of working days, we divide the average desired working time by the average length of the shifts per day, $\bar{l}$. To evaluate the deviation from the average desired number of working days to the assigned working days, we introduce the parameter:

$$\xi_i = \frac{\gamma_i}{\bar{l}} \qquad \forall\ i \in I. \tag{3.23}$$

Where ξ_i is the average desired number of working days for employee i. By adding the following constraints to the formulation

$$\tau_i \geq \xi_i - \sum_{d \in D} g_{id}\ \forall\ \ i \in I \tag{3.24}$$

$$\tau_i \geq \sum_{d \in D} g_{id} - \xi_i\ \forall\ \ i \in I \tag{3.25}$$

$$\tag{3.26}$$

we derive the absolute total deviation from the average desired number of working days per employee. By adding

$$\beta \sum_{i \in I} \frac{\tau_i}{\xi_i} \tag{3.27}$$

to the objective of the days-on/off assignment subproblem, the deviation from the average desired number of working days is minimised. In Equation 3.27, the total relative average deviation of the average desired number of working days of all the employees is determined. β is the penalty of the deviation term in the objective.

3.3.2. Second extension: stimulate to grant the preferences of the employee

To stimulate a solution of the first subproblem that gives a good solution at the second subproblem, we incorporate the preferences of employees in the first subproblem. The preferences of an employee can be determined by the parameters p_{idt} and q_{idt}. q_{idt} penalises an objective value if shift type t is not assigned to employee i at day d. If $q_{idt} > 0$, employee i prefers to work shift type t at day d, and if the employee does not work the shift, this is penalised. The parameter p_{idt} is the opposite of q_{idt}. If $p_{idt} > 0$, employee i prefers to not work the shift of type t at day d. If the employee is assigned to a shift he/she does not want to work, this is penalised with p_{idt}. These parameters can be used to indicate if the employee wants to work on a given day. There are four scenarios:

1. Employee i wants to work a shift at day d, the employee has no shifts that he/she does not want to work. This means that on given day d for employee i : $\sum_{t \in T} q_{idt} > 0$ and $\sum_{t \in T} p_{idt} = 0$. In this scenario, we want to stimulate that employee i works on this day d.
2. Employee i has some preferences to work and not work shifts on given day d. The employee has a shift he/she does not want to work during this day and one or more shifts he/she not prefers to work during this day. This is indicated by $\sum_{t \in T} q_{idt} > 0$ and $\sum_{t \in T} p_{idt} > 0$. If the employee does not work at this day, a penalty is given because the preference of the employee to work a shift cannot be granted. Therefore, we want to stimulate that this employee works this day. During the second subproblem the preference of not willing to work a certain shift during this day is taken into account.
3. The employee i has no preference to work a shift on day d but has shifts he/she does not want to work at day d. So, $\sum_{t \in T} q_{idt} = 0$ and $\sum_{t \in T} p_{idt} > 0$ for day d and employee i. In this case, we stimulate that employee i does not work that day, d.
4. The employee i is indifferent to work or not work a shift at day d, so $\sum_{t \in T} q_{idt} = 0$ and $\sum_{t \in T} p_{idt} = 0$. In this case, the choice to work a day is also indifferent.

We want to stimulate that employee i works on day d in case of scenarios 1 and 2. In the case of scenario 3, we want to stimulate that that employee i does not work at day d. By introducing the parameter

$$\nu_{id} = \begin{cases} 1, & \text{if } \sum_{t \in T} q_{idt} > 0 \text{ \& } \sum_{t \in T} p_{idt} = 0 \text{ or} \sum_{t \in T} q_{idt} > 0 \text{ \& } \sum_{t \in T} p_{idt} > 0, \\ -1, & \text{if } \sum_{t \in T} q_{idt} = 0 \text{ \& } \sum_{t \in T} p_{idt} > 0, \\ 0 & \text{if } \sum_{t \in T} q_{idt} = 0 \text{ \& } \sum_{t \in T} p_{idt} = 0, \end{cases} \tag{3.28}$$

we add the term

$$-\alpha \sum_{i \in I} \sum_{d \in D} \nu_{id} g_{id} \tag{3.29}$$

to the objective function. This term stimulates that employee i works on day d in the case of scenario 1 or 2. In case of scenario 3, we stimulate that employee i does not work on day d. In the case of scenario 4, $\nu_{id} = 0$ so Equation 3.29 is also zero, so the choice if employee i works at day d is indifferent. α is the weight of taking preferences of employee i at day d into account.

3.3.3. Third extension: respect the required working hours per employee

The last extension is a hard constraint. This constraint stimulates to have a solution of the first subproblem that complies with the contract hours constraint of the second subproblem (3.16). The extension sets per employee a number of minimum and maximum working days. These minimum and maximum working days are determined by the minimum and maximum contract hours of the employee and the longest and shortest shift. The minimum number of working days is calculated by the minimum working hours, b_i^{min}, divided by the longest shift, l^{max}. The maximum number of working days is determined by the maximum working hours, b_i^{max}, divided by the shortest shift, l^{min}. The constraint we add to the first subproblem as extension is:

$$\frac{b_i^{\text{min}}}{l^{\text{max}}} \leq \sum_{d \in D} g_{id} \leq \frac{b_i^{\text{max}}}{l^{\text{min}}} \; \forall \, i \in I \tag{3.30}$$

The idea of this extension is also used in the construction heuristic of the first subproblem as we explain in Section 3.4.1.

The two subproblems are exactly solved sequentially with all the combinations of extensions and no extensions. The results are given in Chapter 4. We will compare the outcome of this method with the outcome of the other two methods and the best known solutions and compare method. The second and third method are both a HSA, only the third method includes a local search algorithm. In the following section we explain the second method of this thesis.

3.4. Method 2: HSA

The second method to find solutions for the NRP is a Harmony Search Algorithm (HSA). In Figure 3.1 an overview of this method is given. The global idea of the HSA is to create a solution pool and improve this solution pool by generating new solutions in each iteration. The aim is to find a new solution that is better than the worst solution until the stopping criterion is met. In this HSA a solution in the solution pool is an outcome of the first subproblem. The solution pool is initialised by random feasible solutions of the first subproblem that are created by a construction heuristic. This construction heuristic is explained in Section 3.4.1 For every solution in the solution pool, we determine the quality of this solution with a fitness function that is based on the second subproblem. Since a good solution of the first subproblem does not directly lead to a good solution of the second subproblem, we use (an indication of) the objective value of the second subproblem to determine what the quality of the solution of the first subproblem is. The three fitness functions that are used, are:

1. Exact fitness function: the exact solution value of the second subproblem. The second subproblem is solved by using the MIP-solver. This solver solves the second subproblem exactly. The objective value gives the value of the fitness function.
2. Relaxed fitness function: a lower bound of the second subproblem is used to determine the quality of the solution. A relaxation of the second subproblem is solved using the MIP-solver. This gives the lower bound and this value is used as fitness value.
3. Heuristic fitness function: an upper bound of the second subproblem is used as fitness value of the solution. The second subproblem is solved by a heuristic and the objective value of the heuristic gives the upper bound of the second subproblem.

After the quality of each solution is determined, the HSA searches for new and better solutions. The algorithm finds new solutions by executing one of the following considerations: the random consideration, the memory consideration and the pitch consideration. These consideration are explained in more detail in Sections 3.4.3, 3.4.4 and 3.4.5. The random consideration generates a random new solution using the construction heuristic. The memory consideration combines solutions of the solution pool to a new solution. The pitch consideration can only be performed in combination with the memory consideration and adjusts the combined solution from the memory solution. Which consideration is executed is determined by the parameters called HMCR and PAR. The first parameter determines if random consideration or memory consideration is performed and the PAR parameter determines if the pitch consideration is executed. Both values are between one and zero. The higher the value of HMCR, the higher the probability that the random consideration is performed. With a value of PAR close to one, in almost all cases pitch consideration is performed after the memory consideration. The aim of the three considerations is to escape a local optimum and to find the global optimum.

The quality of the new solution is determined by the fitness function. If the new solution is better than the worst solution of the solution pool, the new solution is included in the solution pool and the worst solution is removed from the solution pool. If the new solution is worse than the worst solution of the solution pool, the solution pool is not updated. The HSA stops when the stopping criterion is met. Since we execute the algorithm for a certain computation time, the algorithm always stops when the maximum time is met.

Next to the time we have two other stopping criteria. For these stopping criteria the algorithm stops when the solution has not been improved for a certain number of iterations. This number is related to the size of the problem. The maximum number of iterations depends on the number of employees and the planning horizon. For the first stopping criterion the maximum number of not improved iterations is $2*($the number of days in the planning horizon $+$ the number of employees$)$. The maximum number of not improved iterations for the second stopping criterion is the maximum of the number of days in the planning horizon and the number of employees multiplied by two. These stopping criteria are based on iterative experiments. In these experiments the proposed stopping criteria showed that in most cases the found solution did not improve after these maximum number of iterations. To further research the effect of the two stopping criteria, we apply HSA with both stopping criteria and a no size dependent stopping criteria. The third stopping criteria is only time dependent: the algorithm stops when the maximum running time is reached.

An overview of the HSA is given by the Algorithm 1. Method 3, the Hybrid Harmony Search Algorithm, is also explained in this algorithm. Lines 13 till line 17 apply only in the third method. The third method is similar to the second method only VNS is applied to some new solutions, this method is explained in Section 3.5. First, we explain the elements of the HSA in more detail. The construction heuristic is explained in Section 3.4.1. in Sections 3.4.2, the three fitness functions are explained. Sections 3.4.3, 3.4.4 and 3.4.5 elaborate on the different considerations of the HSA.

Algorithm 1: (Hybrid) Harmony Search Algorithm

input : List of employees and shifts and the planning horizon
output : Solution for the NRP

```
1  Initialise solution pool with random solutions generated by the construction heuristic ;
2  while stopping criteria is not met do
3      Generate RandomVariable1 ~ U[0, 1];
4      if RandomVariable1 ≤ HMCR then
5          Generate new solution by random consideration
6      else
7          Generate new solution by memory consideration;
8          Generate RandomVariable2 ~ U[0, 1];
9          if RandomVariable2 ≤ PAR then
10             Adjust new solution with pitch consideration;
11         end
12     end
13     if Hybrid Harmony Search Algorithm then
14         Generate RandomVariable3 ~ U[0, 1];
15         if RandomVariable3 ≤ VNS_parameter then
16             Improve the new solution by performing VNS;
17         end
18     end
19     if Objective value(new solution) ≤ Objective value(worst solution of solution pool) then
20         Replace the worst solution with the new solution;
21         Determine the best and worst solution of the solution pool;
22     end
23 end
```

3.4.1. Construction heuristic

The construction heuristic generates the initial solutions for the solution pool of the HSA. When the random consideration is performed in the HSA, the construction heuristic is also used to generate a new random feasible solution. The solutions generated by the construction heuristic are random feasible solutions to the first subproblem. For each employee a random pattern of working days and days off is generated. The pattern is a schedule of sequential blocks of working days and days off. The length of each block is randomly chosen while taken the minimum and maximum consecutive days off or working into account. For the duration of a working block, this is: $c_i^{min} \leq$ duration working block $\leq c_i^{max}$. For the consecutive days off, only a minimum is given. To obtain the number of maximum consecutive days off, we determine the desired ratio between being off and working for the employee according to the average contract hours. We multiply the maximum consecutive working day with this ratio to compute the maximum consecutive days off for the employee. First, the activity on the first day of the planning horizon is chosen randomly. The activity can be working or not working. The probability to start with working or with time off is equal. Subsequently, we walk through the planning horizon in steps of activity blocks. For each block we determine the duration of the activity as explained above. After having determined the duration, we assign the days of the activity block to the activity for that employee. Next, we generate an activity block for the other activity. If the current activity is working, the next activity will be time off and vice versa. A short example of the heuristic: imagine that the first activity is working with a duration of 3 days. At day 4, the activity time off starts with a random chosen duration, for example 2 days. The days $[1, 2, 3]$ are assigned to working days and the days $[4, 5]$ are assigned to days off. At day 6, the employee will work again. Since the constraints of the minimum consecutive working days and days off are not forcing to have more than c_i^{min} working days and o_i^{min} days off in the first and last working days of the planning horizon, we allow that the first activity block has a duration of $c_i^{\text{min}} - 1$ or $o_i^{\text{min}} - 1$. The last activity block, can have any duration. This is conform the constraints of the second subproblem.

Algorithm 2: Construction heuristic

input : Employees, planning horizon
output : For all employees a feasible roster on day level

```
for Each employee do
    feasible = False;
    while not feasible do
        d = 1 ;
        Chose starting activity random: current activity;
        while d ≤ planning horizon do
            if current activity = working then
                Determine the duration of working: duration ;
                d_next = d + duration;
                Assign days [d, d_next⟩ to working;
                Set the next activity as time off: new activity = time off
            end
            if current activity = time off then
                Determine the duration of time off: duration ;
                d_next = d + duration;
                Assign days [d, d_next⟩ to time off;
                Set the next activity as working: new activity = working
            end
            d = d_next ;
            current activity = new activity
        end
        Check feasibility and number of working days;
        if feasible and correct of number of working days then
            assign working days and days off to employee;
            feasible = True;
            break
        end
    end
end
```

When a pattern is generated for an employee, we check if the pattern is feasible. The minimum and maximum consecutive days off and working days are already taken into account during the generation of the pattern. There are two other constraints that have to be taken into account to determine whether the generated pattern is feasible. First, the maximum number of weekends is checked. Each employee has a maximum of allowed weekends to work. We check if the maximum is not exceeded in the generated roster. We also check if the requested days-off are given in the generated roster.

Next to the feasibility of the pattern, we check the number of working days in the pattern. This check is based on the third extension of the days-off/on assignment problem. For each employee, we determine a number of required working days in the pattern. We do this in same way as we do in the third extension. The minimum number of required working days is equal to: $\frac{b_i^{\text{min}}}{l^{\text{max}}}$. The maximum of required working days is: $\frac{b_i^{\text{max}}}{l^{\text{min}}}$. Where b_i is the minimum or maximum contract working time of the employee and l the shortest (min) and longest (max) shift. We only allow patterns that contain a number of working days that is equal or more than the minimum of required working days and equal or less than the maximum number of required working days. The purpose of checking the number of working days is to generate a random pattern that is able to respect the contract hours of the employee. By respecting the contract hours of the employee we have a higher probability that this day roster gives a feasible solution to the second subproblem. If the roster is feasible and the number of working days is good, the roster is assigned to the employee. Note that the heuristic does not take the preferred daily cover level into account. It is possible that there are more or less employees than needed, assigned to work on a day. The pseudocode of the construction heuristic is given in Algorithm 2.

3.4.2. Fitness functions

As described in Section 3.4 three different fitness functions can be used to determine the quality of the solution of the first subproblem. The exact fitness value is found by solving the second subproblem using the MIP-solver. The heuristic and relaxed fitness functions are described in the following two sections.

Relaxed fitness function

The relaxed fitness value is the objective value of a relaxation of the MIP-formulation of the second subproblem and the sum of the penalties when an employee does not work a shift. The relaxation of the second subproblem relaxes the shift assignment variable, x_{idt} to a continuous variable instead of a binary variable. The under- and overcoverage variables, y_{dt} and z_dt, are relaxed to continuous variables with a lower bound of zero instead of integer variables with a lower bound of zero. So the relaxed variables are:

$$0 \leq x_{idt} \leq 1 \qquad \forall\, i \in I, d \in G_i, t \in T \tag{3.31}$$

$$y_{dt} \geq 0 \qquad \forall\, d \in D, t \in T \tag{3.32}$$

$$z_{dt} \geq 0 \qquad \forall\, d \in D, t \in T \tag{3.33}$$

The constraints and objective of the second subproblem are the same as explained in Section 3.2. In this relaxation, employees can be assigned to parts of shifts. In practice, this is impossible but this relaxation gives a lower bound of the second subproblem and therefor a indication of the objective value of the second subproblem. The relaxed problem is solved exactly using the MIP-solver. The objective value of the relaxation and the sum of the penalties of the work preferences together give the fitness value of the solution of the first subproblem.

Heuristic fitness function

The fitness value of the heuristic fitness function is found by heuristically solving the second subproblem. The aim of the heuristic is to assign all shifts on the working days to employees without violating any constraints. The heuristic takes the constraints of the second subproblem into account. E. K. Burke et al. (2008) proposed a ordering heuristic to assign employees to shifts. First, a heuristic fitness function based on the ordering heuristic of E. K. Burke et al. (2008) was proposed. Since this ordering heuristic fitness function was not capable of finding feasible solutions, another heuristic fitness function is used. This heuristic fitness function heuristically assigns employees to shifts on their working days. For each sequence of working days, the heuristic searches for a feasible sequence of shifts. A found sequence of shifts is feasible when no constraints are violated by this sequence. To check if the sequence does not violate a constraint, we check the current status of the employee during the assignment of shifts. The status of the employee contains the following information:

- The working days to which shifts have to be assigned.
- The leftover maximum number of a shift type.
- The current working time of the employee. This is necessary to not exceed the maximum contract hours of the employee. The employee can also not work less than the minimum contract hours.
- The already assigned shifts and which shift type is not allowed to follow this shift.

Algorithm 3: Heuristic fitness function

```
   input  : Solution of the first subproblem, information about the employees and shifts
   output : Heuristic second subproblem fitness value solution to the first subproblem
1  for Each employee do
2      StatusEmployee = status of employee without any shifts assigned;
3      possible = True;
4      StatusAssigning = StatusEmployee ;
5      for Each sequence of working days do
6          if not possible then
7              StatusAssigning = StatusEmployee
8          end
9          for Each day in the sequence of working days do
10             Find possible shifts at the working day while checking StatusAssigning;
11             if Possible shifts are found then
12                 Chose one shift according probablistically with the maximum number of shift
                    types;
13                 StatusAssigning = update StatusAssigning according to chosen shift;
14                 possible = True;
15             else
16                 Number of no possible shifts found += 1 ;
17                 if number of no possible shifts found > 10 then
18                     Start over at the first working day of the employee; go to line 2;
19                 else
20                     Start over at first day at sequence; go to line 6;
21                     possible = False;
22                 end
23             end
24         end
25         StatusEmployee = StatusAssigning
26     end
27 end
```

Per working day in the sequence of working days we determine which possible shifts that take place that day can be added to the sequence. A shift is possible if the maximum number of shift types is not exceeded for all shift types; the maximum working hours are not exceeded and the shift can follow the shift of the previous day. If the previous day was a rest day, we do not take the last argument into account. If no constraints are violated, the shift is considered as a possible shift. Of all the possible shifts, we choose one shift which is added to the sequence. After choosing the shift, we update the status of the employee. We assign shifts to the working days of the employee until the employee works a shift at all working days. The probability of choosing the shift is aligned with the current maximum shifts of each type, for example: an employee has two possible shifts on working day x, a night shift and a day shift. This employee is allowed to have maximum 4 night shifts and 20 days shifts and is already assigned to 3 night shifts and 13 day shifts. The probability of having the night shift is $\frac{4-3}{(4-3)+(20-13)} = \frac{1}{8}$ and the probability of working the day shift is $\frac{7}{8}$. The probability that the employee is assigned to work the day shift at day x is higher than working the night shift.

If no possible shifts are found for a work day, the heuristic starts again at the first day of the sequence and assigns shifts to the sequence. If the heuristic re-starts at the first day of a sequence for a few attempts and still does not succeed to find a possible shift to assign to a day in this sequence, we start again at the first working day of the employee. If the heuristic does not succeed to assign shifts to all working days after 100 attempts of restarting at the first working day, the solution is set to infeasible.

When we succeed to assign shifts to all working days, we check if the employee works less than the minimum working hours. If this is the case, we re-assign the shifts to all working days while stimulating that longer shifts are assigned. We stimulate this by increasing the possibility of choosing a possible shift with a long duration. We re-assign shifts to employees on their working days until the minimum working hours constraint is not violated. If we do not succeed to fulfil the minimum working hours after 100 attempts, we conclude that the solution is infeasible.

When shifts are assigned to all working days of all employees and the shift assignment is feasible, the objective function is computed. The objective function is the same as the objective function of the second subproblem. In this case, also the penalties if an employee is not assigned to a preferred shift are added to the objective value. The objective function consists of two parts, the first part has to do with the demand and the under- and overcoverage. The second part is calculated by the preferences penalties. For the first part the under- or overcoverage of each shift are determined and if necessary the penalties are applied. The second part is calculated by checking if the preference to not work a shift is granted. If the preference is not granted, a penalty is applied. The heuristic fitness function pseudocode is given in Algorithm 3.

3.4.3. Memory Consideration

In the HSA, a new solution is created by one of the considerations. One of this consideration is the memory consideration. The memory consideration constructs a new solution of elements from the solution pool. An element of a new solution is an individual rosters in the solution pool. Per employee we choose randomly one of the ten best solutions in the solution pool. From this solution, we select the individual roster of the current employee. For example, for the first employee we choose the second solution of the solution pool. From the second solution, we select the individual roster of the first employee. This will be the new day roster of the first employee. All chosen individual roster together form a new solution.

3.4.4. Pitch consideration

To change the solution of the memory consideration, pitch consideration is applied. The pitch consideration is only performed in combination with the memory consideration. The pitch consideration adjusts the solution to move a day from one activity to another for one employee. The activity is either working or not working. In the pitch consideration, one roster is adjusted of a randomly chosen employee. While taking the minimum and maximum consecutive working days and days off into account, we search for days that can be moved to the other activity. To simplify the procedure we only consider the first or the last day of the activity as day that can be moved. For the first activity only the last day is considered and only the first day is considered for the last activity.

In the pitch consideration we can change a day in two possible ways. One possibility is moving a day from working days to days off and the other is moving from days off to working days. These moves are only possible when the activity where the day moves away has a duration more than the minimum consecutive days and the activity to which the day moves will not exceed the maximum consecutive days. For example, if the minimum consecutive working days are three, the maximum consecutive working days are four and the minimum consecutive days off are two. Assume the following roster for an employee with a planning horizon of 10 days: working at the days $[1, 2, 3, 4]$ and $[8, 9, 10]$ with the days off at days $[5, 6, 7]$ in between. It is possible to swap day 4 and day 8 to days off and swap day 7 to working day. It is not possible to swap day 5 to working because the maximum consecutive working days will be exceeded with this move.

After all possible moves are found, we randomly pick one possible move to execute. After the move, we check if the new solution is still feasible for the first subproblem. We check if the new solution does not exceed the maximum weekend and still assigns requested days off to the employee. If the new individual solution is not feasible or no move is possible, the consideration gives no new solution so the new solution is equal to the solution of the memory consideration. When the move is successful, the adjusted solution is the new solution.

3.4.5. Random consideration

The random consideration generates a new solution by using the construction heuristic as explained in Section 3.4.1. By the construction heuristic, a random feasible solution to the first subproblem is generated. The aim of generating a random solution is to find a different solution than the solutions in the solution pool. If this different solution is better than the worst solution in the solution pool, we possibly escape a local minimum.

3.5. Method 3: HHSA

Method 3 is the Hybrid Harmony Search Algorithm (HHSA). This the HSA combined with the Variable Neighbourhood Search algorithm. As shown in Figure 3.1, method 3 is similar to method 2 only the VNS is added. After a new solution has been created using one of the considerations, VNS is performed for some new solutions.VNS is applied if a random variable between 0 and 1 is less than or equal to the VNS parameter. In Algorithm 1 an overview of the HHSA is given. Gong et al. (2021) proposed a HSA combined with Tabu search. Since this research showed promising results for larg-size instances, first Tabu search was added to the HSA. The first results showed that the combination of HSA and Tabu search obtained no good results for the NRP. Therefore, a VNS algorithm is added to the HSA instead of the Tabu search. As described by Tassopoulos et al. (2015), VNS is a promising method to solve the NRP. The VNS algorithm we use is party based on the VNS applied by Tassopoulos et al. (2015). The aim of adding VNS to HSA is to further differentiate the new solution and thereby escape the local minima. We will first explain the general approach of the VNS algorithm in Section 3.5.1. Then we explain the neighbourhoods of the VNS algorithm.

3.5.1. General approach VNS

The VNS algorithm aims to improve the solution to the first subproblem by searching for better solutions. A better solution is a solution with a lower fitness value. The VNS starts with a solution that is generated during the random, memory or pitch consideration in the HSA: the start solution. We search for the best new solution by adjusting the start solution for each employee. The adjustments to the solution are made by considering different changes to the solutions; these changed solutions are called neighbourhoods.

In each neighbourhood we search for a better solution for the current employee. In this VNS uses five different neighbourhoods to find a better solution. These neighbourhoods are explained in the Sections 3.5.2 to 3.5.6. For each employee we initially set the start solution as current solution and apply the first neighbourhood search to the current solution. If a better solution is found in the first neighbourhood, we replace the current solution by the better solution and search again in the first neighbourhood. If no better solution is found, we apply the second neighbourhood search to the current solution. Every time we find no better solution in a neighbourhood, we apply the next neighbourhood search to the current solution until no better solution is found in all five neighbourhoods. Every time we find a new better solution, we replace the current solution by the new found solution and start again in the first neighbourhood. For each search in a neighbourhood the quality of the new solution is determined to evaluate if the new solution is better than the current solution. The quality of the new solution is determined by the fitness function of the HSA. If the fitness function is the heuristic fitness function, we determine the quality of the new solution also by the heuristic fitness function.

To find the best new solution we store the new solution as the best known solution if this one is better than the best known solution until so far. After we did not find a better solution for an employee in all five neighbourhoods, we search for a new better solution for the next employee. The pseudocode of the VNS algorithm is given by Algorithm 4.

3.5.2. First neighbourhood

The first neighbourhood changes a working day to a day off for the employee. Only one day is changed due to the computation time that is required to determine the quality of the changed solution. To determine which day is changed to a day off, we first determine all days that can be changed. A day can be changed to a day off if the sequence of working days to which the day belongs is longer than the minimum consecutive days and the day is the last or the first day of the sequence. If the last day of the planning horizon is in the sequence of working days, we only consider the first day of the working sequence. If day one is in the sequence of working days, the first day can be changed if the number of minimum consecutive days off is either 2 or 1. Of all the working days of an employee that can be changed to a day off, we choose randomly one day. This day is changed from working day to a day off. An example to illustrate the change in the solution is shown in Figure 3.2: assume a roster for an employee with a planning horizon of 14 days. The minimum number of consecutive working days is two and the maximum is five. The minimum number of consecutive days off is two. The employee works on the days: 1,2,3,4,7,8,9,10,13 and 14 (the blue marked days) and the employee is free on the other days. All possible working days that can be changed are: day 1, 4, 7 and 10. We choose day 7

Algorithm 4: The Variable Neighbourhoods Search Algorithm

input : start solution to the first subproblem, information about the employees
output : New solution to the first subproblem

```
k_max = 5;
Best known solution = Start solution;
for Each employee do
    Best individual solution = start solution;
    k = 1;
    while k ≤ k_max do
        if k = 1 then
            Perform first neighbourhood search to find new solution
        else if k = 2 then
            Perform second neighbourhood search to find new solution
        else if k = 3 then
            Perform third neighbourhood search to find new solution
        else if k = 4 then
            Perform fourth neighbourhood search to find new solution
        else
            Perform fifth neighbourhood search to find new solution
        end
        if Objective value(new solution) ≤ Objective value(best individual solution) then
            Best individual solution = new solution ;
            k = 1;
            if Objective value(new solution) ≤ Objective value(Best know solution) then
                best known solution = new solution
            end
        else
            k = k + 1
        end
    end
end
```

to change. After the change the employee works on days 1,2,3,4,8,9,10,13 and 14 and is off at days 5,6,7,11 and 12. It is important to note that we only apply this neighbourhood search if the number of working days in the start solution is more than the minimum number of required working days. This minimum number of required working days is determined in the same way as the third extension of the first subproblem (see Section 3.3.3).

3.5.3. Second neighbourhood

The second neighbourhood search changes the solution in the opposite way of the first neighbourhood search. In this neighbourhood we change one day from a day off to a working day. With an example we show how the neighbourhood change is performed. Assume again a solution for an employee with a planning horizon of 14 days. The minimum consecutive days working is two, the maximum is five. The minimum consecutive days off is two. The employee works on days 4,5,8,9,10,11. First, we determine which days could be added to sequences of working days. For each sequence of working days that is shorter than the maximum of consecutive working days, the days before and after the sequence can be added to the sequence. Note that it is not possible to add a day that is not in the planning horizon, for example day 0. If day 1 is not a working day, this day can always be added if the minimum consecutive working days is 2 or 1. In our example, we could add days 1,3,6,7 and 12 to the working days. During the change we also take into account that we should not violate the minimum consecutive days off. Therefore, we are not able to change day 6 or 7 into a working day. Next, we check whether a day we want to change is a day the employee cannot work or if by changing the day to a working day, we violate the maximum number of weekends of work for the employee. If this is not the case, we choose one of the days we could change to a working day. In our case, we can change days 1,3 and 12. We choose day 3 randomly, now the employee works at day 3,4,5,8,9,10,11. This example is shown in Figure 3.2. We only add a working day if the number of working days of the employee will not exceed the maximum number of working days.

3.5.4. Third neighbourhood

The third neighbourhood is a combination of the first and second neighbourhoods. For the current employee we swap one working day to a day off and one day off to a working day. During the swap the minimum and maximum consecutive working days and the minimum consecutive days off are respected at all times. Next to this, we check if for each employee the requests to have a day off are granted and the maximum number of weekends with work are not violated. This neighbourhood does not change the number of working days for the employee, as it swaps two days in the solution. In the neighbourhood we first determine which combinations of two days can be swapped. Of all the possible swaps, we randomly pick one swap and perform this swap. In Figure 3.2, an example is shown of how the solution can be changed in this neighbourhood.

3.5.5. Fourth neighbourhood

This neighbourhood changes the solution for two employees. Since we perform the VNS algorithm for each employee, one of the employees is fixed. The other employee for which the solution is changed, is randomly chosen. If it is not possible to apply the change for this employee we choose another employee until the change is possible or no employee to choose is left. We perform a swap between two employees. At the same day for both employees, we swap the activity between the two employees. At the chosen day we change a working day to a day off for one employee. For the other employee we change a day off to a working day at the same day. For this change we respect all constraints of the first subproblem. We respect the minimum and maximum consecutive days and the maximum number of weekends with work at all times. Next to this, we still want an employee to have a days off when this is requested and we do not change the solution if the new solution does not fulfil the minimum and maximum required working days for one of the employees. In Figure 3.2, an example of this neighbourhood is given. The start solution is given by days $[2, 3, 4, 9, 10, 11, 12, 13]$ as working days for the first employee and the working days $[1, 2, 3, 4, 5, 11, 12, 13]$ for the second employee. In the example, the swap is performed at day 5 because this day is a working day for the second employee and a day off for the first employee. At this day it is possible to perform the swap without violating any constraints of subproblem 1. In the new solution we see that day 5 is now a working day for the first employee and a day off for the second employee.

3.5.6. Fifth neighbourhood

The last neighbourhood also changes the solution for two employees. Again the first employee for which the change is made is fixed in the VNS algorithm, the other employee is chosen randomly such that the change can be performed. In this neighbourhood we cut the roster of both employees at a certain day. We swap the roster of the employees after the cut. During this swap we make sure that constraints of the first subproblem are not violated. We do not perform the swap if in the new solution the number of working days is less than the minimum allowed working days for an employee or the number of working days is more than the maximum allowed working days for one of the employees. The swap is illustrated in Figure 3.2, where the days roster of both employees are cut at day 3. The roster of employee 1 after day 3 is given to the second employee and vice versa.

3.6. Compare method

To compare the above explained methods, we can compare the results of the methods with the results of the benchmark instances as explained in Section 2.2. The results of the three methods are also compared with the results of the compare method. This method solves the complete NRP using the MIP-solver with a maximum solving time; see Figure 3.1 for an illustration of this method. This maximum solving time is equal to the maximum solving time of the three methods. The MIP-solver solves the instances to optimality. If the found lower bound is equal to the best objective, the problem has been solved to optimality. If this is not the case, the lower bound differs from the objective value.

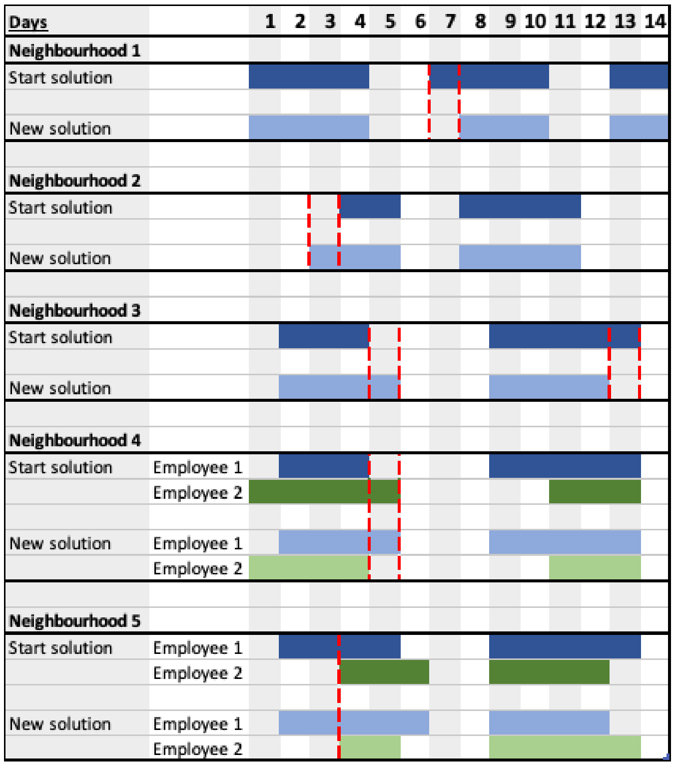


Figure 3.2: Overview of the neighbourhoods of the VNS. The blue and green marked days are the working days of an employee

4
Results

As shown in Figure 3.1, in this thesis three methods are used to find solutions to the NRP. These three methods are described in Chapter 3. As explained in Chapter 2, we use the benchmark instances to compare the results of the methods with the best best known results. Next to this, we use the results of the compare method to compare the obtained results. Per method, we obtained results from the benchmark instances. These results are the objective value and solving time per instance. In Section 4.2, the results of method 1 are discussed for the different combinations of extension as explained in Section 3.3. The Sections 4.3 and 4.4 the results of the HSA and HHSA are discussed. The results of HSA and HHSA are shown per fitness function and the general results are discussed. First, the results of the compare method are given. These results can be compared with the results of the three methods. All the results are obtained under the same circumstances, for example the same computer. The results of the three methods are given in more detail in Appendices B,C and D.

Table 4.1: Result of the compare method

Instance	Best known objective value	Lower Bound	Found objective value	Deviation rate from best known objective value (%)	Solving time(s)
1	607	607	607	100.00	6.87
2	828	828	828	100.00	38.83
3	1001	1000	1001	100.00	600
4	1716	1205.88	1863	108.57	600
5	1143	708.35	1174	102.71	600
6	1950	1811.93	2381	122.10	600
7	1056	1024.98	1070	101.33	600
8	1300	-	-	-	600
9	439	57.89	571	130.07	600
10	4631	4236	5282	114.06	600
11	3443	-	-	-	600
12	4040	-	-	-	600
13	1348	-	-	-	600
14	1278	-	-	-	600
15	3829	-	-	-	600
16	3225	-	-	-	600
17	5746	-	-	-	600
18	4459	-	-	-	600
19	3149	-	-	-	600
20	4769	-	-	-	600
21	21133	-	-	-	600
22	30240	-	-	-	600
23	16990	-	-	-	600
24	26571	-	-	-	600

4.1. Results compare method

We compare results of the three methods with the benchmark instances and the results of the compare method. The compare method solves the complete NRP by using the MIP-solver. The results of the compare method are per instance the lower bound, objective value and solving time. If the lower bound is equal to the objective value, the found solution is optimal. The maximum solving time is set to 600 seconds. The results are given in Table 4.1. As shown in the results, instance 1 and 2 are solved to optimality in less than the maximum solving time. For the other instances, no optimal solution is found within 600 seconds. For the instance 2 to 7 and 9, 10, a solution is found. For the other instances, no solution is found. We can use this results to compare the solving time of the complete NRP against the solving time of the decomposed NRP.

4.2. Results method 1: Solving the subproblems sequentially exactly

The first method as explained in Section 3.3 searches for a solution to the NRP by solving both subproblems sequentially with the MIP-solver. First, we search for the optimal solution for the first subproblem and with this solution, we solve the second subproblem to optimality. For both subproblems we set the maximum computation time to 600 seconds. We solve the first subproblem for different combinations of the three extensions. The combinations we apply are: no extensions, only one extension, combination of two extensions and all extensions. For the first two extensions, we penalise the objective by a weight. We apply this extension for different weights. The first extension, called the working hours extension, penalises the deviation of the desired working days, the different value of the weights are $10, 20, 50$. The weights of the second extension, the preference extension, are $5, 10, 15$. These weights are chosen because first experiments of solving the NRP with these values showed good results. The third extension has no weights so it is either added to the first problem or not. The results of the different combinations are given in Appendix B. Since the objective value of the first subproblem does not indicate anything about the quality of the solution of the second subproblem, we mostly look at the solving times and the objective value of the second subproblem.

For solving the two subproblems to optimality, no feasible solution of the second subproblem was found for the combination in which no extensions are applied. For 18 of the 24 benchmark instances we found a feasible solution for the days-on/off assignment subproblem. Of these 18 instances, 9 instances used the maximum computation time. In general, the solving time of the first subproblem is much higher than the second subproblem. This indicates that finding a feasible day-roster takes more time than assigning shifts to employees at their working days.

For the combination with only the working hours extension, see Table B.5, we did not obtain a feasible solution to the complete NRP. We retrieved a feasible solution to the first subproblem for all weights of the working hours extension for 17 instances, but for the second subproblem no feasible solutions were found. The solving time is again higher for the first subproblem compared to the second subproblem. The second subproblem is solved in 30,61 seconds at most while the first subproblem reaches the maximum computation time for almost all instance 11 to 24. When the weight of the working hours extension is 50, the solving time of the first subproblem is the lowest for most instance that stop before the maximum solving time (5 of 10 instances).

For the combination with only the preference extension, we see approximately the same. No feasible solutions are found for the NRP and the solving time of the first subproblem is higher than for the second subproblem. We can solve the days-on/off assignment subproblem for 18 of the 24 instances. It is not possible to relate the solving times to the weights of the second extension. The solving times of the first subproblem are lower with only the preference extension than when we apply the working hours extension.

When we apply only the third extension, we also have no solutions to the NRP. We obtain a solution to the first subproblem for 18 instance. For 10 instances, this solution is obtained in less than 600 seconds. The solving times of both subproblems are comparable to the solving time of the combination with only the preference extension. There are three possible combinations with two extensions, these combination are:

1. The combination of the working hours extension and the preference extension. We call this combination combination A.

2. Combination B: combination of the working hours extension and the third extension.
3. Combination C: the combination of the preference extension and the third extension.

Combination A and C do not give a feasible solution for the shift assignment problem either. Both have a higher solving time for the first subproblem than the second subproblem. Method 1 with combination C is able to find a feasible solution to the first subproblem for 18 instances for all weights of the preference extension. With combination A, we find for 18 instances a feasible solution to the first subproblem for all weights of the working hours and preference extension.

The two combinations of extensions for which we have found a feasible solution to the second problem are combination B and the combination of all extension. Combination B is the combination of the working hours extension and the third extension. Of these two combinations, the deviation of the best found solution to the best known solutions are calculated. For the best found solutions the objective value and the corresponding solving time for both subproblems are given in Table 4.2. For combination B, the result show that we found a feasible solution for 9 of the 24 instances for all weights of the working hours extension. For instance 18, we only found a feasible solution to the entire NRP when the weight of the working hours extension is 20. The objective values we find are much higher than the best known objective values. The best found objective value is between 163.41 and 503.19 percent of the best know objective value. The average deviation rate from the best known objective value is 289.79 percent. There is no clear relation between the weight of the working hours extension and the best found objective value.

Table 4.2: Results of method 1 with the combination of the working hours and third extension and the combination of all extensions

	Working hours and third extension					All extenstion				
Instance	Best objective value second subproblem	Deviation rate from best known objective value (%)	Solving time first subproblem (s)	Solving time second subproblem (s)	Weight of working hours extension	Best objective value second subproblem	Deviation rate from best known objective value (%)	Solving time first subproblem (s)	Solving time second subproblem	[Weight of working hours extension, Weight of preference extension]
1	1325	218.29	0.92	0.05	10	1321	217.63	1.41	0.05	[10, 5]
					20					[10, 15]
					50					[20, 5]
										[20, 15]
										[50, 15]
2	1353	163.41	3.09	0.14	10	1341	161.96	3.25	0.12	[20, 10]
3	2342	233.97	5.66	0.44	20	2328	232.57	2.78	0.34	[10, 10]
4	3470	202.21	30.19	0.19	50	3364	196.04	24.86	0.22	[10, 10]
5	3696	323.36	87.02	0.33	10	3582	313.39	161.03	0.35	[10, 5]
										[10, 15]
6	4497	230.62	74.76	0.62	50	4486	230.05	92.96	0.61	[50, 5]
7	2728	258.33	100.68	0.92	50	2697	255.40	600.00	0.97	[20, 15]
8	-	-	-	-	-	-	-	-	-	-
9	2209	503.19	67.39	4.66	50	1777	404.78	600.00	4.86	[50, 5]
10	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-
12	13800	341.58	600.00	12.00	20	15754	389.95	600.00	14.03	[20, 15]
13	-	-	-	-	-	15265	1132.42	600.00	147.99	[50, 5]
14	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-
17	-	-	-	-	-	-	-	-	-	-
18	18855	422.85	456.93	3.91	20	18404	412.74	600.00	3.17	[20, 5]
19	-	-	-	-	-	-	-	-	-	-
20	-	-	-	-	-	-	-	-	-	-
21	-	-	-	-	-	-	-	-	-	-
22	-	-	-	-	-	-	-	-	-	-
23	-	-	-	-	-	-	-	-	-	-
24	-	-	-	-	-	-	-	-	-	-
Mean		*289.78*					*358.81*			

In general the results of this combination show that the solution time of the second subproblem is less than the solving time of the first subproblem as we also see at other combinations. With this method, we are not capable to find a feasible solution to large instances. The largest instance for which we find one feasible solution is instance 18 (12 weeks, 22 employees), of which the objective value (18855) is again much higher than the best known solution for this instance (4459). The first method with extension combination B is not capable to find a solution for the first subproblem within 600 seconds for instance 8, 11 and from instance 16 to 24, except from instance 18. For instance 1 to 9, except 8, method 1 with extension combination B is able to find a feasible solution to the NRP in less time than the compare method. The found objective values by using method 1 are worse than the objective values that are found by using the compare method.

Method 1 with the combination off all extensions is able to find a feasible solution to the NRP for 8 instances for all values of the preference and working hours extension. For instance 12, 13 and 18, we are able to find a solution for some combinations of values for the preference and working hours weights. The objective values of the feasible solutions are much higher than the best known solutions. The found objective values are between 161.96 and 1132.42 percent of the best known objective value. The mean deviation rate from the best known objective value obtained by using method 1 with all extension is 358.81 percent. The weights of the extensions for which the best objective values are found differ too much to determine which weights give the most best results. The solving times are again less than the solving times of the compare method but the found objective values are worse.

Method 1 extension combination B was not capable of finding a feasible solution to instance 13. If we neglect this instance, the mean deviation rate for method 1 with all extensions is 281.45 percent. This indicates that the found objective values with method 1 with all extensions are slightly better or equal to the objective value with extension combination B. The solving time for the first subproblem are much higher than for the second subproblem. Overall the solving times of the both subproblems are higher for the combination with all extensions compared to the combination with the working hours and third extension.

In short, the first method we use to solve the NRP in two subproblems finds mostly infeasible solutions to the NRP. By adding extensions to the first subproblem, we find feasible solutions to the NRP for most of non medium-size and small-size instances. We find feasible solutions if we apply the combination of all extensions or the combination of the third and working hours extension. The obtained objective values for both combinations of extensions are much higher than the best known objective values. The solving times are less than the solving times of the compare method. The solving time of the first subproblem is higher than the solving time of the second subproblem.

4.3. Results method 2: HSA

The second method we used to solve the NRP is the HSA. We initialised the algorithm with a solution pool of random solutions to the first subproblem. Until the stopping criterion is met, the HSA generates new solutions by performing the random or memory or pitch consideration. After a new solution is generated, we determined the quality of this solution by a fitness function. This fitness function is based on the second subproblem. The three fitness function that we use are the exact, relaxation and heuristic fitness functions. First, we discuss the results of the HSA per fitness function with different parameters. Second, we discuss the difference between the three fitness function in Section 4.3.4.

The NRP is solved with the HSA with different parameters. These parameters are the stopping criterion, solution pool size and values of the PAR and HMCR parameter. The last two parameters determine which consideration will take place to generate a new solution. The value of the HMCR parameter is the probability of performing the random consideration. If the HMCR parameter has a value of one, we will always perform the random consideration and never the memory consideration. A value of one for the PAR parameter means that the pitch consideration almost always is performed. The HSA is always terminated after 10 minutes but in two cases the algorithm is terminated earlier. These two cases are based on the number of iterations in which the solutions in the solutions pool are not improved. If the number of iterations in which the solutions are not improved exceeds the maximum number, the HSA is stopped. This maximum number and the values of the other parameters are given in Table 4.3. The values of the parameters are determined by experiments. In the experiments, the NRP was solved using method 2 and different values to the parameters. The values that gave for most experiments good results are chosen as values.

In Appendix C the results of the second method, the HSA are given. Per benchmark instance, the objective values and solving time for each parameter set is given. The parameter set contains the different values of the pool size, stopping criteria and PAR and HMCR parameters. For each fitness function, we are looking for the parameter set that gives the best solution. The best solution is the solution with the lowest objective value for that instance. The heuristic fitness function gives an upper bound of the exact objective value and the relaxation fitness function gives a lower bound of the exact objective value. To be able to compare the outcome of these fitness functions with the results of the

NRP, we solved the end solutions of the non-exact fitness functions exactly to obtain the exact objective value.

First, we discuss the results for the exact fitness function. We compare the pool size, stopping criteria and the values of the PAR and HMCR parameters. Subsequently, we discuss the results of HSA with the relaxation and heuristic fitness functions. To summarise the results of HSA, the best solution and corresponding solving time and parameter set are given per instance per fitness function in Table 4.4.

Table 4.3: The parameters for which the HSA is performed

Parameter	Value
Stopping criterion	10 minutes and the maximum of number that the solution is not improved:
	2$*$(the number of days in the planning horizon + the number of employees)
	2$*$(maximum(number of days in the planning horizon, number of employees))
	No maximum
Pool size	10
	20
PAR	0.3
	0.5
	0.7
HMCR	0.3
	0.5
	0.7

4.3.1. Results of method 2 with the exact fitness function

In Table C.1 to C.6, the results of the HSA with the exact fitness function are given. In general, the HSA is capable to find a feasible solution to NRP for 10 instances, instance 1 to 8 and instance 11 and 12. As shown in Table 4.4, most best solutions of these 10 instances are obtained when the HSA has a pool size of 20 (7 instances) and the second stopping criterion (6 instances). The objective values we found with the HSA with the exact fitness function are much higher than the best known objective values. The smallest difference is for benchmark instance 4, we found an objective value of 2049 while the best is 1716. The objective value we find is 119.41 percent of the best known solution so the found objective value is 19.41 percent higher. For the first six instances, the found objective value is less than 100 percent higher than the best known objective values.

With a pool size of 10, the best results are mostly obtained with the first stopping criterion (7 of 10 instances). With a pool size of 20, the difference between the stopping criteria is not large enough to decide. With a pool size of 10 and the first or second stopping criteria we found one or more feasible solution for the first 8 instances within 600 seconds. The search for feasible solutions to instance 11 and 12 is terminated by the maximum computation time of 600 seconds. When we choose a pool size of 20 solutions, we find approximately the same for the first two stopping criteria. The difference is that the algorithm is terminated by the maximum solving time for the third instance. With the third stopping criterion, the algorithm is terminated by the maximum solving time of 10 minutes so the solving time is always 600 seconds.

For the values of the PAR and HMCR parameters for the pool size of 10, no best values for the PAR and HMCR parameters can be distinguished. The results with the pool size parameter set to 20 and the three stopping criteria do not give a clear relation between the values of PAR and HMCR and the best found solutions. For the values of PAR and HMCR per each stopping criterion, the results show that the value of 0.3 is best for both parameters with the third stopping criterion. For the other stopping criteria, no best values of the PAR and HMCR can be obtained.

4.3.2. Results of method 2 with the relaxation fitness function

When we apply the relaxation fitness function to the HSA, we find the same number of best solutions but less feasible solutions compared to the exact fitness function. For most combinations of stopping criteria and pool sizes, we can find one or more feasible solution to ten instances. For all parameter set combinations, we cannot find a feasible solution to the NRP for instances 9,10 and 13 to 24.

For each stopping criterion, the pool size has no direct influence on the best solution. With the same stopping criterion, an equal number of best solutions are found for both pool sizes. Only for the third stopping criterion, we found that the pool size of 20 solutions is better; 5 of the best solution have a pool size of 20 and 3 a pool size of 10. With both solution pool sizes we find the most best solutions (5 of the 10 instances) when we apply the third stopping criterion, the HSA terminates after 10 minutes.

If we look per instance which best objective value is found, the best solutions have mostly a pool size of 20 (6 of 10 instance) and stopping criterion 3 (6 of 10 instances). For the PAR and HMCR, it is not possible to conclude which combination performs best. When we look per pool size, it is also not possible to find the best combination of the PAR and HMCR parameters. The same holds if we take only the stopping criteria into account.

The objective values are more than the best known objective values, as shown in Table 4.4. In the best case scenario (instance 4), we find a best solution with an objective value of 2040 where the best known solution is 1716. For all other instances, the HSA with the exact fitness function performs worse. Only for instance 1 and 5 the best found solutions are found withing 600 seconds. In general, the solving times are almost always 600 for the instance 3,8, 11 and 12.

4.3.3. Results of method 2 with the heuristic fitness function

The HSA with the heuristic fitness function is able to find a feasible solution for 14 instances, namely instance 1 to instance 14. For instance 13, we only find two feasible solutions. The results show that for stopping criteria 1 and 3, we find the most best solutions with a pool size of 20 (for stopping criterion 1: 11 of 14 instances and 8 of 13 for stopping criterion 3). For the second stopping criterion, a pool size of 10 or 20 are indifferent.

When we purely look into the effect of the stopping criteria, the results show that with a pool size of 10, the third stopping criterion is the best (7 out of 13 instances). For a pool size of 20, six of the best solutions have stopping criterion 1 and six have stopping criterion 3. We found the best solutions with a pool size of 10 when HMCR has a value of 0.3 for stopping criteria 1 and 2. The value of the PAR parameter does not matter for finding the best solution, so the number of times that the pitch consideration is performed has no influence on finding a best solution for that instance. For the combination of the third stopping criterion and pool size 10, we see that the combination of HMCR = 0.5 and PAR = 0.5 gives the most best solutions. If we performed HSA with a pool size of 20, we see approximately the same as with a pool size of 10 but the combination of a value of 0.7 for both the PAR and HMCR parameter gives the best solutions.

The best solution we found per instance shows that the most best solutions are found when the HSA is performed with a pool size of 20 (10 of 14 instances) and the third stopping criterion (7 of 14 instances). The combination of the values of PAR-HMCR that performs best are 0.3-0.3 and 0.5-0.5. Again, the HSA does not provide good results, see Table 4.4. The best solution compared to the best known solution is found for the first instance. The best known objective value is 607 and we find a objective value of 717. The worst solution we find is at instance 13, the objective differs 2301 percent with the best known solution. In general, the solving time is less than 600 seconds for the first 9 instances. For the larger instances and the solutions obtained with the third stopping criterion, the solving time is 600 seconds.

4.3.4. Overall results of the HSA

In Table 4.4, the best found solutions per instance per fitness function are given with the corresponding solving time and parameter set. For each best found solution we determine the deviation rate from the best known objective as shown in the table. The most of the best solutions are found with the relaxation fitness function (6 of 10 instances). Most of these solutions have a pool size of 20, the third stopping criterion and value of 0.3 or 0.7 to the PAR parameter. The value of HMCR differs a lot for the best found solutions. For most of the largest instances of which we found feasible solutions with the relaxation fitness function we see that a pool size of 20, stop criterion 3 and a value of 0.3 for the PAR parameter give the most best solutions.

For the exact and heuristic fitness function a pool size of 20 and stopping criterion 3 results in the most best solutions. The HSA with the heuristic fitness function is able to find a feasible solution for more instances than the other two fitness functions. The performance of HSA with the heuristic fitness function is worse than HSA with other fitness functions. The mean deviation rate per fitness function are, exact: 219.95, relaxation: 198.79 and heuristic: 479.37. These mean deviation rates indicate that the relaxation function finds feasible solutions that are on average 98.79 higher than the best known solutions. With the exact function this is 119.95 percent on average. With the heuristic fitness function the found solutions differ on average the most from the best known solutions: 379,37 percent. It is not fair to compare the solving times per fitness function for each instance since some best solutions are found with the third stopping criterion, so the algorithm is only terminated by the maximum solving time.

Table 4.4: The best objective values of HSA with corresponding deviation rate, solving time and parameter set per instance per fitness function

		Exact fitness function				Relaxation fitness function				Heuristic fitness function			
Instance	Best known objective value	Best objective value	Deviation rate from best known objective value (%)	Solving time (s)	Parameterset: [pool size, stopping criterion, PAR, HMCR]	Best Objective value	Deviation rate from best known objective value (%)	Solving time (s)	Parameterset: [pool size, stopping criterion, PAR, HMCR]	Best Objective value	Deviation rate from best known objective value (%)	Solving time (s)	Parameterset: [pool size, stopping criterion, PAR, HMCR]
1	607	812	133.77	600.00	[20, 3, 0.3, 0.7]	821	135.26	16.86	[20, 2, 0.5, 0.7]	717	118.12	600.00	[20, 3, 0.7, 0.7]
2	828	1232	148.79	600.00	[10, 3, 0.7, 0.5]	1432	172.95	600.00	[10, 3, 0.3, 0.3]	1527	184.42	600.00	[20, 3, 0.7, 0.5]
3	1001	1831	182.92	600.00	[10, 3, 0.3, 0.7]	1320	131.87	600.00	[20, 1, 0.7, 0.5]	2234	223.18	21.84	[10, 2, 0.5, 0.7]
4	1716	2049	119.41	600.00	[10, 3, 0.3, 0.7]	2040	118.88	600.00	[20, 3, 0.7, 0.5]	2562	149.30	600.00	[20, 3, 0.7, 0.5]
5	1143	1853	162.12	330.37	[20, 2, 0.7, 0.3]	1982	173.40	124.54	[10, 2, 0.5, 0.3]	2883	252.23	600.00	[10, 3, 0.3, 0.7]
6	1950	2969	152.26	600.00	[20, 3, 0.7, 0.5]	3060	156.92	600.00	[10, 3, 0.7, 0.3]	4378	224.51	600.00	[10, 3, 0.5, 0.5]
7	1056	2898	274.43	600.00	[20, 1, 0.7, 0.5]	2504	237.12	600.00	[20, 3, 0.3, 0.7]	3819	361.65	410.24	[20, 1, 0.3, 0.7]
8	1300	5381	413.92	600.00	[20, 3, 0.3, 0.5]	4443	341.77	600.00	[20, 3, 0.3, 0.3]	6761	520.08	600.00	[20, 1, 0.3, 0.3]
9	439	-	-	-	-	-	-	-	-	3989	908.66	600.00	[20, 1, 0.5, 0.5]
10	4631	-	-	-	-	-	-	-	-	9640	208.16	600.00	[20, 2, 0.5, 0.5]
11	3443	7511	218.15	600.00	[20, 1, 0.7, 0.3]	6412	186.23	600.00	[10, 2, 0.5, 0.5]	8934	259.48	600.00	[20, 3, 0.3, 0.3]
12	4040	15907	393.74	600.00	[20, 2, 0.3, 0.3]	13472	333.47	600.00	[20, 3, 0.3, 0.5]	15809	391.31	600.00	[20, 3, 0.5, 0.5]
13	1348	-	-	-	-	-	-	-	-	31014	2300.74	600.00	[10, 1, 0.3, 0.3]
14	1278	-	-	-	-	-	-	-	-	6894	539.44	600.00	[20, 1, 0.3, 0.7]
15	3831	-	-	-	-	-	-	-	-	-	-	-	-
16	3225	-	-	-	-	-	-	-	-	-	-	-	-
17	5745	-	-	-	-	-	-	-	-	-	-	-	-
18	4459	-	-	-	-	-	-	-	-	-	-	-	-
19	3149	-	-	-	-	-	-	-	-	-	-	-	-
20	4769	-	-	-	-	-	-	-	-	-	-	-	-
21	21133	-	-	-	-	-	-	-	-	-	-	-	-
22	30244	-	-	-	-	-	-	-	-	-	-	-	-
23	17428	-	-	-	-	-	-	-	-	-	-	-	-
24	42463	-	-	-	-	-	-	-	-	-	-	-	-

4.4. Results method 3: HHSA

The third method is the HHSA. The HHSA is similar to the HSA. In each iteration of the HSA we create a new solution to the first subproblem. The HHSA aims to further improve the new created solution by performing VNS in some iterations. The only difference between the HSA and HHSA is that we perform a VNS in the HHSA. Because of the VNS we take an extra parameter into account compared to HSA, all the other parameters and their values for which we run HHSA are the same as in Table 4.3. The extra parameter is the VNS parameter, this parameter has the value 0.5. This means that the probability that VNS is performed to improve the solution is 0.5. The value of 0.5 is chosen because first experiments showed that VNS took a lot of time to perform and applying HSA was not always beneficial. Therefore, the decision is made to not perform VNS each iteration. First, we discuss the results of each fitness function then the performance of the HHSA in general in Section 4.4.4. To check the performance of each fitness functions, we obtained the exact objective value of the HHSA with the relaxation and heuristic fitness functions by solving the end solutions of these HHSA exactly and set the exact solution as the solution to the non-exact fitness functions.

4.4.1. Results of method 3 with the exact fitness function

The HHSA with the exact fitness function is not capable to find a feasible solution for instance 13 and instances 15 to 24. For instance 8 and 14, not with all parameter sets a feasible solution is found. For all other instances, we found a feasible solution to the NRP for all parameter sets. For HHSA with a pool size of 10 and the first stopping criterion, the solving time is less than 600 seconds until instance 19, except from instance 13. Except for the first instance, the maximum solving time of 600 seconds is used to find a feasible solution to all instances.

When we perform the HSA with a stopping criterion of 1 or 3, we obtained the best results with a pool size of 20 (for 9 of 13 instance with stopping criteria 1 and 3). For the second stopping criterion, the HSA with a pool size of 10 performs better. If we applied a pool size of 10, the third stopping criterion obtained the most best solutions (9 of 13 instances). For the same pool size, we find the most best solution when HMCR is 0.3, regardless of the value of PAR.

For a pool size of 20 the stopping criteria has no influence on the solutions and we obtained the most best solutions when both PAR and HMCR have a value of 0.3. For the first stopping criterion, we also obtained the most best solutions when both PAR and HMCR have a value of 0.3. The same holds for HMCR for the second stopping criterion, the value of PAR has no influence. Method 3 with the third stopping criterion performs best when HMCR is 0.3 or 0.5, the value of PAR is not of influence.

The best results per instance show that there is no large difference between a pool size of 10 or 20 and stopping criteria 1 and 2 (both 6 instances). The third stopping criterion obtains the best solutions for two of 13 instances. When PAR and HMCR have both a value of 0.3, the most best objective values are found. The best solution we found for the first instance is almost equal to the best known solution: we obtained a best objective value of 612 while the best known objective value is 607. For the fourth instance we obtained a relative good result: we find an objective value of 1883 against the best known value of 1716. For all other instances, we find a solution that differs more than 47 percent from the best known solution.

4.4.2. Results of method 3 with the relaxation fitness function

The HHSA with the relaxation fitness function finds a feasible solution for the same instances as the exact fitness function. With the third stopping criterion, we did not find a feasible solution for instance 2. If we find a feasible solution to the first instance, we obtain this in less than 600 seconds for the first two stopping criteria. For all stopping criteria, we find the most best solutions when PAR has a value of 0.3. The value of 0.3 for both PAR and HMCR, results in the most best solutions for the first and second stopping criteria. For the second stopping criterion, a value of 0.7 to the PAR parameter is better. For method 3 with a pool size of 10 and 20 and the relaxation fitness function, a value of 0.3 to HMCR performs best or equal compared to other values of HMCR for all stopping criteria. With a pool size of 10, HSA performs best when PAR is equal to 0.3. Approximately the same holds with a pool size of 20, with the second stopping criterion 0.5 is the best value to the PAR parameter.

The HHSA with a pool size of 20 performs better than 10 for the first and third stopping criteria: for the second stopping criterion this is indifferent. For both pool sizes the second stopping criterion obtains the most best solution for the HSA.

When we only look at the best results per instance, we find for 10 of 13 instance the best solution when the pool size is 20. We obtained the most best solutions when the first stopping criterion is applied (7 of 13 instances) but this does not differ much from the second stopping criterion (5 of 13 instances). For four of 13 instances, the best solutions are obtained when both PAR and HMCR have a value of 0.3. For three of 13 instance the combination of 0.5-0.3 or 0.7-0.3 for the combination of PAR-HMCR gives the best solutions. The HHSA with the relaxation fitness function obtains even better results for the first (609) and fourth instance (1742). For all other instances, the found objective value is much higher than the best known solution. The worst solution compared to the best known objective value we find is the solution of instance 9 (4554 against the best known: 439).

4.4.3. Results of method 3 with the heuristic fitness function

The HHSA with a heuristic fitness function is similar to the exact and relaxation fitness function capable of finding a feasible solution to the first 12 instances and instance 14. We obtained not for all parameter sets of instance 8 and 14 a feasible solution. For instances with a size larger than the first instance, the HHSA solves the problem in the maximum solving time for the first and second stopping criteria. For the third stopping criterion, the solving time is always 600 seconds since this is the stopping criterion.

Comparing the pool sizes per stopping criteria, we obtained the best results with a pool size of 10 for stopping criteria 1 and 3. For the third stopping criterion, a pool size of 10 obtained the best objective values to the small instance 1,2,3 and 4. For both pool sizes, the different stopping criteria have different influence on the solutions. For a pool size of 10 (respectively 20), 4 (respectively 5) of the solutions are found with the first stopping criterion and 5 (respectively 5) with the second stopping criterion.

For a pool size of 10, a value of 0.3 for the HMCR parameter obtains the best solutions, for the PAR parameter, the best value differs per stopping criteria so the effect of the value of the PAR parameter is not clear. With a pool size of 20 no clear best value for the PAR and HMCR parameters are distinguished. For the third stopping criterion, it is also not possible to distinguish a best value for the PAR and HMCR parameters. For the first stopping criterion, the combination 0.3 for HMCR and 0.3 for the PAR parameter obtains the most best solutions. For the second stopping criterion this is the combination of 0.7 for the PAR parameter and 0.3 for the HMCR parameter.

The best objective values for each instance show that for 11 of 13 instance the HHSA with a pool size of 20 is the best and for 6 of 13 instances, the second stopping criterion is the best. The combination of 0.3 for the PAR parameter and 0.7 for the HMCR parameter results in the most best solutions (4 of 13 instances). The best objective values we find are in general much higher than the best known solutions, only for the first instance we only differ two percent of the best known objective value.

4.4.4. Overall results of the HHSA

The best found solutions of the HHSA are given in Table 4.5. The HHSA finds for 13 instances feasible solution to the NRP. In the table we see that again the most best found solutions for these instances are found by using the relaxation function. The average deviation rate of the relaxation fitness function is 337.8 percent. This is lower than the average deviation rate of the exact fitness function (357 percent) and the heuristic fitness function (622.5 percent). Feasible solutions are found for the first twelve instances and the fourteenth instance, so no feasible solutions for large-size instances.

We are not able to compare the solving time for an instance between fitness functions since some best solutions are found by the running HHSA with the third stopping criterion. For the instance 1,2 and 4, the HHSA finds better solutions than the HSA and for the instance 9, 10 and 14 the HSA was not able to find a feasible solution so for these instances, the HHSA performs better for these three instances. HSA finds better solutions for the other seven instances. The assumption that HSA performs better than HHSA if we run both for 10 minutes is reinforced by the average deviation rate per fitness function. For each fitness function the average deviation rate are lower for the HSA than the HHSA. We look further into the difference between the HSA and HHSA and the different fitness functions in the following section.

Table 4.5: The best objective values of HHSA with corresponding deviation rate, solving time and parameter set per instance per fitness function

		Exact fitness function				Relaxation fitness function				Heuristic fitness function			
Instance	Best known objective value	Best objective value	Deviation rate from best known objective value (%)	Solving time (s)	Parameterset: [pool size, stopping criterion, PAR, HMCR]	Best objective value	Deviation rate from best known objective value (%)	Solving time (s)	Parameterset: [pool size, stopping criterion, PAR, HMCR]	Best objective value	Deviation rate from best known objective value (%)	Solving time (s)	Parameterset: [pool size, stopping criterion, PAR, HMCR]
1	607	612	100.82	541.27	[20, 1, 0.5, 0.7]	609	100.33	61.10	[20, 1, 0.5, 0.3]	618	101.81	549.19	[20, 2, 0.3, 0.7]
2	828	1221	147.46	600.00	[10, 1, 0.7, 0.3]	1737	209.78	365.38	[10, 2, 0.7, 0.7]	1547	186.84	600.00	[10, 2, 0.7, 0.3]
3	1001	2217	221.48	600.00	[10, 2, 0.7, 0.3]	1517	151.55	600.00	[20, 2, 0.7, 0.3]	2527	252.45	600.00	[10, 3, 0.3, 0.7]
4	1716	1833	106.82	600.00	[20, 3, 0.3, 0.3]	1742	101.52	600.00	[20, 1, 0.7, 0.3]	3150	183.57	600.00	[20, 3, 0.7, 0.3]
5	1143	2380	208.22	600.00	[10, 1, 0.3, 0.3]	2048	179.18	600.00	[10, 2, 0.3, 0.3]	3283	287.23	600.00	[20, 2, 0.5, 0.7]
6	1950	4577	234.72	600.00	[20, 2, 0.5, 0.3]	3771	193.38	600.00	[20, 3, 0.5, 0.3]	4995	256.15	600.00	[20, 3, 0.5, 0.7]
7	1056	4230	400.57	600.00	[20, 2, 0.3, 0.5]	3517	333.05	600.00	[20, 1, 0.3, 0.3]	4620	437.50	600.00	[20, 2, 0.3, 0.3]
8	1300	7492	576.31	600.00	[10, 2, 0.5, 0.3]	6900	530.77	600.00	[20, 1, 0.3, 0.3]	7767	597.46	600.00	[20, 1, 0.3, 0.7]
9	439	4904	1117.08	600.00	[10, 2, 0.7, 0.7]	4553	1037.13	600.00	[20, 2, 0.7, 0.3]	4995	1137.81	600.00	[20, 2, 0.3, 0.7]
10	4631	12263	264.80	600.00	[20, 2, 0.3, 0.3]	11734	253.38	600.00	[20, 1, 0.3, 0.3]	12447	268.78	600.00	[20, 2, 0.3, 0.5]
11	3443	9534	276.91	600.00	[20, 1, 0.3, 0.3]	9927	288.32	600.00	[20, 2, 0.5, 0.7]	9148	265.70	600.00	[20, 3, 0.3, 0.7]
12	4040	16924	418.91	600.00	[20, 3, 0.3, 0.3]	17375	430.07	600.00	[10, 1, 0.5, 0.3]	16796	415.74	600.00	[20, 1, 0.5, 0.3]
13	1348	-	-	-	-	-	-	-	-	-	-	-	-
14	1278	7237	566.28	600.00	[10, 1, 0.3, 0.5]	7446	582.63	600.00	[20, 2, 0.5, 0.7]	8974	702.19	600.00	[20, 1, 0.3, 0.3]
15	3831	-	-	-	-	-	-	-	-	-	-	-	-
16	3225	-	-	-	-	-	-	-	-	-	-	-	-
17	5745	-	-	-	-	-	-	-	-	-	-	-	-
18	4459	-	-	-	-	-	-	-	-	-	-	-	-
19	3149	-	-	-	-	-	-	-	-	-	-	-	-
20	4769	-	-	-	-	-	-	-	-	-	-	-	-
21	21133	-	-	-	-	-	-	-	-	-	-	-	-
22	30244	-	-	-	-	-	-	-	-	-	-	-	-
23	17428	-	-	-	-	-	-	-	-	-	-	-	-
24	42463	-	-	-	-	-	-	-	-	-	-	-	-

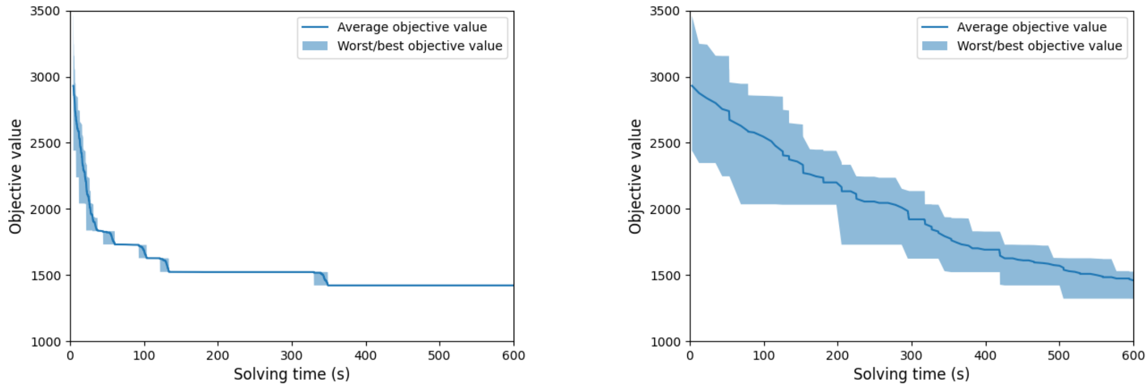
4.5. Extra results to gain more insights in the methods

To compare the HSA with the HHSA in more detail, we performed an extra experiment. This experiment runs both the HSA and the HHSA for the same instances and the same parameter set. The exact fitness function is used. In both the HSA and the HHSA, each iteration generates a new solution and the best, worst and mean objective value of this solution pool are known. In Figure 4.1 and 4.2b the best, worst and mean objective value over time are given for instance 2 and instance 7. For both instances the objective value of the HSA decreases faster than the HHSA.

At instance 2, around 300 seconds the solution pool of the HHSA has a mean objective value of 2000 while the mean of the HSA solution pool decreases to 2000 in approximately 20 seconds. At the seventh instance, the objective value decreases slower for both the HSA and the HHSA than at the second instance. The initial best objective value of HHSA of instance 7 is around 5800 and after 600 seconds the objective value is 5300. The HSA starts with the same initial best objective value, after 600 seconds the best objective value is approximately 4000.

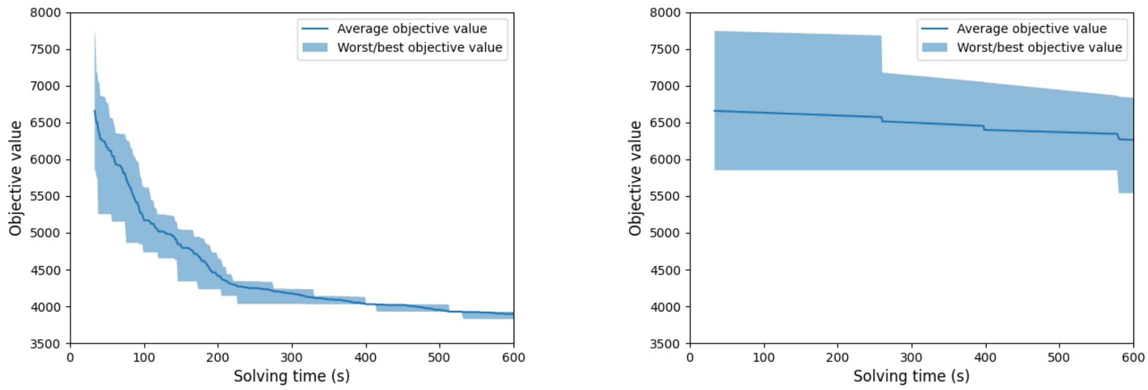
The bad performance of the HHSA indicates that the VNS is not capable of improving the solution. Another reason for the worse performance of HHSA could be that it takes too much time to perform the VNS and therefore less iterations are done. During each iteration of the VNS, the quality of the new solution is determined using the fitness function. Since the VNS consists of several iterations and each iteration we determine the quality of the new solution, the fitness value is calculated several times. If we do not apply the VNS, we determine the quality of the new solution only once, after we created a new solution.

Figure 4.1: The average objective value and the best/worst objective values of the solutions in the solution pool over time while using the HSA (a) and the HHSA (b) for Instance 2



(a) HSA **(b)** HHSA

Figure 4.2: The average objective value and the best/worst objective values of the solutions in the solution pool over time while using the HSA (a) and the HHSA (b) for Instance 7



(a) HSA **(b)** HHSA

To determine how much time it takes to determine the quality of a solution we performed the following experiment. For each instance we generated ten random solutions to the first subproblem. If it took longer than 10 minutes to generate these ten solutions, we did not perform the experiment for this instance. We determined the quality of each solution by using the different fitness functions. For each solution, we determined the fitness value and the corresponding solving time, see Appendix E. The mean objective value and mean solving time per instance and fitness function are given in Table 4.6.

We compare the exact fitness value with the relaxation and heuristic fitness values and look into the solving time per fitness function. The exact fitness values are marked bold, if the relaxation or heuristic fitness values are the same these are also marked bold. Note that this experiment is performed with 10 random solution so this only gives an indication about the performance of the fitness functions.

For the first instance the same objective values are obtained for all solutions so the mean fitness function are equal. For this instance the heuristic fitness function performs best since it obtained the same result but solves the solutions the fastest (0.04 seconds). For instance 2, 3, 4, 5, 8, 10, 11, 12, 14, 15, 16, 17, 18, 19 and 21 the relaxation fitness function obtained the same fitness values as the exact fitness function. In general, the relaxation fitness functions obtained these results in less solving time than the exact fitness function. Therefore, this experiment indicates that the relaxation fitness performs best for the instances 2, 3, 4, 5, 8, 10, 11, 12, 14, 15, 16, 17, 18, 19 and 21. The instances 6,7 and 8 can be solved by the relaxation fitness function since the mean fitness value does not differ that much from the mean exact fitness value. A disadvantage of the relaxation fitness function is that the relaxation assumes that employees can be assigned to part of a shift. In practise this in not possible so the relaxation fitness function can give infeasible solutions. This is the case for instance 13. All the solutions are infeasible as indicated by the exact fitness function but the relaxation fitness function gives a fitness value for all solutions.

In general, the heuristic fitness function performs worse compare to the other fitness functions. This is because of the heuristic is not capable of obtaining a good indication of the objective value. The found average fitness values differ a lot from the average exact fitness values. The solving times are comparable to the solving times of the relaxation fitness function for feasible solutions.

For the instances 8, 13 to 19 and 21 only infeasible solutions are generated. For the instance 14 to 19 and 21 all fitness functions determine that the solutions are infeasible but the relaxation fitness function is the fastest if we neglect instance 15 and 21. For the instances 13 to 19 and 21 the relaxation and exact fitness function bot have disadvantages and advantages. The disadvantage of the exact fitness function is that the solving time is higher in general. The advantage is that the obtained fitness value is the true objective value. The advantage of the relaxation fitness function is that it determines the quality faster than the exact fitness function in general but does not guarantee that the solution is feasible. In general, the heuristic fitness function is slower than the other functions when the solutions are infeasible. We were not able to generate ten random solutions for the three largest instance and instance 20, therefor the experiment was not performed for these instances.

In Table 4.7, the duration of generating ten random solution to the first subproblem by using the construction heuristic are given per instance. Since the ten solutions are generated randomly this gives an indication of the duration, the real duration can deviate. The time of generating the solutions are acceptable for the first 15 instances. For the instances 16 to 19 this takes between half a minute and a little bit more than a minute. For the five largest instances generating ten solutions takes too long. Since the maximum solving time is 10 minutes for the last 5 instances there is no time left to find a good feasible solution. This could a reason why we do not find any feasible solution for the largest instances with the second and third method.

In Sections 4.3.4 and 4.4.4, we concluded that the algorithm with the relaxation fitness function performed best. The reason for this is that the relaxation fitness function is capable to give an indication of the objective value is reasonable time. Because of this, method 2 or 3 with the relaxation fitness function can perform more iterations in the same time than with the exact fitness function. To illustrate this, we calculate the theoretical number of iterations of HSA with a pool size of 10 and the third stopping criterion. We use the times of Tables E.1 and 4.7. If we neglect the time to generate a new solution with one of the considerations we are able to perform 54 iterations with the exact fitness function and 285 iterations with the heuristic fitness function. Since we are able to generate more iterations the algorithm creates more new solutions and therefore the algorithm has a higher probability to find a better solution.

To determine what the effect of the maximum computation time is, two instances are solved using the HSA with the relaxation fitness function with a maximum computation time of 3600. The pool size is set to 20 solutions, the value of the PAR variable to 0.5 and 0.3 for the value of the HMCR parameter. The instances 14 and 20 are solved for all three stopping criteria. For both instances, an infeasible solution was found with the first stopping criterion. The solving time is 600 seconds. For the instance 14 we obtained an objective value of 10492 with the second stopping criterion and 11187 for the third stopping criterion. The solving time is for all stopping criteria 3600 seconds. The best known objective value is 1278, so the deviation rate is 820.97 percent for the second stopping criterion and 875.35 for the third stopping criterion. This deviation rates are worse than the deviation rates of the best solutions of method 2 and 3.

The results of instance 20 show that no feasible solutions could be obtained with all stopping criteria. The solving time for all stopping criteria is 3600 seconds. The reason that no feasible solution could be obtained is that the construction heuristic was not capable of generating an initial solution pool within 3600 seconds.

Table 4.6: Average objective values and solving time of solving 10 random solutions to the first subproblem with the exact, relaxation and heuristic fitness function. An infeasible solution is given by -

Instance	Exact fitness function		Relaxation fitness function		Heuristic fitness function	
	Objective value	Solving time (s)	Objective value	Solving time (s)	Objective value	Solving time (s)
1	**2138.10**	0.20	**2138.10**	0.14	**2138.10**	0.04
2	**2951.90**	0.43	**2951.90**	0.10	3978.30	0.06
3	**4004.20**	0.64	**4004.20**	0.23	5074.80	0.07
4	**4392.30**	0.43	**4392.30**	0.14	6409.80	0.18
5	**5119.60**	0.51	**5119.60**	0.21	6835.50	0.17
6	**6666.50**	0.65	6639.10	0.18	11255.20	0.14
7	**6796.70**	0.71	6756.20	0.16	9334.00	0.12
8	-	1.18	-	0.35	-	2.06
9	**7435.60**	1.31	7425.10	0.26	12170.30	0.21
10	**14830.30**	2.50	**14830.30**	0.65	25407.50	0.40
11	**13119.60**	3.38	**13119.60**	0.73	34620.60	0.31
12	**19675.50**	9.31	**19675.50**	2.02	43254.00	0.47
13	-	461.72	35317.60	261.33	-	4.88
14	-	2.23	-	0.75	-	5.83
15	-	4.82	-	29.60	-	5.39
16	-	1.15	-	0.34	-	6.16
17	-	3.02	-	1.29	-	6.58
18	-	2.06	-	0.64	-	11.47
19	-	9.02	-	3.34	-	14.40
21	-	97.95	-	785.79	-	66.42

Table 4.7: The duration of generating ten random solutions to the first subproblem for each instance

Instance	Time (s)	Instance	Time(s)
1	0.06	13	3.28
2	0.08	14	3.52
3	0.12	15	6.37
4	0.17	16	23.85
5	0.44	17	33.69
6	0.62	18	65.87
7	0.73	19	68.20
8	0.78	20	>600
9	0.90	21	582.57
10	0.94	22	>600
11	2.67	23	>600
12	2.80	24	>600

5

Conclusion and discussion

In this chapter conclusions about the methods and the decomposition of the problems are presented. Further research and possible adjustments to the models are also discussed. First, we discuss each method individually. Subsequently, we discuss the general approach such as the decomposition we used.

5.1. Method 1: Solving the subproblems sequentially exactly

The first method used in this thesis solves the two subproblems sequentially to optimality. The first subproblem determines which day each employee works, the second subproblem assigns shifts to employees on their working days. The solution of the first subproblem is the input of the second subproblem. The MIP-formulations of both subproblems are explained in Sections 3.1 and 3.2. The objective value of the first subproblem is to minimise under- and overcoverage at day level. The second subproblem aims to minimise the under- and overcoverage at shift level and stimulates to grant the preferences of the employee. The objective value and the solution of the second subproblem are equal to those of the NRP. A challenge of the proposed decomposition is that an optimal solution to the first subproblem is not necessarily the optimal solution to the NRP. To tackle this challenge and handle the symmetry in the first subproblem, three extensions to the first subproblem are proposed and examined.

The results of the first method show that a combination of the extensions is necessary to find a feasible solution to the NRP. When the first subproblem is solved to optimality with or without one extension, no feasible solutions are found for all instances of the benchmark instances. When all extensions are applied, we find feasible solutions to the NRP for 10 instances. The objective values of these solutions are higher than the best known solutions of the instances. The best obtained objective value is 61.96 percent more than the best known objective value. The worst obtained objective value is 1032.42 percent more than the best known objective value. The first method also obtained feasible solutions to the NRP when the combination of the first and third extensions was applied. This found feasible solutions to 11 instances. In general, the first method with all extensions performs slightly better than the first method with the combination of first and third extension.

For the four largest instances, the first method is not able to find a solution to the first subproblem due to the maximum solving time of 10 minutes. For the other instances, the first method finds solutions in a reasonable time but the solutions are infeasible or of bad quality. Because the NRP is a NP-hard problem, it can be expected that that the first method is not capable of finding feasible solutions to large-size instances such as instances 20 to 24 in limited time.

The solving time of the first subproblem is higher than the solving time of the second subproblem. The solving times of the first method are lower than the solving times of the compare method. The objective values found are higher than the objective values of the compare method. The first method is capable of finding solutions to more instances than the compare method.

When we compare the results of the first method with the results of the compare method, we can answer the second research question ”*Is the solving time of the NRP reduced by decomposing the NRP into a days-on/off assignment subproblem and shift assignment subproblem?*” with yes: solving the NRP in two phases reduces the solving time. The third research question ”*When solving the decomposed NRP, is the value of the solution of the NRP not more than 20 percent above the best known objective value?*” has to be answered negatively. The found objective values are often more than 20 percent worse than the solutions of the compare method and the best known objective values.

The results of the first method provide insight in the proposed decomposition. The advantage of the proposed decomposition is that the complexity of solving the NRP is reduced. A disadvantage is that the quality of the found solutions is decreased because of the decomposition. Because of the decomposition, both subproblems miss information about the entire NRP. This manifests mostly in the first subproblem, since the constraints of the second subproblem are not taken into account. The constraints of the first subproblem are taken into account in the second subproblem since the solution of the first subproblem is the input of the second subproblem. This loss of information in the first subproblem causes that the quality of the solution to the NRP is lower than when the NRP is not decomposed. The extensions we use in the first method stimulate that constraints of the second subproblem are partly taken into account in the first subproblem. This stimulates that the first subproblem finds a solution that gives a feasible solution to the second subproblem and thereby the NRP. The results of the first method show that the addition of extensions are a promising method to compensate for the loss of information when the NRP is decomposed. In further research, we could look further into the addition of extensions to a decomposed problem. For the NRP, we could also add an extension to take the shift sequence into account. The extension could for example make sure that an employee can work certain shifts on the working days, such that the sequence constraint and the maximum shift number are not exceeded. The weights of the current extensions show that they are capable of finding feasible solutions, but more research on combinations of weights can be carried out.

In short, method 1 is not able to find reasonable solutions to the NRP. The first method does not have a good trade-off between the solving time and the quality of the solutions. The addition of extensions improves the found solutions and are promising for further research.

5.2. Method 2: HSA

The second method is the HSA, this algorithm starts with an initial pool of solutions. In each iteration the algorithm creates new solutions to improve the solutions in the solution pool. The HSA terminates when one of the stopping criteria is met or the maximum solving time is reached. The initial solution pool is generated by a construction heuristic and the new solutions are obtained by one of the three considerations: random, memory or pitch. To determine the quality of the new solution we use the exact, relaxation or heuristic fitness functions.

Method 2 with the exact and relaxation fitness functions found feasible solutions for 10 instances: with the heuristic fitness function we found feasible solutions for 14 instances. The instances for which we found a feasible solution are all small or medium-size instances. No feasible solutions are obtained for large-size instances. The HSA performs best with the relaxation fitness function with a pool size of 20 and the third stopping criterion (the algorithm runs until the maximum solving time is reached). The values for PAR and HMCR for the largest instances for which we find the best feasible solution are 0.3 and 0.5. This shows that the HSA performed best with the parameter set of [pool size, stopping criteria, PAR, HMCR] = $[20, 3, 0.3, 0.5]$ and the relaxation fitness function.

The solving time of method 2 was for almost all instances 600 seconds, partly because most of the best solutions are found with the third stopping criterion. These solving times are comparable to the solving times of the compare method. The best objective values differ from the best known solutions. The best solution with the relaxation fitness function we found has a deviation rate of 118.88 percent. The other found objective values differ more from the best known objective values. From the obtained results we can answer the third research question: *When solving the decomposed NRP, is the best found objective value of the solution of the NRP not more than 20 percent above the best known objective value?*. For the first instance the best found objective value differs not more than 20 percent of the best known objective value. For all other instances, the best found objective value is more than 20 percent above the best known objective value. The solving times of method 2 is comparable to the solving times of the first method. The trade-off between the solving time and the quality of the solutions is better for method 2 than for method 1 since the found objective values of method 2 are better than for method 1.

In method 2, each iteration creates a new solution to the first subproblem. In the first method only one solution to the first subproblem is considered. Since multiple solutions to the first subproblem are considered in method 2, the found solutions vary more. This could be a reason why we find better solutions when using method 2. The creation of multiple solutions to the first subproblem comes at a cost. For each new solution, the quality of this solution is determined. Since the time to determine the quality of a new solution increases as the size of the instance increases, an iteration takes more time when the size of the instance increases. Determining the quality of the solution with the exact fitness function takes the most time compared to the other fitness functions. An indication of the objective value is provided by the relaxation or heuristic fitness function. These fitness functions take less time to determine the quality of a solution. The relaxation fitness function approaches the objective value accurately but we have to be aware that the relaxation fitness function can find a fitness value, even when the solution is infeasible. As shown in Table 4.6, this happens only for 1 of the 9 instances in our experiment.

The HSA with heuristic fitness function finds for most instances an upper bound to the objective value in a short time. For infeasible solutions the solving time was high compared to the other fitness functions. The results of the HSA show that the HSA with the heuristic fitness function does obtain solutions with high deviation rates from the best known results. As shown in Table 4.6, the obtained objective values are much higher than the objective values obtained with the exact fitness function. Therefore, the heuristic fitness function is not capable of obtaining an upper bound that is a good indication of the exact objective value. Since this upper bound differs a lot from the objective value, the heuristic fitness function does not give a good indication of the objective value. As explained in Section 3.4.2 first another construction heuristic is used as heuristic fitness function. This construction heuristic was not capable of finding a feasible solution to the second subproblem. The proposed heuristic fitness function heuristically assigns shifts to employees while taking the maximum number of shifts and minimum working hours into account in a smart way. In further research, we could search for a learning heuristic fitness function that smartly assigns shifts to employees while taking previous iterations into account.

In short, method 2 performs better than method 1. The solving times are longer than the solving times of the first method but better solutions are found. Despite the better results, the best found solutions differ more than 20 percent from the best known objective values. Method 2 gives us insights in the different fitness functions and their capability to determine the quality of a new solution. The exact fitness function is reliable but relatively slow. The relaxation fitness function is a good fitness function to determine the quality relatively fast and with good quality. The proposed heuristic fitness function is not suitable for giving an accurate indication of the objective value.

5.3. Method 3: HHSA

The third method is approximately the same as the HSA method, only VNS is added to the HSA to improve the newly created solutions. The results show that the HHSA performs worse than the HSA. The HHSA is not capable of finding reasonable results to the NRP, especially not for large size instances.

A possible reason for this is that it takes a lot of time to perform VNS because the quality of the solutions are determined in each iteration of the VNS. Because of this, each iteration of the HHSA takes more time compared to each iteration of method 2 and therefore fewer iterations are performed in the HHSA. If the VNS is capable of finding a much better solution, less iterations are necessary to find a good solution. Our research shows that the effect of the VNS is minimal and the VNS is not capable of finding much better solutions. Therefore, there is no positive trade-off between the time VNS takes and the capability of finding a better solution. For HHSA with the relaxation and heuristic fitness functions, worse solutions than the HSA were found. Other neighbourhoods in the VNS might be capable of finding a better solution such that the trade-off between the time and quality is more positive. When applying a new neighbourhood, the time it takes to perform this neighbourhood has to be taken into account. We should consider other neighbourhoods than proposed in the literature, since the neighbourhoods we applied in the VNS are based on the literature. Another local search method that does not determine the quality of the solution in each iteration or needs fewer iterations to improve the solution could be worth researching. One local search method that could improve the solution in fewer iterations is Tabu search. In this thesis we obtained worse results when we applied Tabu search, so there is no clear indication that this would improve the HHSA.

To conclude, method 3 is in general not capable of finding a solution that is that differs at most more than 20 percent from the best known solution. Also the time to find these solutions are worse than for method 2. This makes the HHSA not a good method to solve the decomposed NRP. Using VNS is not beneficial as the time it takes is not justified by the improvement in quality. For the proposed decomposition of the NRP, it is difficult to apply a local search method that is capable of escaping a local minimum in a reasonable time.

5.4. Decomposition of the NRP and the chosen methods

In this thesis, we composed the NRP into two subproblems. The first subproblem assigns employee to working days. The second subproblem assigns employees to shifts on their working days. The solution to the second subproblem is the solution to the NRP. This decomposition was proposed by several researches and provided good results. A challenge of this decomposition is that an optimal solution to the first subproblem does not have to be the optimal solution to the NRP. This has to do with that information is lost when decomposing the problem. As shown in the results of the first method, decomposing a problem can reduce the complexity of solving the problem but this comes at a cost of losing information. As shown by the results of method 1, extension are promising to compensate for the loss of information without losing the decrease of complexity. The extensions provide information about the second subproblem in the first subproblem, such that less information is lost. The addition of extensions can be applied to several other decomposed problems.

From the results of the first method we derive that it is more complex to find a feasible solution to the first subproblem than to the second subproblem. In the proposed HSA we generate and improve feasible solutions to the first subproblem. The quality of this solution is determined by the objective value of the second subproblem. The solutions of the first subproblem are generate randomly or constructed by adjusting solutions. Solving the NRP with a HSA suits the structure of the decomposed NRP since the focus of the HSA is not on solving the first subproblem.

Despite the good structure of the HSA to the decomposed NRP, the HSA is not capable of finding good solutions to the NRP. We are not able to escape the local minima and find the global minimum, even not when we add VNS to the HSA. Since the first subproblem has no information if a solution is a good solution to the NRP, it is hard to escape the local minima. With method 3, HSA with VNS, we are not capable of finding reasonable solutions to the NRP. The reason for this is that the VNS takes too much time and is not able to escape the local minima in this time. The HSA is a good method for problems with this structure. However, the HSA is not able to find the global minimum. To escape the local minimum we could add another local search method or use another pitch consideration. However, method 3 showed that adding a local search algorithm to the HSA is time consuming and did not improving the results.

In the HSA we use a construction heuristic to initialise the solution pool. As shown in Table 4.7, it took a lot of time to generate an initial solution pool for the larger instances of the NRP. This is a reason why we are not able to find a feasible solution to the NRP for the large instances with method 2 and 3. When generating the initial solution pool takes a lot of time, less time is left to create new solutions and thereby improve the solutions in the solution pool. Especially for the large size instances, finding a feasible solution to the first subproblem is difficult. One of the reasons for this is the structure of the benchmark instances. For example, in benchmark instance 23 some employees are not allowed to work a weekend in a year. With the random nature of the construction heuristic, this constraint is hard to grant. For instances with less complex constraints the proposed construction heuristic is more suitable. To stimulate that the construction heuristic generates solutions of good quality, the third extension of method 1 is taken into account. The generated pattern of working days takes the minimum and maximum number of working days according to the contract hours into account. This extension in the construction heuristic stimulates that the generated solution is a feasible solution to the NRP. By taking this extension into account more time is needed to generating a solution but this results in a solution of higher quality. In further research, we could research the effect of adding more extensions to the construction heuristic and thereby improving the quality of the solution.

The current results are obtained with a maximum solving time of 10 minutes. By increasing the time limit, we could maybe find feasible solutions for more instances. However, the results of running instances 14 and 20 for one hour indicate that increasing the time limit does not improve the solution.

In this thesis the methods are examined against the benchmark instances. As explained before, the benchmark instances are hard to solve and there are only 24 instances available. To further validate the results of the proposed methods, the methods can be examined against more instances. To compare the methods against more instances, the best objective values of these instances have to be known. The objective values can be known from other researches or by solving the instances exactly. The NRP is a $\mathcal{NP}$-hard problem, which prohibits exact solutions in reasonable time, especially for large-size instances. Currently, there are no other instances in the literature for the considered NRP research, we could not examine the methods against other instances.

To conclude this research, we answer the main research question of this thesis: *Which method to* *solve the subproblems gives the best trade-off between the solving time and the quality of the solution when solving the decomposed NRP?*. HSA (method 2) gives the best trade-off between the time and the quality but still does not perform as desired. The goal is to find solutions to the NRP that are not more than 20 percent of the best solutions. That goal is not achieved with the current methods. Although the obtained results are not as desired, the results give useful insights in the proposed decomposition and its characteristics. Thereby, more insight is gained about the NRP. Further research could focus on the effects of extensions on decomposed problems or on other construction heuristics to generate the initial solution pool of the second subproblem.

References

Abdelghany, M., Yahia, Z., & Eltawil, A. B. (2021). A new two-stage variable neighborhood search algorithm for the nurse rostering problem. *RAIRO: Operations Research*, *55*, 673–687.

Aickelin, U., & Dowsland, K. A. (2000). Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem. *Journal of scheduling*, *3*(3), 139–153.

Awadallah, M. A., Al-Betar, M. A., Khader, A. T., Bolaji, A. L., & Alkoffash, M. (2017). Hybridization of harmony search with hill climbing for highly constrained nurse rostering problem. *Neural Computing and Applications*, *28*(3), 463–482.

Awadallah, M. A., Bolaji, A. L., & Al-Betar, M. A. (2015). A hybrid artificial bee colony for a nurse rostering problem. *Applied Soft Computing*, *35*, 726–739.

Bai, R., Burke, E. K., Kendall, G., Li, J., & McCollum, B. (2010). A hybrid evolutionary approach to the nurse rostering problem. *IEEE Transactions on Evolutionary Computation*, *14*(4), 580–590.

Baxter, J., & Mosby, M. (1988). Generating acceptable shift-working schedules. *Journal of the Operational Research Society*, *39*(6), 537–542.

Brucker, P., Burke, E. K., Curtois, T., Qu, R., & Berghe, G. V. (2010). A shift sequence based approach for nurse scheduling and a new benchmark dataset. *Journal of Heuristics*, *16*(4), 559–573.

Burke, E., De Causmaecker, P., & Berghe, G. V. (1998). A hybrid tabu search algorithm for the nurse rostering problem. *Asia-Pacific Conference on Simulated Evolution and Learning*, 187–194.

Burke, E., De Causmaecker, P., Petrovic, S., & Berghe, G. V. (2003). Variable neighborhood search for nurse rostering problems. *Metaheuristics: Computer decision-making* (pp. 153–172). Springer.

Burke, E. K., & Curtois, T. (2014). New approaches to nurse rostering benchmark instances. *European Journal of Operational Research*, *237*(1), 71–81.

Burke, E. K., Curtois, T., Post, G., Qu, R., & Veltman, B. (2008). A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem. *European journal of operational research*, *188*(2), 330–341.

Ceschia, S., Thanh, N. D. T., Causmaecker, P. D., Haspeslagh, S., & Schaerf, A. (2015). Second international nurse rostering competition (INRC-II) - problem description and rules -. *CoRR*, *abs/1501.04177*.

Curtois, T. (n.d.). Nurse rostering benchmark instances. Retrieved September 21, 2021, from http://www.schedulingbenchmarks.org/nrp/

Curtois, T., & Qu, R. (2014). Computational results on new staff scheduling benchmark instances. *tech. report*.

Gong, J., Zhang, Z., Liu, J., Guan, C., & Liu, S. (2021). Hybrid algorithm of harmony search for dynamic parallel row ordering problem. *Journal of Manufacturing Systems*, *58*, 159–175.

Hadwan, M., Ayob, M., Sabar, N. R., & Qu, R. (2013). A harmony search algorithm for nurse rostering problems. *Information Sciences*, *233*, 126–140.

Haspeslagh, S., De Causmaecker, P., Stølevik, M., & Schaerf, A. (2010). First international nurse rostering competition 2010. *PATAT 2010-Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling, Belfast, Northern-Ireland, UK*.

Jaumard, B., Semet, F., & Vovor, T. (1998). A generalized linear programming model for nurse scheduling. *European journal of operational research*, *107*(1), 1–18.

Kim, S.-J., Ko, Y.-W., Uhmn, S., & Kim, J. (2014). A strategy to improve performance of genetic algorithm for nurse scheduling problem. *International journal of software engineering and its applications*, *8*(1), 53–62.

Lau, H. C. (1996). On the complexity of manpower shift scheduling. *Computers & operations research*, *23*(1), 93–102.

Liu, Z., Liu, Z., Zhu, Z., Shen, Y., & Dong, J. (2018). Simulated annealing for a multi-level nurse rostering problem in hemodialysis service. *Applied Soft Computing*, *64*, 148–160.

Margot, F. (2010). Symmetry in integer linear programming. *50 Years of Integer Programming 1958-2008*, 647–686.

Osogami, T., & Imai, H. (2000). Classification of various neighborhood operations for the nurse scheduling problem. *International Symposium on Algorithms and Computation*, 72–83.

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

Rahimian, E., Akartunalı, K., & Levine, J. (2017). A hybrid integer programming and variable neighbourhood search algorithm to solve nurse rostering problems. *European Journal of Operational Research*, *258*(2), 411–423.

Santos, H. G., Toffolo, T. A., Gomes, R. A., & Ribas, S. (2016). Integer programming techniques for the nurse rostering problem. *Annals of Operations Research*, *239*(1), 225–251.

Smet, P. (2018). Constraint reformulation for nurse rostering problems. *Proceedings of the 12th international conference on the practice and theory of automated timetabling*, 69–80.

Solos, I. P., Tassopoulos, I. X., & Beligiannis, G. N. (2013). A generic two-phase stochastic variable neighborhood approach for effectively solving the nurse rostering problem. *Algorithms*, *6*(2), 278–308.

Strandmark, P., Qu, Y., & Curtois, T. (2020). First-order linear programming in a column generation-based heuristic approach to the nurse rostering problem. *Computers & Operations Research*, *120*, 104945.

Tassopoulos, I. X., Solos, I. P., & Beligiannis, G. N. (2015). A two-phase adaptive variable neighborhood approach for nurse rostering. *Computers & Operations Research*, *60*, 150–169.

Valouxis, C., Gogos, C., Goulas, G., Alefragis, P., & Housos, E. (2012). A systematic two phase approach for the nurse rostering problem. *European Journal of Operational Research*, *219*(2), 425–433.

Warner, D. M. (1976). Scheduling nursing personnel according to nursing preference: A mathematical programming approach. *Operations Research*, *24*(5), 842–856.

Zhuo, X., Huang, H., Cai, Z., & Hu, H. (2015). An hybrid evolutionary algorithm with scout bee global search strategy for chinese nurse rostering problems. *2015 IEEE Congress on Evolutionary Computation (CEC)*, 769–775.

A

Example benchmark instance 1

```
# This is a comment. Comments start with #
SECTION_HORIZON
# All instances start on a Monday
# The horizon length in days:
14

SECTION_SHIFTS
# ShiftID, Length in mins, Shifts which cannot follow this shift |
    separated
D,480,

SECTION_STAFF
# ID, MaxShifts, MaxTotalMinutes, MinTotalMinutes, MaxConsecutiveShifts,
    MinConsecutiveShifts, \newline MinConsecutiveDaysOff, MaxWeekends
A,D=14,4320,3360,5,2,2,1
B,D=14,4320,3360,5,2,2,1
C,D=14,4320,3360,5,2,2,1
D,D=14,4320,3360,5,2,2,1
E,D=14,4320,3360,5,2,2,1
F,D=14,4320,3360,5,2,2,1
G,D=14,4320,3360,5,2,2,1
H,D=14,4320,3360,5,2,2,1

SECTION_DAYS_OFF
# EmployeeID, DayIndexes (start at zero)
A,0
B,5
C,8
D,2
E,9
F,5
G,1
H,7

SECTION_SHIFT_ON_REQUESTS
# EmployeeID, Day, ShiftID, Weight
A,2,D,2
A,3,D,2
B,0,D,3
B,1,D,3
B,2,D,3
B,3,D,3
B,4,D,3
```

```
C,0 ,D,1
C,1 ,D,1
C,2 ,D,1
C,3 ,D,1
C,4 ,D,1
D,8 ,D,2
D,9 ,D,2
F,0 ,D,2
F,1 ,D,2
H,9 ,D,1
H,10 ,D,1
H,11 ,D,1
H,12 ,D,1
H,13 ,D,1

SECTION_SHIFT_OFF_REQUESTS
# EmployeeID , Day, ShiftID , Weight
C,12 ,D,1
C,13 ,D,1
F,8 ,D,3
H,2 ,D,3
H,3 ,D,3

SECTION_COVER
# Day, ShiftID , Requirement , Weight for under , Weight for over
0,D,5 ,100 ,1
1,D,7 ,100 ,1
2,D,6 ,100 ,1
3,D,4 ,100 ,1
4,D,5 ,100 ,1
5,D,5 ,100 ,1
6,D,5 ,100 ,1
7,D,6 ,100 ,1
8,D,7 ,100 ,1
9,D,4 ,100 ,1
10,D,2 ,100 ,1
11,D,5 ,100 ,1
12,D,6 ,100 ,1
13,D,4 ,100 ,1
```

B

Results of method 1

Table B.1: Results of method 1 without any extensions.

Instance	Objective value first subproblem	Solving time first subproblem (s)	Objective value second subproblem	Solving time second subproblem (s)
1	1301	1.06	-	0.02
2	700	0.98	-	0.19
3	2000	3.78	-	0.41
4	3404	18.51	-	0.25
5	3509	466.92	-	0.37
6	4011	72.90	-	0.72
7	1600	26.23	-	0.8
8	2300	600.00	-	1.95
9	0	2.53	-	1.58
10	9937	415.70	-	2.64
11	10930	600.00	-	3.2
12	13441	600.00	-	6.75
13	14247	600.00	-	34.87
14	3634	600.00	-	2.77
15	-	600.00	-	-
16	10532	600.00	-	1.56
17	19239	600.00	-	3.72
18	17754	600.00	-	2.77
19	24569	600.00	-	9.36
20	-	600.00	-	-
21	-	600.00	-	-
22	-	600.00	-	-
23	-	600.00	-	-
24	-	600.00	-	-

Table B.5: Results of method 1 with the combination of the working hours and preference extension.

Instance	Weight of working extension	Weight of preference extension	Objective value first subproblem	Solving time first subproblem (s)	Objective value second subproblem	Solving time second subproblem (s)
1	10	5	1234	1.58	-	0.02
		10	1154	3.12	-	0
		15	1074	1.03	-	0.02
	20	5	1244	0.92	-	0.02
		10	1164	1.14	-	0.02

		15	1084	1.06	-	0.02
	50	5	1327	2.58	-	0
		10	1264	3.39	-	0.02
		15	1211	3.09	-	0.02
2	10	5	640	7.48	-	0.18
		10	450	7.80	-	0.19
		15	235	6.95	-	0.19
	20	5	750	2.50	-	0.19
		10	600	6.95	-	0.19
		15	385	7.67	-	0.19
	50	5	1180	7.20	-	0.17
		10	920	3.89	-	0.19
		15	730	7.20	-	0.17
3	10	5	1948	13.19	-	0.25
		10	1763	9.41	-	0.27
		15	1593	14.92	-	0.34
	20	5	2136	16.78	-	0.3
		10	2003	13.00	-	0.3
		15	1747	18.12	-	0.36
	50	5	2340	7.76	-	0.08
		10	2164	24.98	-	0.27
		15	2020	23.94	-	0.28
4	10	5	3319	20.67	-	0.25
		10	3035	33.08	-	0.23
		15	2766	37.39	-	0.25
	20	5	3316	23.75	-	0.31
		10	3135	31.94	-	0.22
		15	2866	33.22	-	0.23
	50	5	3625	33.34	-	0.23
		10	3336	31.61	-	0.25
		15	3136	28.62	-	0.25
5	10	5	3325	600.00	-	0.34
		10	3040	428.03	-	0.34
		15	2814	600.00	-	0.37
	20	5	3554	600.00	-	0.42
		10	3210	285.68	-	0.41
		15	3019	600.00	-	0.41
	50	5	4029	600.00	-	0.36
		10	3789	600.00	-	0.39
		15	3484	108.59	-	0.37
6	10	5	3877	85.51	-	0.62
		10	3613	98.26	-	0.67
		15	3207	86.54	-	0.67
	20	5	3978	104.76	-	0.69
		10	3722	600.00	-	0.62
		15	3428	106.27	-	0.67
	50	5	4338	600.00	-	0.72
		10	4033	329.87	-	0.67
		15	3823	600.00	-	0.47
7	10	5	1511	35.84	-	0.83
		10	993	74.75	-	0.84
		15	588	59.98	-	0.84
	20	5	1663	56.86	-	0.83
		10	1193	81.15	-	0.86
		15	804	85.88	-	0.81

	50	5	2208	62.36	-	0.86
		10	1865	40.73	-	0.2
		15	1403	202.09	-	0.83
8	10	5	1915	600.00	-	1.97
		10	1395	600.00	-	1.97
		15	877	600.00	-	2.03
	20	5	2254	600.00	-	1.97
		10	1695	600.00	-	1.95
		15	1213	600.00	-	1.95
	50	5	3220	600.00	-	1.98
		10	2732	600.00	-	2.05
		15	2142	600.00	-	1.86
9	10	5	-255	20.37	-	1.33
		10	-860	11.87	-	1.42
		15	-1485	17.70	-	1.41
	20	5	105	20.19	-	1.31
		10	-500	20.98	-	1.44
		15	-1125	16.00	-	1.37
	50	5	1185	24.70	-	1.39
		10	570	23.06	-	1.39
		15	-45	26.59	-	1.42
10	10	5	11235	600.00	-	2.55
		10	10381	600.00	-	3.06
		15	9452	600.00	-	2.53
	20	5	11428	600.00	-	2.59
		10	10831	600.00	-	2.59
		15	9967	600.00	-	2.59
	50	5	12346	600.00	-	2.59
		10	11734	600.00	-	2.45
		15	11053	600.00	-	2.55
11	10	5	9557	600.00	-	3.14
		10	9390	600.00	-	3.14
		15	8131	600.00	-	3.08
	20	5	9478	600.00	-	3.25
		10	9105	600.00	-	3.03
		15	8345	600.00	-	3.12
	50	5	10409	600.00	-	3.25
		10	10135	600.00	-	3.12
		15	9583	600.00	-	3.25
12	10	5	13397	600.00	-	6.48
		10	11325	600.00	-	6.91
		15	9962	600.00	-	6.89
	20	5	13915	600.00	-	6.16
		10	12128	600.00	-	6.16
		15	14439	600.00	-	5.66
	50	5	16279	600.00	-	6.06
		10	14792	600.00	-	6.33
		15	14908	600.00	-	6.28
13	10	5	18930	600.00	-	38.95
		10	15264	600.00	-	37.18
		15	23235	600.00	-	37.31
	20	5	-	600.00	-	-
		10	14932	600.00	-	38.76
		15	14829	600.00	-	39.17
	50	5	18355	600.00	-	24.97

		10	-	600.00	-	-
		15	-	600.00	-	-
14	10	5	2848	600.00	-	2.72
		10	4044	600.00	-	2.55
		15	-221	600.00	-	2.53
	20	5	3600	600.00	-	2.5
		10	831	600.00	-	2.8
		15	-83	600.00	-	2.67
	50	5	2746	173.49	-	2.62
		10	1830	600.00	-	2.67
		15	892	600.00	-	2.67
15	10	5	5080	401.91	-	6.58
		10	3780	337.02	-	7.84
		15	2475	356.88	-	4.97
	20	5	5642	442.48	-	5.28
		10	4240	399.07	-	6.64
		15	2896	303.26	-	6.84
	50	5	6894	332.88	-	6.48
		10	5590	463.95	-	5.83
		15	4275	300.84	-	6.94
16	10	5	10120	600.00	-	1.61
		10	9699	600.00	-	1.59
		15	9013	600.00	-	1.62
	20	5	10424	600.00	-	1.2
		10	10107	600.00	-	1.55
		15	9014	600.00	-	1.52
	50	5	10227	600.00	-	1.37
		10	9905	600.00	-	1.67
		15	9409	600.00	-	1.61
17	10	5	14978	600.00	-	3.73
		10	14197	600.00	-	3.34
		15	12926	600.00	-	3.78
	20	5	15792	600.00	-	3.06
		10	14132	600.00	-	3.67
		15	12706	600.00	-	3.98
	50	5	16262	600.00	-	3.83
		10	14646	600.00	-	2.64
		15	13246	600.00	-	3.73
18	10	5	14512	600.00	-	2.78
		10	13910	600.00	-	2.52
		15	12606	600.00	-	2.61
	20	5	15420	600.00	-	2.75
		10	14626	600.00	-	2.62
		15	12306	600.00	-	2.7
	50	5	15177	600.00	-	2.75
		10	14305	600.00	-	2.73
		15	12807	600.00	-	2.42
19	10	5	25300	600.00	-	8.25
		10	25957	600.00	-	8.36
		15	23700	600.00	-	8.42
	20	5	26499	600.00	-	8.3
		10	21168	600.00	-	8.72
		15	24512	600.00	-	8.48
	50	5	27887	600.00	-	8.52
		10	26899	600.00	-	7.97

		15	22752	600.00	-	8.45
20	10	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	20	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	50	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
21	10	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	20	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	50	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
22	10	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	20	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	50	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
23	10	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	20	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	50	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
24	10	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	20	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	50	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-

Table B.8: Results of method 1 with all extensions

Instance	Weight of working extension	Weight of preference extension	Objective value first subproblem	Solving time first subproblem (s)	Objective value second subproblem	Solving time second subproblem (s)
1	10	5	1240	1.94	1321	0.05
		10	1184	1.39	1324	0.05

		15	1090	1.11	1321	0.09
	20	5	1250	1.39	1321	0.03
		10	1266	1.28	1420	0.05
		15	1100	1.50	1321	0.05
	50	5	1386	0.97	1424	0.05
		10	1296	1.22	1420	0.05
		15	1130	1.12	1321	0.03
2	10	5	1375	6.95	1443	0.13
		10	1140	3.72	1343	0.13
		15	1015	2.31	1343	0.14
	20	5	1355	6.91	1343	0.11
		10	1230	3.25	1341	0.12
		15	1105	5.45	1343	0.14
	50	5	1570	2.33	1440	0.14
		10	1430	2.19	1437	0.14
		15	1295	8.27	1439	0.12
3	10	5	2242	2.48	2331	0.44
		10	2106	2.78	2328	0.34
		15	2055	9.36	2421	0.44
	20	5	2303	4.30	2338	0.28
		10	2177	2.61	2332	0.48
		15	2042	3.28	2332	0.27
	50	5	2483	11.18	2337	0.48
		10	2358	-5.31	2338	0.5
		15	2318	9.12	2434	0.45
4	10	5	3332	40.06	3456	0.22
		10	3072	24.86	3364	0.22
		15	3130	23.45	3654	0.22
	20	5	3429	27.56	3458	0.22
		10	3354	27.01	3554	0.22
		15	3144	22.19	3556	0.22
	50	5	3919	22.47	3653	0.27
		10	3552	45.79	3455	0.22
		15	3474	20.97	3554	0.22
5	10	5	3409	180.64	3582	0.37
		10	3149	115.98	3679	0.41
		15	2904	141.42	3582	0.33
	20	5	3589	91.95	3592	0.23
		10	3329	113.24	3681	0.33
		15	3034	600.00	3676	0.33
	50	5	4164	126.91	3685	0.27
		10	3868	80.20	3675	0.17
		15	3660	99.26	3783	0.39
6	10	5	4503	36.34	4691	0.61
		10	4037	63.60	4592	0.64
		15	3907	71.60	4592	0.62
	20	5	4429	86.77	4492	0.61
		10	4196	81.15	4491	0.61
		15	4062	67.71	4598	0.61
	50	5	4802	92.96	4486	0.61
		10	4638	75.20	4585	0.64
		15	4351	53.25	4490	0.61
7	10	5	2483	78.76	2713	0.78
		10	2092	93.17	2698	0.78
		15	1708	80.47	2796	0.77

	20	5	2640	322.83	2699	0.61
		10	2283	131.94	2699	0.94
		15	1922	600.00	2697	0.97
	50	5	3362	59.84	2803	0.77
		10	2922	62.26	2705	0.83
		15	2624	127.05	2799	0.75
8	10	5	-	6.20	-	-
		10	-	6.23	-	-
		15	-	6.52	-	-
	20	5	-	6.94	-	-
		10	-	7.03	-	-
		15	-	6.06	-	-
	50	5	-	5.66	-	-
		10	-	5.97	-	-
		15	-	6.19	-	-
9	10	5	-225	391.61	2383	4.87
		10	-810	184.99	2587	3.61
		15	-1380	192.24	2182	5.7
	20	5	140	600.00	1781	3.87
		10	-450	130.46	2385	4.86
		15	-1035	111.57	2181	5.87
	50	5	1215	600.00	1777	4.86
		10	630	600.00	2281	5.12
		15	45	600.00	2181	5.7
10	10	5	11245	600.00	-	2.48
		10	10481	600.00	-	2.59
		15	9467	600.00	-	2.47
	20	5	11240	600.00	-	2.58
		10	10873	600.00	-	2.37
		15	8780	600.00	-	2.72
	50	5	13264	600.00	-	2.66
		10	11696	600.00	-	2.66
		15	10955	600.00	-	2.44
11	10	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	20	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	50	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
12	10	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	20	5	-	600.00	-	-
		10	-	600.00	-	-
		15	13980	600.00	15754	14.03
	50	5	17752	600.00	15867	11.75
		10	-	600.00	-	-
		15	-	600.00	-	-
13	10	5	18860	600.00	20045	438.14
		10	15444	600.00	18931	468.29
		15	13694	600.00	-	29.92
	20	5	-	600.00	-	-

		10	-	600.00	-	-
		15	15877	600.00	-	35.98
	50	5	18768	600.00	15265	147.99
		10	17185	600.00	15857	163.61
		15	14777	600.00	15644	159.35
14	10	5	4817	600.00	-	2.52
		10	3658	600.00	-	2.48
		15	2606	600.00	-	2.67
	20	5	4177	600.00	-	1.23
		10	-	600.00	-	-
		15	2558	600.00	-	0.78
	50	5	5481	600.00	-	2.36
		10	4839	600.00	-	2.53
		15	4237	600.00	-	2.55
15	10	5	5583	416.97	-	6.7
		10	3991	406.95	-	6.61
		15	2600	504.42	-	6.66
	20	5	5568	502.49	-	6.64
		10	4283	457.38	-	6.22
		15	3074	441.64	-	5.92
	50	5	7314	538.26	-	6.67
		10	5871	363.75	-	6.53
		15	4439	482.23	-	6.55
16	10	5	-	13.11	-	-
		10	-	13.05	-	-
		15	-	11.87	-	-
	20	5	-	15.37	-	-
		10	-	12.34	-	-
		15	-	13.70	-	-
	50	5	-	13.86	-	-
		10	-	12.17	-	-
		15	-	11.17	-	-
17	10	5	-	14.95	-	-
		10	-	11.84	-	-
		15	-	14.17	-	-
	20	5	-	16.41	-	-
		10	-	15.73	-	-
		15	-	13.83	-	-
	50	5	-	13.61	-	-
		10	-	14.78	-	-
		15	-	10.75	-	-
18	10	5	17503	570.01	18531	4.08
		10	-	600.00	-	-
		15	-	600.00	-	-
	20	5	17576	600.00	18404	3.17
		10	-	600.00	-	-
		15	-	600.00	-	-
	50	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
19	10	5	-	30.58	-	-
		10	-	55.65	-	-
		15	-	39.47	-	-
	20	5	-	64.84	-	-
		10	-	40.48	-	-

		15	-	34.00	-	-
	50	5	-	33.69	-	-
		10	-	34.70	-	-
		15	-	48.87	-	-
20	10	5	-	194.25	-	-
		10	-	170.47	-	-
		15	-	152.80	-	-
	20	5	-	211.08	-	-
		10	-	176.24	-	-
		15	-	173.82	-	-
	50	5	-	192.08	-	-
		10	-	138.71	-	-
		15	-	172.44	-	-
21	10	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	20	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	50	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
22	10	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	20	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	50	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
23	10	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	20	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	50	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
24	10	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	20	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-
	50	5	-	600.00	-	-
		10	-	600.00	-	-
		15	-	600.00	-	-

Table B.2: Results of method 1 with only the working hours extension.

Instance	Weight of working extension	Objective value first subproblem	Solving time first subproblem (s)	Objective value second subproblem	Solving time second subproblem (s)
1	10	1331	4.30	-	0.02
	20	1324	2.61	-	0.02
	50	1354	1.27	-	0.03
2	10	820	6.03	-	0.19
	20	960	9.67	-	0.11
	50	1400	5.78	-	0.22
3	10	2075	3.62	-	0.44
	20	2164	9.98	-	0.3
	50	2404	22.98	-	0.28
4	10	3505	28.95	-	0.25
	20	3507	23.25	-	0.27
	50	3906	19.05	-	0.25
5	10	3669	600.00	-	0.28
	20	3829	600.00	-	0.37
	50	4309	133.82	-	0.37
6	10	4162	124.51	-	0.7
	20	4212	87.27	-	0.75
	50	4613	600.00	-	0.7
7	10	1800	55.18	-	0.75
	20	2100	49.43	-	0.81
	50	2600	79.71	-	0.84
8	10	2420	600.00	-	1.91
	20	2642	600.00	-	1.95
	50	3702	600.00	-	1.34
9	10	360	14.33	-	1.33
	20	720	15.25	-	1.28
	50	1800	26.42	-	1.61
10	10	10435	475.92	-	2.86
	20	11730	600.00	-	2.53
	50	11936	580.38	-	1.47
11	10	11123	600.00	-	2.91
	20	9648	600.00	-	1.61
	50	11062	600.00	-	3.45
12	10	15034	600.00	-	6.3
	20	15838	600.00	-	6.69
	50	17938	600.00	-	6.44
13	10	15056	600.00	-	28.09
	20	15357	600.00	-	26.79
	50	20070	600.00	-	30.61
14	10	-	600.00	-	-
	20	5248	600.00	-	2.2
	50	3817	506.98	-	2.55
15	10	10161	600.00	-	6.67
	20	-	600.00	-	-
	50	12278	600.00	-	7.12
16	10	10636	600.00	-	1.62
	20	10940	600.00	-	1.64
	50	10953	600.00	-	1.62
17	10	17663	600.00	-	3.22
	20	17388	600.00	-	3.8
	50	16667	600.00	-	3.7
18	10	16067	600.00	-	2.81
	20	15791	600.00	-	2.14
	50	16548	600.00	-	2.42
19	10	27584	600.00	-	8.44
	20	26660	600.00	-	8.02
	50	29265	600.00	-	8.45
20	10	-	600.00	-	-
	20	-	600.00	-	-
	50	-	600.00	-	-
21	10	-	600.00	-	-
	20	-	600.00	-	-
	50	-	600.00	-	-
22	10	-	600.00	-	-
	20	-	600.00	-	-
	50	-	600.00	-	-
23	10	-	600.00	-	-
	20	-	600.00	-	-
	50	-	600.00	-	-
24	10	-	600.00	-	-
	20	-	600.00	-	-
	50	-	600.00	-	-

Table B.3: Results of method 1 with only the preference extension.

Instance	Weight of preference extension	Objective value first subproblem	Solving time first subproblem (s)	Objective value second subproblem	Solving time second subproblem (s)
1	5	1221	0.84	-	0.02
	10	1141	0.70	-	0.02
	15	1061	0.64	-	0.02
2	5	495	1.06	-	0.19
	10	300	1.31	-	0.19
	15	85	1.39	-	0.19
3	5	1845	3.44	-	0.34
	10	1660	3.22	-	0.36
	15	1490	1.81	-	0.39
4	5	3214	17.05	-	0.23
	10	3024	17.94	-	0.25
	15	2636	26.40	-	0.25
5	5	3234	600.00	-	0.41
	10	2959	600.00	-	0.39
	15	2639	600.00	-	0.39
6	5	3632	55.17	-	0.56
	10	3332	97.02	-	0.67
	15	3156	68.25	-	0.69
7	5	1202	30.73	-	0.84
	10	812	28.95	-	0.69
	15	388	35.40	-	0.77
8	5	1497	600.00	-	2.05
	10	1053	600.00	-	1.92
	15	470	600.00	-	1.97
9	5	-615	4.19	-	1.64
	10	-1230	4.28	-	1.61
	15	-1845	4.45	-	1.62
10	5	9376	600.00	-	2.67
	10	8547	600.00	-	2.53
	15	8075	600.00	-	2.48
11	5	9752	600.00	-	2.89
	10	8494	600.00	-	3.19
	15	5560	594.28	-	3.03
12	5	11994	600.00	-	6.77
	10	10760	600.00	-	6.73
	15	8882	600.00	-	6.77
13	5	-	600.00	-	-
	10	-	600.00	-	-
	15	6544	600.00	-	38.67
14	5	1236	415.47	-	2.77
	10	143	600.00	-	2.75
	15	-852	292.60	-	2.61
15	5	4630	301.10	-	5.66
	10	3409	228.82	-	6.47
	15	1995	290.65	-	7.66
16	5	7966	600.00	-	1.73
	10	9101	600.00	-	1.67
	15	8713	600.00	-	1.67
17	5	15564	600.00	-	3.39
	10	13371	600.00	-	3.77
	15	12957	600.00	-	2.84
18	5	14278	600.00	-	2.77
	10	13611	600.00	-	2.66
	15	12035	600.00	-	2.72
19	5	24495	600.00	-	8.37
	10	22453	600.00	-	8.61
	15	21016	600.00	-	8.36
20	5	-	600.00	-	-
	10	-	600.00	-	-
	15	-	600.00	-	-
21	5	-	600.00	-	-
	10	-	600.00	-	-
	15	-	600.00	-	-
22	5	-	600.00	-	-
	10	-	600.00	-	-
	15	-	600.00	-	-
23	5	-	600.00	-	-
	10	-	600.00	-	-
	15	-	600.00	-	-
24	5	-	600.00	-	-
	10	-	600.00	-	-
	15	-	600.00	-	-

Table B.4: Results of method 1 with the third extension.

Instance	Objective value first subproblem	Solving time first subproblem (s)	Objective value second subproblem	Solving time second subproblem (s)
1	1301	1.06	-	0.02
2	700	0.98	-	0.19
3	2000	3.70	-	0.36
4	3404	19.34	-	0.27
5	3509	474.43	-	0.38
6	4011	74.82	-	0.62
7	1600	26.58	-	0.83
8	2200	600.00	-	1.34
9	0	2.53	-	1.59
10	9937	432.81	-	2.75
11	10930	600.00	-	3.16
12	13441	600.00	-	6.81
13	14247	600.00	-	46.17
14	3634	600.00	-	2.75
15	-	600.00	-	-
16	10532	600.00	-	1.12
17	19239	600.00	-	2.72
18	17754	600.00	-	2.81
19	24569	600.00	-	8.81
20	-	600.00	-	-
21	-	600.00	-	-
22	-	600.00	-	-
23	-	600.00	-	-
24	-	600.00	-	-

Table B.6: Results of method 1 with the combination of the working hours and third extension.

Instance	Weight of working extension	Objective value first subproblem	Solving time first subproblem (s)	Objective value second subproblem	Solving time second subproblem (s)
1	10	1315	0.84	1325	0.05
	20	1325	1.17	1325	0.05
	50	1355	0.75	1325	0.05
2	10	1390	3.09	1353	0.14
	20	1560	7.12	1454	0.16
	50	1700	5.17	1453	0.12
3	10	2368	5.25	2347	0.39
	20	2428	5.66	2342	0.44
	50	2608	3.12	2345	0.33
4	10	3613	18.89	3563	0.22
	20	3713	29.89	3571	0.22
	50	3912	30.19	3470	0.19
5	10	3768	87.02	3696	0.33
	20	4029	99.34	3793	0.33
	50	4409	84.28	3699	0.34
6	10	4648	63.81	4602	0.55
	20	4696	80.92	4502	0.61
	50	5116	74.76	4497	0.62
7	10	2801	82.57	2829	0.92
	20	3000	63.62	2737	0.73
	50	3600	100.68	2728	0.92
8	10	-	5.91	-	-
	20	-	6.78	-	-
	50	-	5.86	-	-
9	10	360	27.11	2918	3.87
	20	720	34.56	2409	3.67
	50	1800	67.39	2209	4.66
10	10	11728	600.00	-	2.67
	20	10736	597.51	-	2.5
	50	12035	580.82	-	2.59
11	10	-	600.00	-	-
	20	-	600.00	-	-
	50	-	600.00	-	-
12	10	15258	600.00	14930	14.3
	20	14741	600.00	13800	12
	50	16844	600.00	14101	14.39
13	10	15153	600.00	-	30.15
	20	18452	600.00	-	28.95
	50	25022	600.00	-	36.93
14	10	3061	600.00	-	2.64
	20	3876	600.00	-	2.59
	50	6032	600.00	-	2.42
15	10	9571	600.00	-	7.06
	20	10620	600.00	-	6.06
	50	15716	600.00	-	7.7
16	10	-	12.31	-	-
	20	-	12.80	-	-
	50	-	12.47	-	-
17	10	-	11.87	-	-
	20	-	18.92	-	-
	50	-	13.42	-	-
18	10	-	600.00	-	-
	20	19040	456.93	18855	3.91
	50	-	600.00	-	-
19	10	-	33.14	-	-
	20	-	42.20	-	-
	50	-	48.93	-	-
20	10	-	263.42	-	-
	20	-	260.75	-	-
	50	-	379.41	-	-
21	10	-	600.00	-	-
	20	-	600.00	-	-
	50	-	600.00	-	-
22	10	-	600.00	-	-
	20	-	600.00	-	-
	50	-	600.00	-	-
23	10	-	600.00	-	-
	20	-	600.00	-	-
	50	-	600.00	-	-
24	10	-	600.00	-	-
	20	-	600.00	-	-
	50	-	600.00	-	-

Table B.7: Results of method 1 with the combination of the preference and third extension.

Instance	Weight of preference extension	Objective value first subproblem	Solving time first subproblem (s)	Objective value second subproblem	Solving time second subproblem (s)
1	5	1221	0.84	-	0.02
	10	1141	0.70	-	0.02
	15	1061	0.64	-	0.02
2	5	495	1.06	-	0.19
	10	300	1.31	-	0.19
	15	85	1.39	-	0.19
3	5	1845	3.44	-	0.34
	10	1660	3.22	-	0.36
	15	1490	1.81	-	0.39
4	5	3214	17.05	-	0.23
	10	3024	17.94	-	0.25
	15	2636	26.40	-	0.25
5	5	3234	600.00	-	0.41
	10	2959	600.00	-	0.39
	15	2639	600.00	-	0.39
6	5	3632	55.17	-	0.56
	10	3332	97.02	-	0.67
	15	3156	68.25	-	0.69
7	5	1202	30.73	-	0.84
	10	812	28.95	-	0.69
	15	388	35.40	-	0.77
8	5	1497	600.00	-	2.05
	10	1053	600.00	-	1.92
	15	470	600.00	-	1.97
9	5	-615	4.19	-	1.64
	10	-1230	4.28	-	1.61
	15	-1845	4.45	-	1.62
10	5	9376	600.00	-	2.67
	10	8547	600.00	-	2.53
	15	8075	600.00	-	2.48
11	5	9752	600.00	-	2.89
	10	8494	600.00	-	3.19
	15	5560	594.28	-	3.03
12	5	11994	600.00	-	6.77
	10	10760	600.00	-	6.73
	15	8882	600.00	-	6.77
13	5	-	600.00	-	-
	10	-	600.00	-	-
	15	6544	600.00	-	38.67
14	5	1236	415.47	-	2.77
	10	143	600.00	-	2.75
	15	-852	292.60	-	2.61
15	5	4630	301.10	-	5.66
	10	3409	228.82	-	6.47
	15	1995	290.65	-	7.66
16	5	7966	600.00	-	1.73
	10	9101	600.00	-	1.67
	15	8713	600.00	-	1.67
17	5	15564	600.00	-	3.39
	10	13371	600.00	-	3.77
	15	12957	600.00	-	2.84
18	5	14278	600.00	-	2.77
	10	13611	600.00	-	2.66
	15	12035	600.00	-	2.72
19	5	24495	600.00	-	8.37
	10	22453	600.00	-	8.61
	15	21016	600.00	-	8.36
20	5	-	600.00	-	-
	10	-	600.00	-	-
	15	-	600.00	-	-
21	5	-	600.00	-	-
	10	-	600.00	-	-
	15	-	600.00	-	-
22	5	-	600.00	-	-
	10	-	600.00	-	-
	15	-	600.00	-	-
23	5	-	600.00	-	-
	10	-	600.00	-	-
	15	-	600.00	-	-
24	5	-	600.00	-	-
	10	-	600.00	-	-
	15	-	600.00	-	-

C
Results of method 2

Table C.1: Results of method 2 with the exact fitness function, pool size = 10 and first stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1015	1222	1023	8.7	6.22	15.3
	0.5	1015	1317	915	9.48	6.62	19.65
	0.7	1122	1226	1216	8.42	6.45	6.94
2	0.3	1646	1730	1831	42.39	50.45	74.23
	0.5	1624	1628	1428	49.09	53.78	65.64
	0.7	1635	1644	1332	36.83	65.46	47.17
3	0.3	2122	3039	2241	77.09	600	600
	0.5	2223	3457	-	600	600	600
	0.7	2033	1847	2538	600	207.58	600
4	0.3	2352	2460	2666	105.68	117.05	102.38
	0.5	2747	2254	2762	105.1	182.77	202.5
	0.7	2849	2654	2555	95.87	213.89	238.73
5	0.3	2460	2777	2878	164.74	236.95	271.23
	0.5	2495	2677	2688	238.39	219.73	248.62
	0.7	2670	2276	2460	304.97	423.13	400.72
6	0.3	3474	4382	3886	381.03	258.33	230.7
	0.5	3877	4079	3865	418.36	600	600
	0.7	3871	3886	3972	352.27	465.98	600
7	0.3	3616	3313	3529	600	600	600
	0.5	3338	3729	3329	600	600	600
	0.7	3411	3609	3355	582.8	600	600
8	0.3	6273	5478	5698	600	600	600
	0.5	5774	6080	7685	600	600	600
	0.7	7588	-	5774	289.14	161.83	600
9	0.3	-	-	-	119.77	124.29	121.13
	0.5	-	-	-	121.69	125.29	123.04
	0.7	-	-	-	121.27	122.68	121.76
10	0.3	-	-	-	286.56	291.01	288.14
	0.5	-	-	-	292.38	295.73	288.09
	0.7	-	-	-	293.85	294.99	287.23
11	0.3	8719	8745	9132	600	600	600
	0.5	9024	9026	9136	600	600	600
	0.7	8869	8532	9520	600	600	600
12	0.3	16993	17203	17204	600	600	600
	0.5	16097	17506	17245	600	600	600
	0.7	18135	16921	18360	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	-	-	271.82	262.97	264.43
	0.5	-	-	-	265.92	265.5	263.48
	0.7	-	-	-	263.62	264.28	266.96
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	268.59	321.93	361.5
	0.5	-	-	-	274.64	321.84	360.27
	0.7	-	-	-	273.46	307.43	360.04
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
18	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.2: Results of method 2 with the exact fitness function, pool size = 10 and second stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1121	1112	1321	11.42	8.31	7.25
	0.5	1217	1221	1521	8.05	12.83	3.67
	0.7	1323	1216	1011	7.5	10.61	12.36
2	0.3	1930	1538	1837	26.37	27.56	32.69
	0.5	1644	1638	1746	18.72	26.76	33.45
	0.7	1844	1852	1750	29.31	31.15	15.3
3	0.3	2328	2230	2631	106.98	600	600
	0.5	2436	2636	2943	600	600	600
	0.7	1942	2739	-	600	600	600
4	0.3	2244	2561	2465	145.61	97.99	153.22
	0.5	2649	2441	2249	61.04	100.88	128.62
	0.7	2545	2442	2239	111.04	150.3	167.93
5	0.3	2878	2888	3073	99.74	137.77	130.57
	0.5	3096	2456	2492	109.27	300.95	401.17
	0.7	2690	2359	2275	192.88	336.8	281.04
6	0.3	4282	4175	4495	217.73	308.65	297.42
	0.5	4383	4077	3868	204.69	312.59	600
	0.7	3367	3776	3668	525.8	600	600
7	0.3	3605	3714	3726	394.17	352.77	600
	0.5	3510	3494	3327	546.52	600	600
	0.7	3210	3609	3292	477.07	498.28	600
8	0.3	9806	6999	5887	155.46	100.9	600
	0.5	9306	6394	10022	99.04	600	95.41
	0.7	8407	7373	8519	125.18	142.1	122.68
9	0.3	-	-	-	74.92	72.01	71.51
	0.5	-	-	-	71.54	74.64	73.65
	0.7	-	-	-	72.52	72.37	70.92
10	0.3	-	-	-	181.3	178.47	177.33
	0.5	-	-	-	175.5	178.74	180.61
	0.7	-	-	-	183.02	176.58	179.55
11	0.3	7991	8105	8981	600	600	600
	0.5	9044	9757	9325	600	600	600
	0.7	7823	8939	10152	600	600	600
12	0.3	17544	17124	17817	600	600	600
	0.5	17647	16129	18110	600	600	600
	0.7	17408	18312	17529	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	-	-	157.02	157.89	158.5
	0.5	-	-	-	160.49	157.47	157.66
	0.7	-	-	-	158.77	157.32	156.96
15	0.3	-	-	-	509.12	496.56	493.01
	0.5	-	-	-	505.26	484.45	499.71
	0.7	-	-	-	506.7	498.21	500.31
16	0.3	-	-	-	211.13	258.79	281.85
	0.5	-	-	-	193.16	239.5	273.62
	0.7	-	-	-	210.23	236.58	279.76
17	0.3	-	-	-	467.78	516.9	585.88
	0.5	-	-	-	486.47	504.76	576.52
	0.7	-	-	-	460.01	530.9	572.72
18	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.3: Results of method 2 with the exact fitness function, pool size = 10 and third stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1316	1117	1015	600	600	600
	0.5	1009	1015	1012	600	600	600
	0.7	1109	911	817	600	600	600
2	0.3	1630	1728	1439	600	600	600
	0.5	1530	1726	1626	600	600	600
	0.7	1439	1232	1538	600	600	600
3	0.3	-	2347	1831	600	600	600
	0.5	2435	2133	2531	600	600	600
	0.7	2536	2333	3551	600	600	600
4	0.3	2451	2134	2049	600	600	600
	0.5	2149	2248	2151	600	600	600
	0.7	2138	2442	2639	600	600	600
5	0.3	2581	2682	2467	600	600	600
	0.5	2665	2260	2483	600	600	600
	0.7	2066	2373	2767	600	600	600
6	0.3	4078	3875	3988	600	600	600
	0.5	3659	3867	4079	600	600	600
	0.7	3564	3772	3765	600	600	600
7	0.3	3514	3608	3626	600	600	600
	0.5	3585	3413	3714	600	600	600
	0.7	3811	3629	3630	600	600	600
8	0.3	5993	6190	7797	600	600	600
	0.5	6470	6986	7782	600	600	600
	0.7	7247	6081	7492	600	600	600
9	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
10	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
11	0.3	7540	8537	9224	600	600	600
	0.5	9214	8507	8718	600	600	600
	0.7	8328	9023	9555	600	600	600
12	0.3	16715	18222	17223	600	600	600
	0.5	16714	17538	18394	600	600	600
	0.7	17427	17129	15919	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
18	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.4: Results of method 2 with the exact fitness function, pool size = 20 and first stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1017	1022	1020	9.39	14.05	31.14
	0.5	1320	1018	1017	6.98	16.24	14.72
	0.7	1116	1013	919	12.89	17.03	41.45
2	0.3	1741	1634	1534	34.26	35.58	76.89
	0.5	1632	1739	1723	42.67	39.76	99.59
	0.7	1538	1633	1534	68.31	53.7	56.28
3	0.3	-	-	-	600	600	600
	0.5	3153	-	2931	600	600	600
	0.7	2938	2227	2640	600	600	600
4	0.3	2752	2955	2651	89.7	108.88	84.85
	0.5	2258	2444	2439	105.74	174.72	296.63
	0.7	2457	2346	2145	161.85	224.22	292.82
5	0.3	2983	2279	2698	100.6	272.12	175.61
	0.5	2573	2452	2584	237.22	482.64	369.32
	0.7	2679	2265	2475	196.25	524.02	600
6	0.3	4260	3988	3986	584.47	457.15	600
	0.5	4168	3880	3862	361.43	464.53	600
	0.7	4273	3770	3481	578.68	600	600
7	0.3	3106	4419	3629	600	258.43	600
	0.5	3802	3834	3728	600	600	600
	0.7	3314	2898	3813	600	600	600
8	0.3	7881	-	6284	253.56	177.67	600
	0.5	6068	5556	6586	600	600	600
	0.7	7486	7785	6985	600	420.76	600
9	0.3	-	-	-	129.91	129.57	131.93
	0.5	-	-	-	128.26	130.21	130.68
	0.7	-	-	-	131.13	131.38	135.69
10	0.3	-	-	-	306.46	305.27	308.01
	0.5	-	-	-	303.88	311.02	308.1
	0.7	-	-	-	305.84	308.7	309.42
11	0.3	8755	8349	9262	600	600	600
	0.5	7803	8918	9523	600	600	600
	0.7	7511	8848	9434	600	600	600
12	0.3	17834	17316	18005	600	600	600
	0.5	17740	17676	17413	600	600	600
	0.7	17551	16803	17706	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	-	-	279.92	281.6	280.79
	0.5	-	-	-	281.92	281.1	281.76
	0.7	-	-	-	280.11	281.62	280.71
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	293.65	352.82	387.47
	0.5	-	-	-	313.15	362.52	378.02
	0.7	-	-	-	306.6	350.12	395.3
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
18	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.5: Results of method 2 with the exact fitness function, pool size = 20 and second stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	921	1012	916	8.5	8.48	16.94
	0.5	1318	1116	1013	7.55	16.75	11.11
	0.7	1016	1111	1021	17.83	15.3	18.7
2	0.3	1540	1837	1524	38.15	22.6	45.76
	0.5	1544	1538	1536	29.56	32.98	43.67
	0.7	1538	1528	1426	48.79	40.95	57.12
3	0.3	2030	2842	3342	600	600	600
	0.5	2239	-	2636	600	600	600
	0.7	2134	-	2644	600	600	600
4	0.3	2549	2845	2350	119.98	140.21	301.06
	0.5	2437	2444	2252	200.28	114.02	199.71
	0.7	2447	2655	2358	148.77	148.11	291.67
5	0.3	2758	2862	2671	196.83	276.75	344.32
	0.5	2777	2782	2681	185.21	351.14	332.16
	0.7	1853	2268	2361	330.37	375.92	600
6	0.3	4080	3861	4087	572.18	352.91	468.65
	0.5	3982	3774	4173	246.24	600	600
	0.7	3575	3762	3588	414.41	574.58	600
7	0.3	3598	4039	3331	600	292.11	459.81
	0.5	3292	3818	3622	600	600	600
	0.7	3800	3511	3333	600	600	600
8	0.3	-	5974	6564	104.45	600	600
	0.5	5988	8790	7187	600	130.6	600
	0.7	6488	9109	-	600	109.52	103.91
9	0.3	-	-	-	79.92	81.85	81.98
	0.5	-	-	-	80.67	81.12	82.26
	0.7	-	-	-	79.67	83.59	81.15
10	0.3	-	-	-	197.61	200.38	201.53
	0.5	-	-	-	201.24	198.15	197.42
	0.7	-	-	-	200.81	196.78	197.49
11	0.3	8213	8692	8148	600	600	600
	0.5	8121	8950	8318	600	600	600
	0.7	9129	9412	9743	600	600	600
12	0.3	15907	18119	17939	600	600	600
	0.5	17214	18660	18217	600	600	600
	0.7	17383	17230	18242	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	-	-	177.97	180.52	175.83
	0.5	-	-	-	175.74	177.1	177.49
	0.7	-	-	-	176.46	174.77	176.99
15	0.3	-	-	-	550.52	550.39	547.65
	0.5	-	-	-	540.09	552.26	546.91
	0.7	-	-	-	543.62	542.6	557.49
16	0.3	-	-	-	240.25	271.01	309.37
	0.5	-	-	-	225.3	282.6	297.26
	0.7	-	-	-	246.4	259.61	292.95
17	0.3	-	-	-	540.19	600	600
	0.5	-	-	-	528.76	600	600
	0.7	-	-	-	532.79	600	600
18	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.6: Results of method 2 with the exact fitness function, pool size = 20 and third stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1217	1114	812	600	600	600
	0.5	1115	1017	1009	600	600	600
	0.7	919	911	911	600	600	600
2	0.3	1438	1529	1536	600	600	600
	0.5	1538	1541	1537	600	600	600
	0.7	1729	1740	1545	600	600	600
3	0.3	2116	2836	-	600	600	600
	0.5	2132	3039	2539	600	600	600
	0.7	2131	2227	3155	600	600	600
4	0.3	2657	2354	2349	600	600	600
	0.5	2240	2440	2342	600	600	600
	0.7	2152	2237	2151	600	600	600
5	0.3	2478	1970	2259	600	600	600
	0.5	1961	2661	2375	600	600	600
	0.7	2864	2767	2076	600	600	600
6	0.3	4061	3678	3567	600	600	600
	0.5	3565	3471	4073	600	600	600
	0.7	3654	2969	3664	600	600	600
7	0.3	3516	4215	3816	600	600	600
	0.5	2918	3729	3409	600	600	600
	0.7	3317	3027	3409	600	600	600
8	0.3	5766	5381	5759	600	600	600
	0.5	6367	6366	6471	600	600	600
	0.7	5882	6156	6662	600	600	600
9	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
10	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
11	0.3	8209	8710	9158	600	600	600
	0.5	8234	9538	9431	600	600	600
	0.7	8619	9314	9833	600	600	600
12	0.3	16540	16005	17798	600	600	600
	0.5	18045	18132	17811	600	600	600
	0.7	17010	17717	16831	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
18	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.7: Results of method 2 with the relaxation fitness function, pool size = 10 and first stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	-	1016	1120	5.77357	5.669	6.42033
	0.5	-	-	1016	3.26539	10.0747	8.9837
	0.7	-	-	915	6.29641	8.23858	12.0595
2	0.3	1931	-	-	8.34314	15.1864	28.5917
	0.5	-	-	-	24.342	32.5132	16.4207
	0.7	-	-	-	23.6858	23.342	36.3255
3	0.3	2327	2238	2435	600	600	600
	0.5	3441	3234	2534	600	600	600
	0.7	2130	2135	-	600	600	600
4	0.3	-	2862	2751	46.9497	71.0822	72.9322
	0.5	2551	2557	2254	58.6676	50.2462	89.3372
	0.7	2648	-	-	78.7528	104.086	147.583
5	0.3	3177	-	2991	36.2005	150.723	112.351
	0.5	3167	-	3080	56.8865	168.535	106.367
	0.7	2556	-	2869	81.3222	122.741	113.351
6	0.3	-	4399	4479	134.334	120.585	211.922
	0.5	-	4284	-	194.861	201.313	358.167
	0.7	-	-	-	215.031	265.637	391.105
7	0.3	-	-	-	253.388	258.278	218.578
	0.5	-	-	-	257.887	225.983	347.022
	0.7	-	-	-	385.425	473.278	297.26
8	0.3	-	-	9114	70.2292	311.946	79.6661
	0.5	-	-	-	600	600	80.4629
	0.7	-	8681	-	600	123.866	600
9	0.3	-	-	-	45.3404	46.1216	48.2621
	0.5	-	-	-	45.5435	46.3247	49.2464
	0.7	-	-	-	46.2935	46.4341	45.7154
10	0.3	-	-	-	107.555	103.977	102.539
	0.5	-	-	-	103.196	105.664	104.914
	0.7	-	-	-	106.648	107.914	106.227
11	0.3	7814	7003	7110	600	600	600
	0.5	7717	7003	7112	600	600	600
	0.7	7298	7823	7732	600	600	600
12	0.3	14598	13536	14679	600	600	600
	0.5	14185	15699	15721	600	600	600
	0.7	13782	15384	15809	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	-	-	86.6812	89.5247	89.556
	0.5	-	-	-	90.1653	88.431	90.4309
	0.7	-	-	-	89.3685	89.8216	90.634
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	153.957	200.157	228.593
	0.5	-	-	-	150.067	193.298	231.358
	0.7	-	-	-	161.738	176.018	227.296
17	0.3	-	-	-	407.814	478.752	533.914
	0.5	-	-	-	383.425	485.793	549.08
	0.7	-	-	-	390.721	464.06	533.82
18	0.3	-	-	-	448.764	600	600
	0.5	-	-	-	462.591	597.363	600
	0.7	-	-	-	458.029	575.474	600
19	0.3	-	-	-	444.686	456.795	459.763
	0.5	-	-	-	448.998	448.702	459.654
	0.7	-	-	-	448.264	463.091	447.342
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.8: Results of method 2 with the relaxation fitness function, pool size = 10 and second stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1315	-	1224	4.53092	8.29627	6.06206
	0.5	-	-	1117	6.5464	7.34321	8.78061
	0.7	-	-	1121	5.26524	5.7652	9.90553
2	0.3	-	-	1929	9.4993	15.3114	10.4524
	0.5	-	-	-	13.3896	9.96802	25.9356
	0.7	-	-	2141	15.6395	23.1389	5.53085
3	0.3	3050	2332	-	600	600	600
	0.5	-	2841	2548	47.0903	600	600
	0.7	2753	2538	2943	600	600	600
4	0.3	2652	2966	-	28.9198	24.1545	56.9802
	0.5	2753	-	-	37.0442	68.8543	90.1497
	0.7	-	2656	-	46.4654	69.6824	177.675
5	0.3	3170	3170	2572	41.372	49.2777	87.5249
	0.5	1982	-	2368	124.538	132.631	144.208
	0.7	2150	-	-	138.74	118.616	184.362
6	0.3	-	-	-	132.35	120.421	167.347
	0.5	4173	-	-	190.861	214.098	214.062
	0.7	3892	4093	-	161.16	190.38	312.368
7	0.3	-	-	-	121.96	99.1021	227.811
	0.5	-	-	3219	157.02	159.738	314.836
	0.7	-	-	-	164.535	214.359	385.878
8	0.3	-	-	9495	46.4341	452.358	59.5581
	0.5	-	8490	8064	468.106	55.9803	51.1681
	0.7	-	-	7888	45.9341	46.731	107.726
9	0.3	-	-	-	32.0133	31.2477	31.3415
	0.5	-	-	-	31.1852	31.5289	31.2321
	0.7	-	-	-	31.2477	32.3883	31.154
10	0.3	-	-	-	70.8386	70.0261	74.6977
	0.5	-	-	-	71.7135	71.276	73.2759
	0.7	-	-	-	70.7761	71.6354	70.9167
11	0.3	8212	7721	7636	600	600	600
	0.5	7611	6412	7824	600	600	600
	0.7	7930	7213	8016	600	600	600
12	0.3	14103	14502	15104	600	600	600
	0.5	14586	-	15897	600	600	600
	0.7	13563	15598	15334	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	-	-	58.9332	57.6208	59.8238
	0.5	-	-	-	58.652	58.5113	59.8238
	0.7	-	-	-	56.9177	58.6207	61.7611
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	571.021	600	600
16	0.3	-	-	-	121.226	156.801	184.346
	0.5	-	-	-	114.492	145.989	176.315
	0.7	-	-	-	119.818	148.067	193.189
17	0.3	-	-	-	277.323	311.493	369.832
	0.5	-	-	-	283.854	301.65	356.005
	0.7	-	-	-	270.277	316.571	372.941
18	0.3	-	-	-	323.773	490.308	600
	0.5	-	-	-	356.521	457.342	600
	0.7	-	-	-	385.159	521.103	600
19	0.3	-	-	-	321.258	320.961	330.304
	0.5	-	-	-	325.664	328.538	325.961
	0.7	-	-	-	323.57	321.945	326.679
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.9: Results of method 2 with the relaxation fitness function, pool size = 10 and third stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	-	910	-	600	600	600
	0.5	-	1117	-	600	600	600
	0.7	-	-	-	600	600	600
2	0.3	1432	-	-	600	600	600
	0.5	-	-	1540	600	600	600
	0.7	-	-	-	600	600	600
3	0.3	2526	2339	3235	600	600	600
	0.5	2835	-	2831	600	600	600
	0.7	2428	2137	2348	600	600	600
4	0.3	2451	-	-	600	600	600
	0.5	-	2145	2151	600	600	600
	0.7	2150	2248	2254	600	600	600
5	0.3	-	-	2253	600	600	600
	0.5	2360	-	2266	600	600	600
	0.7	2262	2571	-	600	600	600
6	0.3	3775	-	-	600	600	600
	0.5	-	-	3069	600	600	600
	0.7	3061	-	3785	600	600	600
7	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
8	0.3	5774	-	-	600	600	600
	0.5	4916	-	-	600	600	600
	0.7	-	-	-	600	600	600
9	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
10	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
11	0.3	7498	7411	7424	600	600	600
	0.5	7520	7183	7918	600	600	600
	0.7	6907	7285	7901	600	600	600
12	0.3	14543	15104	13997	600	600	600
	0.5	14598	14681	15003	600	600	600
	0.7	15191	14789	15802	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
18	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.10: Results of method 2 with the relaxation fitness function, pool size = 20 and first stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1213	-	-	5.10472	7.71818	9.14157
	0.5	917	-	-	10.037	11.8429	10.8742
	0.7	-	-	1116	8.32752	11.9679	14.8751
2	0.3	-	-	-	17.5143	21.5765	34.4662
	0.5	-	-	-	27.5449	26.4981	60.2925
	0.7	-	-	-	31.2008	27.5292	32.6382
3	0.3	-	-	3241	46.8872	600	600
	0.5	1828	-	1627	600	600	600
	0.7	2333	1320	3144	600	600	600
4	0.3	-	2439	2263	65.464	115.554	169.255
	0.5	-	-	-	94.6493	125.35	160.253
	0.7	2247	2349	2453	106.805	107.226	218.271
5	0.3	-	-	2959	68.3388	70.6198	198.126
	0.5	2476	-	-	85.6187	188.611	291.276
	0.7	-	2464	-	150.536	160.52	228.936
6	0.3	4296	-	-	148.067	200.048	352.505
	0.5	-	-	-	177.721	267.48	555.647
	0.7	-	-	3229	185.908	175.097	600
7	0.3	-	3711	-	262.731	141.349	279.839
	0.5	-	-	-	167.269	325.586	285.729
	0.7	-	-	-	280.386	441.062	600
8	0.3	-	-	-	600	600	600
	0.5	-	-	-	88.0404	600	600
	0.7	9172	-	-	89.2122	600	600
9	0.3	-	-	-	57.0896	57.3708	57.5739
	0.5	-	-	-	58.2145	57.6364	59.1363
	0.7	-	-	-	57.9801	57.4021	57.8708
10	0.3	-	-	-	128.834	132.365	128.069
	0.5	-	-	-	130.787	129.334	127.709
	0.7	-	-	-	125.866	130.975	129.584
11	0.3	8610	7216	7194	600	600	600
	0.5	7336	7639	8045	600	600	600
	0.7	7822	7313	6917	600	600	600
12	0.3	-	14468	15803	600	600	600
	0.5	14605	14966	14815	600	600	600
	0.7	14372	15413	16103	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	-	-	107.414	107.727	111.586
	0.5	-	-	-	109.992	115.159	110.945
	0.7	-	-	-	110.992	111.382	110.367
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	181.127	211.547	279.245
	0.5	-	-	-	166.113	228.14	259.95
	0.7	-	-	-	166.316	229.186	250.325
17	0.3	-	-	-	481.902	530.274	600
	0.5	-	-	-	464.857	528.805	600
	0.7	-	-	-	464.919	542.788	600
18	0.3	-	-	-	515.462	600	600
	0.5	-	-	-	481.449	600	600
	0.7	-	-	-	496.682	600	600
19	0.3	-	-	-	544.82	539.617	550.96
	0.5	-	-	-	537.695	550.632	537.398
	0.7	-	-	-	544.46	539.148	531.211
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.11: Results of method 2 with the relaxation fitness function, pool size = 20 and second stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1016	1219	1023	4.49967	5.12463	9.34307
	0.5	-	1016	821	10.2336	10.5305	16.8581
	0.7	-	-	-	8.40564	10.9367	13.0459
2	0.3	1738	-	2031	11.6866	35.8179	10.2661
	0.5	-	-	1738	31.0322	18.4518	21.1546
	0.7	-	-	1649	19.5492	30.1088	24.2951
3	0.3	-	-	-	31.0134	600	600
	0.5	-	-	3442	600	41.3251	600
	0.7	-	2429	3444	56.7459	600	600
4	0.3	2357	-	2649	63.4016	53.418	79.5879
	0.5	2550	2449	-	72.401	64.4015	138.49
	0.7	-	2750	2249	86.8374	65.1984	61.8236
5	0.3	-	2575	-	81.8534	141.725	153.161
	0.5	2773	-	2560	79.1505	159.894	163.582
	0.7	-	-	-	76.4632	251.95	298.059
6	0.3	3971	3679	-	110.961	180.784	392.315
	0.5	-	-	-	200.97	241.685	185.361
	0.7	-	-	3464	196.892	232.249	556.881
7	0.3	-	-	-	141.818	223.421	274.714
	0.5	-	-	-	183.987	406.595	368.114
	0.7	-	-	-	359.005	496.573	466.45
8	0.3	8714	-	-	94.54	596.583	600
	0.5	-	-	-	600	524.318	600
	0.7	-	7875	-	64.9171	66.417	600
9	0.3	-	-	-	42.2625	42.8875	47.2309
	0.5	-	-	-	42.6531	42.7469	43.6374
	0.7	-	-	-	42.3719	43.1843	43.1062
10	0.3	-	-	-	96.7898	94.6962	97.1023
	0.5	-	-	-	95.8836	96.7898	95.993
	0.7	-	-	-	98.3834	97.8678	96.7273
11	0.3	8012	6917	7193	600	600	600
	0.5	7814	6900	8007	600	600	600
	0.7	7702	-	7820	600	600	600
12	0.3	-	14300	14688	600	600	600
	0.5	14482	15410	15409	600	600	600
	0.7	15076	15515	15388	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	-	-	81.6347	80.4316	78.9474
	0.5	-	-	-	80.2285	79.8535	79.5098
	0.7	-	-	-	78.8067	80.9785	80.791
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	151.63	182.377	220.187
	0.5	-	-	-	139.896	185.139	208.797
	0.7	-	-	-	138.099	190.283	208.813
17	0.3	-	-	-	339.022	391.44	425.922
	0.5	-	-	-	324.32	394.237	441.702
	0.7	-	-	-	317.508	378.55	432.281
18	0.3	-	-	-	430.14	559.272	600
	0.5	-	-	-	450.108	569.427	600
	0.7	-	-	-	432.047	534.023	600
19	0.3	-	-	-	393.643	405.33	405.158
	0.5	-	-	-	399.111	408.189	409.47
	0.7	-	-	-	399.486	403.752	410.376
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.12: Results of method 2 with the relaxation fitness function, pool size = 20 and third stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	-	912	910	600	600	600
	0.5	-	915	-	600	600	600
	0.7	-	-	-	600	600	600
2	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
3	0.3	1527	2127	-	600	600	600
	0.5	-	2345	2032	600	600	600
	0.7	1925	1628	2533	600	600	600
4	0.3	2243	2247	-	600	600	600
	0.5	2439	2139	2437	600	600	600
	0.7	2238	2040	2140	600	600	600
5	0.3	2670	2469	-	600	600	600
	0.5	-	-	2373	600	600	600
	0.7	-	-	-	600	600	600
6	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
7	0.3	-	-	2504	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
8	0.3	4444	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	5267	600	600	600
9	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
10	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
11	0.3	8114	7211	7408	600	600	600
	0.5	7888	7110	7400	600	600	600
	0.7	7408	7731	7615	600	600	600
12	0.3	-	13472	15194	600	600	600
	0.5	13651	14590	15706	600	600	600
	0.7	15196	14813	15687	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
18	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.13: Results of method 2 with the heuristic fitness function, pool size = 10 and first stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	912	1115	1015	25.37	18.37	16.81
	0.5	1213	920	1116	29.33	26.83	25.74
	0.7	912	1112	1020	102.2	21.37	42.37
2	0.3	2136	2348	2346	13.39	8	10.25
	0.5	1749	2136	1730	57.44	30.84	30.31
	0.7	1745	2436	2539	69.15	22.63	27.42
3	0.3	4686	-	4174	600	600	600
	0.5	4585	3565	3775	600	600	600
	0.7	4781	4292	4483	600	600	600
4	0.3	3262	3267	3266	267.39	75.32	46.43
	0.5	3354	3375	3363	88.08	226.84	181.48
	0.7	2955	2966	3764	274.64	159.98	91.37
5	0.3	2988	3696	3689	280.35	228.78	179.35
	0.5	3282	3485	4081	309.31	309.83	94.58
	0.7	3485	2993	3885	384.01	446.22	97.66
6	0.3	5202	5096	5109	238.92	123.65	125.71
	0.5	5608	5283	5486	117.64	203.92	101.61
	0.7	6186	5698	5594	100.57	97.05	121.42
7	0.3	4136	4324	4140	271.9	203.72	150.58
	0.5	5228	4630	4116	312.99	293.41	195.18
	0.7	4919	4751	4835	297.04	191.38	116
8	0.3	-	7798	7307	600	600	600
	0.5	6886	7565	7890	600	600	600
	0.7	7277	7780	7590	600	600	600
9	0.3	5414	5616	4300	258	261.24	413.36
	0.5	4986	5506	5396	507.92	600	436.29
	0.7	5505	4385	5398	508.49	511.71	255.12
10	0.3	13870	12720	12645	600	600	600
	0.5	13563	12331	13070	600	600	600
	0.7	13868	11545	12758	600	600	600
11	0.3	8962	11343	11252	600	600	538.25
	0.5	9619	10836	11569	600	572.77	600
	0.7	10035	30791	9152	600	600	600
12	0.3	16409	17790	17739	600	600	600
	0.5	17214	17606	38406	600	600	600
	0.7	17616	18240	18123	600	600	600
13	0.3	31014	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	9068	7535	8727	600	600	600
	0.5	8046	8683	8643	600	600	600
	0.7	9654	8868	9048	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
18	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.14: Results of method 2 with the heuristic fitness function, pool size = 10 and second stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1113	1115	1217	23.98	13.13	10.5
	0.5	1013	1010	1114	22.28	18.52	17.62
	0.7	1324	909	815	23.95	51.48	37.99
2	0.3	1835	2449	2146	15.25	5.9	11.08
	0.5	1548	2242	2249	37.62	12.81	12.29
	0.7	2240	2153	2246	35.78	12.16	11.1
3	0.3	3675	3770	4391	600	600	600
	0.5	2631	3458	2234	49.74	600	21.84
	0.7	4169	2638	3874	600	25.35	600
4	0.3	2858	3164	3071	95.31	86.99	72.2
	0.5	3263	3066	3370	219.73	178.06	93.25
	0.7	3158	3470	3258	202.75	182.83	87.45
5	0.3	3479	3588	3685	162.58	103.42	150.3
	0.5	3586	3092	3590	101.11	239.33	77.41
	0.7	3789	3594	3796	192.83	270.34	142
6	0.3	5284	5604	5593	111.46	111.47	64.74
	0.5	6204	5091	5593	63.14	175.77	91.21
	0.7	5583	4996	5091	237.76	181.08	121.85
7	0.3	4425	4234	4621	158.49	150.92	71.82
	0.5	4622	4633	5425	212.8	101.08	49.18
	0.7	4300	4425	5240	207.17	229.81	130.14
8	0.3	7886	8190	8486	600	227.97	406.26
	0.5	7883	8499	8181	378.05	313.15	600
	0.7	7700	8199	7171	600	218.79	600
9	0.3	6028	5404	5702	370.9	359.41	336.09
	0.5	6386	5920	4816	218.24	222.43	164.34
	0.7	6076	6187	4690	364.07	433.87	243.07
10	0.3	11844	13754	12145	600	600	405.7
	0.5	13778	13085	11457	600	600	600
	0.7	12473	13372	12324	600	600	600
11	0.3	11339	10262	9972	267.5	475.28	402.91
	0.5	9546	11257	10936	600	258.85	357.55
	0.7	10644	10850	10641	472.42	600	600
12	0.3	17406	17208	18311	600	600	411.64
	0.5	16493	17796	17940	600	600	600
	0.7	17817	18207	18013	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	9638	9256	-	600	600	600
	0.5	9176	8762	8940	600	600	600
	0.7	8648	10059	8743	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
18	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.15: Results of method 2 with the heuristic fitness function, pool size = 10 and third stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1119	1113	915	600	600	600
	0.5	1113	1017	1014	600	600	600
	0.7	913	1008	1119	600	600	600
2	0.3	1839	1752	1854	600	600	600
	0.5	2033	1844	1838	600	600	600
	0.7	1841	1641	1535	600	600	600
3	0.3	3170	4155	3673	600	600	600
	0.5	3668	4378	3881	600	600	600
	0.7	3767	4486	4054	600	600	600
4	0.3	3365	3054	3076	600	600	600
	0.5	3164	3377	3257	600	600	600
	0.7	3471	2968	4212	600	600	600
5	0.3	3689	3185	2883	600	600	600
	0.5	3596	3398	3190	600	600	600
	0.7	3592	3089	3380	600	600	600
6	0.3	4789	5190	5184	600	600	600
	0.5	5397	4378	4973	600	600	600
	0.7	5085	5090	5601	600	600	600
7	0.3	4518	4342	4040	600	600	600
	0.5	4436	4238	3824	600	600	600
	0.7	4626	3936	4122	600	600	600
8	0.3	8001	8396	7797	600	600	600
	0.5	7984	7079	7778	600	600	600
	0.7	7803	7873	8500	600	600	600
9	0.3	5916	4815	5722	600	600	600
	0.5	6600	4797	5701	600	600	600
	0.7	6526	4993	5699	600	600	600
10	0.3	13354	13057	11636	600	600	600
	0.5	13461	11954	12949	600	600	600
	0.7	11355	12154	13069	600	600	600
11	0.3	11237	10557	10560	600	600	600
	0.5	11235	9832	10820	600	600	600
	0.7	10254	10557	10529	600	600	600
12	0.3	17840	38826	38726	600	600	600
	0.5	17316	16299	38424	600	600	600
	0.7	17153	18322	18235	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	9664	8763	7909	600	600	600
	0.5	9076	8662	9464	600	600	600
	0.7	19614	-	8848	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
18	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.16: Results of method 2 with the heuristic fitness function, pool size = 20 and first stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1211	1114	814	14.13	26.68	18.02
	0.5	1019	1012	921	49.33	40.7	21.53
	0.7	1215	1116	1015	89.29	29.36	47.61
2	0.3	1838	1832	2145	20.81	44.53	20.1
	0.5	1738	1756	2144	48.65	64.41	21.44
	0.7	1854	2048	1849	51.47	85.3	29.34
3	0.3	2630	3760	3666	38.17	600	600
	0.5	4179	4383	4288	600	600	600
	0.7	3373	3568	-	600	600	600
4	0.3	3463	2661	2870	416.45	273.36	329.28
	0.5	3076	3479	3360	315.08	258.57	202.29
	0.7	3264	3060	3163	600	199.98	281.98
5	0.3	3190	3392	3186	386.15	370.43	374.97
	0.5	2891	3059	3686	600	384.71	205.4
	0.7	3667	3182	3373	600	495.47	389.85
6	0.3	5205	5288	5406	285.31	336.63	169.62
	0.5	4883	5192	5001	305.37	387.67	306.59
	0.7	5395	5385	5802	432.88	410.72	199.6
7	0.3	4633	4436	3819	342.76	355.48	410.24
	0.5	4117	3947	3938	360.49	600	600
	0.7	4556	3943	4432	475.13	600	372.25
8	0.3	6761	7583	7480	600	600	600
	0.5	7699	7600	7995	600	600	600
	0.7	8291	7185	8171	600	600	600
9	0.3	4589	5704	5505	600	600	580.37
	0.5	6313	3989	5116	600	600	457.43
	0.7	5395	5115	6110	600	600	600
10	0.3	12349	11941	13292	600	600	600
	0.5	12971	11646	13062	600	600	600
	0.7	12852	13238	11531	600	600	600
11	0.3	30074	10850	29977	600	600	600
	0.5	11265	30056	10756	600	600	600
	0.7	10524	10340	10059	600	600	600
12	0.3	17727	18436	16904	600	600	600
	0.5	17125	16416	18947	600	600	600
	0.7	17240	17918	17246	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	8037	8956	6894	600	600	600
	0.5	10896	8169	9237	600	600	600
	0.7	7428	8547	9130	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
18	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.17: Results of method 2 with the heuristic fitness function, pool size = 20 and second stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1119	1216	1014	16.28	17.86	14.68
	0.5	1018	1016	916	16.16	33.25	52.45
	0.7	1118	1221	910	47.9	50.89	45.56
2	0.3	1844	2262	2248	30.62	4.53	10.05
	0.5	2037	2332	2054	17.82	34.49	26.38
	0.7	2042	2034	2141	42.62	26.81	33.12
3	0.3	3465	3894	4077	600	600	600
	0.5	3464	3561	3564	600	600	600
	0.7	3558	4082	3680	600	600	600
4	0.3	3071	3448	3168	77.23	139.23	52.06
	0.5	3070	3568	3270	407.13	107.9	159.88
	0.7	3262	3165	3168	217.33	398.45	114.87
5	0.3	3579	3579	3603	283.55	114.18	201.01
	0.5	3304	3795	3678	301.71	409.85	101.25
	0.7	3377	3889	3790	508.41	214.33	181.92
6	0.3	4890	4987	5294	278.69	235.09	113.41
	0.5	5199	5393	5502	392.33	330.59	104.24
	0.7	5190	5593	5784	384.3	376.77	112.75
7	0.3	4411	4429	4438	288.38	191.18	189.2
	0.5	4721	4334	3929	376.72	386.79	206.36
	0.7	3923	4547	5234	576.34	308.76	163.93
8	0.3	8277	7599	-	600	600	404.55
	0.5	7867	-	8408	600	402.81	600
	0.7	8780	-	7884	225.83	400.38	600
9	0.3	4202	4977	6010	332.25	463.87	360.74
	0.5	5370	5522	5726	600	391.22	252.31
	0.7	4898	5324	5405	519.67	576.08	462.32
10	0.3	11331	13343	12458	600	600	600
	0.5	14097	9640	11333	600	600	600
	0.7	11859	13258	9734	600	600	600
11	0.3	11162	10144	9930	600	358.12	483.19
	0.5	10336	9923	30082	600	600	600
	0.7	10347	9844	10471	600	600	600
12	0.3	17017	17129	17048	600	600	600
	0.5	17100	17732	38306	600	600	600
	0.7	17514	17815	17511	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	31092	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	8225	9651	9016	600	600	600
	0.5	10791	8758	8866	600	600	600
	0.7	9433	9083	8164	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
18	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table C.18: Results of method 2 with the heuristic fitness function, pool size = 20 and third stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1012	1015	1017	600	600	600
	0.5	1227	1118	918	600	600	600
	0.7	1015	1108	717	600	600	600
2	0.3	1542	1744	1746	600	600	600
	0.5	1952	1750	2056	600	600	600
	0.7	1540	1527	1739	600	600	600
3	0.3	3481	3366	3975	600	600	600
	0.5	3965	3269	3986	600	600	600
	0.7	4267	4090	4180	600	600	600
4	0.3	3174	2882	3266	600	600	600
	0.5	3262	3057	3815	600	600	600
	0.7	3061	2562	2869	600	600	600
5	0.3	3206	3294	4344	600	600	600
	0.5	3286	3384	3186	600	600	600
	0.7	3393	3258	2978	600	600	600
6	0.3	4992	5488	4680	600	600	600
	0.5	5190	5708	5400	600	600	600
	0.7	5291	5082	5187	600	600	600
7	0.3	4238	4339	6063	600	600	600
	0.5	4317	4219	6644	600	600	600
	0.7	4133	4614	4030	600	600	600
8	0.3	7795	8063	7283	600	600	600
	0.5	6904	8094	7700	600	600	600
	0.7	7194	8579	7492	600	600	600
9	0.3	6215	6590	4811	600	600	600
	0.5	4628	5619	4610	600	600	600
	0.7	5196	6203	5696	600	600	600
10	0.3	10842	12309	13644	600	600	600
	0.5	11414	12149	11828	600	600	600
	0.7	12361	11482	12971	600	600	600
11	0.3	8934	10245	11153	600	600	600
	0.5	10845	11139	10855	600	600	600
	0.7	9949	10156	29091	600	600	600
12	0.3	17403	17448	17404	600	600	600
	0.5	17811	15809	17192	600	600	600
	0.7	18043	18247	38835	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	10261	-	600	600	600
	0.5	9646	8160	8823	600	600	600
	0.7	8242	8361	11393	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
18	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

D
Results of method 3

Table D.1: Results of method 3 with the exact fitness function, pool size = 10 and first stopping criterion.

		Best objective value			time initialize		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	909	1011	818	1.39	0.28	0.3
	0.5	809	909	813	0.27	0.3	0.33
	0.7	914	1109	910	0.3	0.28	0.37
2	0.3	1424	1325	1625	0.77	0.75	0.8
	0.5	1320	1438	1541	0.8	0.77	0.78
	0.7	1221	1532	1641	0.78	0.8	0.81
3	0.3	2829	2444	3039	2.59	2.39	2.16
	0.5	2633	2524	3155	2.66	2.36	2.53
	0.7	-	2428	2935	600	2.48	2.52
4	0.3	2043	2141	2946	1.52	1.53	1.53
	0.5	2351	2242	2953	1.64	2.17	1.59
	0.7	2450	2543	2466	3.07	1.66	1.53
5	0.3	2380	3075	3997	2.55	2.59	2.94
	0.5	3184	4082	3386	2.55	2.61	2.7
	0.7	3092	3470	3674	2.55	2.56	2.52
6	0.3	5403	5084	5388	4.41	5.52	5.03
	0.5	5486	5294	5197	4.73	4.98	4.84
	0.7	4787	5085	5595	4.95	4.81	4.39
7	0.3	5133	5238	5151	6.36	6.78	6.2
	0.5	4908	5439	4833	6.05	5.86	5.95
	0.7	5649	5327	5553	5.23	5.83	7.52
8	0.3	8493	8504	8314	14.72	14.03	13.5
	0.5	-	8514	8084	13.53	13.3	13.34
	0.7	8586	-	-	13.45	13.5	13.36
9	0.3	6322	5713	6512	13.39	14.03	13.45
	0.5	6194	6694	6522	16.62	13.97	15.17
	0.7	5590	6884	5707	15.62	13.94	13.58
10	0.3	13797	12526	13023	28.7	29.25	27.09
	0.5	13440	13344	13972	28.87	29.25	27.94
	0.7	13757	14241	13166	28.53	29.01	27.26
11	0.3	10960	12280	11255	37.83	37.22	39.42
	0.5	12164	11339	11525	37.18	37.29	37.54
	0.7	11762	12049	12659	37.03	37.17	38
12	0.3	18291	18910	17847	103.74	106.99	111.41
	0.5	18710	18440	18128	106.63	111.26	110.26
	0.7	18428	19208	18305	106.48	122.26	110.23
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	7237	-	19.06	23.06	20.23
	0.5	-	10031	-	21.53	19.89	21.23
	0.7	-	8539	-	21.17	23.65	20.72
15	0.3	-	-	-	55.78	50.01	50.04
	0.5	-	-	-	52.64	51.59	52.2
	0.7	-	-	-	51.03	52.14	45.73
16	0.3	-	-	-	26.76	26.89	26.67
	0.5	-	-	-	26.51	27.54	27.53
	0.7	-	-	-	26.67	26.61	27.79
17	0.3	-	-	-	51.93	51.17	51.62
	0.5	-	-	-	52.01	51.45	51.45
	0.7	-	-	-	51.7	52.17	51.82
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.2: Results of method 3 with the exact fitness function, pool size = 10 and second stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	913	820	1112	85.57	115.07	82.73
	0.5	808	713	1009	70.03	199.06	179.05
	0.7	809	812	1012	115.1	143.51	211.52
2	0.3	1520	1424	1524	600	600	600
	0.5	1240	1339	1930	600	600	259.84
	0.7	1619	1322	1644	600	600	600
3	0.3	2337	2732	2948	600	600	600
	0.5	2323	2838	2732	600	600	600
	0.7	2217	2649	3143	600	600	600
4	0.3	1936	2454	2042	600	600	600
	0.5	2454	2852	3040	600	600	600
	0.7	2341	2551	3050	600	600	600
5	0.3	3070	3482	3661	600	600	600
	0.5	3096	3678	3379	600	600	600
	0.7	2967	3279	3892	600	600	600
6	0.3	4882	4882	5183	600	600	600
	0.5	5693	5296	5599	600	600	600
	0.7	5502	5278	5112	600	600	600
7	0.3	4627	5426	5534	600	600	600
	0.5	5221	5436	5314	600	600	600
	0.7	5256	5441	4815	600	600	600
8	0.3	9002	-	8297	600	600	600
	0.5	7492	-	8892	600	600	600
	0.7	7669	8989	7587	600	600	600
9	0.3	5897	6422	6916	600	600	600
	0.5	5700	5190	6333	600	600	600
	0.7	5513	6175	4904	600	600	600
10	0.3	13872	13382	13069	600	600	600
	0.5	13037	13168	14165	600	600	600
	0.7	12544	13428	13143	600	600	600
11	0.3	11452	10860	11955	600	600	600
	0.5	11861	11382	11830	600	600	600
	0.7	11054	11554	11873	600	600	600
12	0.3	19260	17822	19352	600	600	600
	0.5	18248	18619	18651	600	600	600
	0.7	18932	18924	17502	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	10259	-	-	600	600	600
	0.5	9768	-	11373	600	600	600
	0.7	-	8857	11337	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.3: Results of method 3 with the exact fitness function, pool size = 10 and third stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	908	713	812	600	600	600
	0.5	809	1116	812	600	600	600
	0.7	903	812	808	600	600	600
2	0.3	1438	1427	1736	600	600	600
	0.5	1321	1329	1522	600	600	600
	0.7	1421	1433	1423	600	600	600
3	0.3	2935	2535	3556	600	600	600
	0.5	2630	3032	-	600	600	600
	0.7	-	2529	2947	600	600	600
4	0.3	2248	2452	2646	600	600	600
	0.5	2438	2348	2752	600	600	600
	0.7	2352	2148	3066	600	600	600
5	0.3	3576	3696	3599	600	600	600
	0.5	3886	3581	3577	600	600	600
	0.7	3286	3971	3382	600	600	600
6	0.3	4990	5486	5494	600	600	600
	0.5	4869	5390	5394	600	600	600
	0.7	4987	5194	5388	600	600	600
7	0.3	5129	5140	5250	600	600	600
	0.5	5324	4921	5044	600	600	600
	0.7	5234	5666	5027	600	600	600
8	0.3	-	-	7881	600	600	600
	0.5	9006	8069	-	600	600	600
	0.7	7880	-	8475	600	600	600
9	0.3	6395	6223	5108	600	600	600
	0.5	6720	5605	6313	600	600	600
	0.7	6340	6710	6102	600	600	600
10	0.3	12937	14052	13467	600	600	600
	0.5	12853	13673	14454	600	600	600
	0.7	12444	13483	13484	600	600	600
11	0.3	11788	10337	10634	600	600	600
	0.5	11662	10747	11653	600	600	600
	0.7	10977	12259	11165	600	600	600
12	0.3	18739	18359	18537	600	600	600
	0.5	19203	19231	18727	600	600	600
	0.7	18440	18415	19434	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	8637	9772	-	600	600	600
	0.5	8435	-	-	600	600	600
	0.7	7836	-	-	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.4: Results of method 3 with the exact fitness function, pool size = 20 and first stopping criterion.

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1008	714	918	106.49	207.86	227.86
	0.5	1107	813	612	167.46	159.18	541.27
	0.7	908	816	909	115.02	184.81	400.14
2	0.3	1325	1237	1434	600	600	600
	0.5	1529	1331	1727	600	600	600
	0.7	1528	1244	1536	600	600	600
3	0.3	2235	2517	2628	600	600	600
	0.5	2628	2829	-	600	600	600
	0.7	2526	2939	3129	600	600	600
4	0.3	2035	2045	2960	600	600	600
	0.5	2439	2439	2640	600	600	600
	0.7	2348	2250	2445	600	600	600
5	0.3	3073	3193	3592	600	600	600
	0.5	3284	3493	3403	600	600	600
	0.7	3281	3366	3177	600	600	600
6	0.3	4688	4886	5392	600	600	600
	0.5	4793	4909	5001	600	600	600
	0.7	4786	5182	5198	600	600	600
7	0.3	5153	5223	5434	600	600	600
	0.5	4761	5118	4321	600	600	600
	0.7	5225	4839	5050	600	600	600
8	0.3	-	7684	8679	600	600	600
	0.5	-	-	-	600	600	600
	0.7	8819	-	8479	600	600	600
9	0.3	6202	5378	5884	600	600	600
	0.5	5383	4985	6078	600	600	600
	0.7	5898	5910	6312	600	600	600
10	0.3	13643	12348	13436	600	600	600
	0.5	13246	12731	13058	600	600	600
	0.7	13757	13640	13282	600	600	600
11	0.3	9534	11158	11773	600	600	600
	0.5	11056	10957	11449	600	600	600
	0.7	10953	10754	11552	600	600	600
12	0.3	18605	18552	18612	600	600	600
	0.5	17994	18426	18321	600	600	600
	0.7	17616	18135	18542	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	9282	9241	600	600	600
	0.5	10668	9672	-	600	600	600
	0.7	-	10379	9361	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.5: Results of method 3 with the exact fitness function, pool size = 20 and second stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	910	914	907	87.74	141.1	192.69
	0.5	812	815	1011	120.79	129.62	137.58
	0.7	913	810	810	113.79	143.74	219.97
2	0.3	1321	1340	1631	600	600	600
	0.5	1329	1422	1425	600	600	600
	0.7	1321	1523	1534	600	600	600
3	0.3	2722	2957	2537	600	600	600
	0.5	2626	-	-	600	600	600
	0.7	2830	2939	2937	600	600	600
4	0.3	2350	2344	2668	600	600	600
	0.5	2136	2558	3072	600	600	600
	0.7	2355	2450	2959	600	600	600
5	0.3	2973	3987	3683	600	600	600
	0.5	3269	3100	3697	600	600	600
	0.7	3301	3372	3893	600	600	600
6	0.3	4779	4891	5500	600	600	600
	0.5	4577	5186	4892	600	600	600
	0.7	5092	5398	5293	600	600	600
7	0.3	5425	4230	5028	600	600	600
	0.5	4513	4402	5226	600	600	600
	0.7	5041	5148	5234	600	600	600
8	0.3	9000	-	7689	600	600	600
	0.5	-	-	8708	600	600	600
	0.7	-	8266	-	600	600	600
9	0.3	5912	5841	5697	600	600	600
	0.5	5921	6212	6214	600	600	600
	0.7	5994	6018	5315	600	600	600
10	0.3	12263	13763	12842	600	600	600
	0.5	13567	13354	12964	600	600	600
	0.7	13777	12867	13451	600	600	600
11	0.3	10638	11140	11348	600	600	600
	0.5	11172	11879	10358	600	600	600
	0.7	10860	10688	11253	600	600	600
12	0.3	18657	18151	18108	600	600	600
	0.5	19035	17105	19333	600	600	600
	0.7	18052	18951	18320	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	10583	-	10167	600	600	600
	0.5	9281	8843	9542	600	600	600
	0.7	9958	9169	10158	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.6: Results of method 3 with the exact fitness function, pool size = 20 and third stopping criterion.

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1009	808	907	600	600	600
	0.5	1008	709	808	600	600	600
	0.7	805	1205	808	600	600	600
2	0.3	1522	1323	1536	600	600	600
	0.5	1420	1629	1727	600	600	600
	0.7	1231	1436	1731	600	600	600
3	0.3	2529	2942	2739	600	600	600
	0.5	2732	2844	3132	600	600	600
	0.7	2826	2522	2942	600	600	600
4	0.3	1833	2857	2759	600	600	600
	0.5	2249	2451	2951	600	600	600
	0.7	2342	2646	2542	600	600	600
5	0.3	3359	3582	3596	600	600	600
	0.5	3191	3160	3486	600	600	600
	0.7	3386	3161	3685	600	600	600
6	0.3	5298	5081	5386	600	600	600
	0.5	4792	5583	5387	600	600	600
	0.7	4976	5697	4989	600	600	600
7	0.3	4732	4625	5556	600	600	600
	0.5	4911	5136	5147	600	600	600
	0.7	5129	5233	5036	600	600	600
8	0.3	8588	7577	9719	600	600	600
	0.5	8900	-	8206	600	600	600
	0.7	8089	-	9098	600	600	600
9	0.3	5902	5998	5614	600	600	600
	0.5	5493	5608	6205	600	600	600
	0.7	5904	6138	5400	600	600	600
10	0.3	13677	12675	13050	600	600	600
	0.5	12747	13496	13247	600	600	600
	0.7	13352	13332	13354	600	600	600
11	0.3	10141	9927	11073	600	600	600
	0.5	11364	11177	10541	600	600	600
	0.7	10754	10829	10966	600	600	600
12	0.3	16924	18638	18043	600	600	600
	0.5	17615	18140	18014	600	600	600
	0.7	18323	17928	18632	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	9978	-	600	600	600
	0.5	9042	-	-	600	600	600
	0.7	8650	11494	-	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.7: Results of method 3 with the relaxation fitness function, pool size = 10 and first stopping criterion.

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	907	-	809	60.0739	93.3682	62.4486
	0.5	-	-	810	78.0881	186.971	136.584
	0.7	-	1012	-	84.1189	84.5425	324.929
2	0.3	-	2545	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
3	0.3	3234	2220	2838	600	600	600
	0.5	3245	2213	2418	600	600	600
	0.7	-	-	-	600	600	600
4	0.3	2331	2037	2758	576.083	600	600
	0.5	2228	2029	2246	600	600	600
	0.7	2029	1941	2048	600	600	600
5	0.3	2167	2578	3171	600	600	600
	0.5	2073	2962	3290	600	600	600
	0.7	2558	3794	3297	600	600	600
6	0.3	4189	4494	5186	600	600	600
	0.5	4277	4468	4881	600	600	600
	0.7	4697	4699	4991	600	600	600
7	0.3	4031	4921	4955	600	600	600
	0.5	4833	4736	5126	600	600	600
	0.7	4729	4820	4736	600	600	600
8	0.3	8503	8989	8683	600	600	600
	0.5	8979	-	9003	600	600	600
	0.7	8403	8793	9003	600	600	600
9	0.3	5312	6505	6122	600	600	600
	0.5	5169	5699	5099	600	600	600
	0.7	5199	5790	6419	600	600	600
10	0.3	12957	13548	13265	600	600	600
	0.5	12927	13651	14069	600	600	600
	0.7	13042	13859	13367	600	600	600
11	0.3	10563	10441	10241	600	600	600
	0.5	11256	10959	10342	600	600	600
	0.7	11039	11066	11353	600	600	600
12	0.3	19124	18356	18841	600	600	600
	0.5	17375	19048	18045	600	600	600
	0.7	17921	18620	19523	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	-	9254	600	600	600
	0.5	11052	9149	9824	600	600	600
	0.7	9753	10880	11164	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.8: Results of method 3 with the relaxation fitness function, pool size = 10 and second stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	714	-	709	81.6034	87.3061	90.1028
	0.5	1007	-	-	49.5901	129.412	206.938
	0.7	1315	1207	709	46.0123	81.0097	141.724
2	0.3	-	-	-	600	600	600
	0.5	-	1828	-	600	600	600
	0.7	-	-	1737	600	600	365.38
3	0.3	1914	3638	2328	600	600	600
	0.5	-	2134	2531	600	600	600
	0.7	-	-	-	600	600	600
4	0.3	2131	2040	2241	600	600	600
	0.5	2242	-	2443	600	600	600
	0.7	2032	-	-	600	600	600
5	0.3	2048	2466	3084	600	600	600
	0.5	2876	2763	2973	600	600	600
	0.7	2772	2770	3572	600	600	600
6	0.3	4083	4369	4894	600	600	600
	0.5	4785	4388	4473	600	600	600
	0.7	3863	4772	4700	600	600	600
7	0.3	4022	4610	5052	600	600	600
	0.5	4225	4509	4726	600	600	600
	0.7	4624	4546	4760	600	600	600
8	0.3	9283	8894	8575	600	600	600
	0.5	8890	7701	8798	600	600	600
	0.7	7998	8485	8470	600	600	600
9	0.3	5569	5580	5597	600	600	600
	0.5	6103	6525	5481	600	600	600
	0.7	4983	6613	5290	600	600	600
10	0.3	13483	12956	12858	600	600	600
	0.5	13467	14066	13444	600	600	600
	0.7	13752	13882	12944	600	600	600
11	0.3	11676	11373	10753	600	600	600
	0.5	11775	11645	11039	600	600	600
	0.7	12345	10151	10960	600	600	600
12	0.3	18432	18723	17432	600	600	600
	0.5	17697	18841	19002	600	600	600
	0.7	19061	18524	18444	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	8581	-	10044	600	600	600
	0.5	8450	8933	-	600	600	600
	0.7	8872	-	9656	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.9: Results of method 3 with the relaxation fitness function, pool size = 10 and third stopping criterion.

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	-	812	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	909	1009	714	600	600	600
2	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
3	0.3	-	1915	2631	600	600	600
	0.5	3045	2026	2029	600	600	600
	0.7	-	2638	-	600	600	600
4	0.3	1832	1834	2347	600	600	600
	0.5	2132	1838	2148	600	600	600
	0.7	-	1935	2038	600	600	600
5	0.3	2551	2477	3276	600	600	600
	0.5	2457	2672	3174	600	600	600
	0.7	2578	2571	3066	600	600	600
6	0.3	4184	4375	4588	600	600	600
	0.5	4578	4477	4783	600	600	600
	0.7	5086	4369	4586	600	600	600
7	0.3	4415	4326	3840	600	600	600
	0.5	4620	4512	4400	600	600	600
	0.7	4036	4751	4121	600	600	600
8	0.3	9187	7797	7605	600	600	600
	0.5	7817	7895	9106	600	600	600
	0.7	-	9094	8192	600	600	600
9	0.3	6000	6105	5165	600	600	600
	0.5	4876	5109	5805	600	600	600
	0.7	6201	6206	5816	600	600	600
10	0.3	13355	12721	13046	600	600	600
	0.5	13637	13238	14377	600	600	600
	0.7	13772	13877	12949	600	600	600
11	0.3	11444	12376	10239	600	600	600
	0.5	11578	11477	10658	600	600	600
	0.7	12249	11352	11176	600	600	600
12	0.3	17947	18791	19041	600	600	600
	0.5	18295	19346	18531	600	600	600
	0.7	18665	18102	19039	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	9736	8848	600	600	600
	0.5	9250	9438	10174	600	600	600
	0.7	-	-	9150	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.10: Results of method 3 with the relaxation fitness function, pool size = 20 and first stopping criterion.

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1012	-	813	76.4476	101.443	127.303
	0.5	609	1011	706	61.1049	86.8687	115.632
	0.7	-	-	-	123.319	197.814	333.398
2	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	2739	-	-	600	600	600
3	0.3	1607	2227	2220	600	600	600
	0.5	1620	2018	-	600	600	600
	0.7	-	-	-	600	600	600
4	0.3	-	1836	2251	600	600	600
	0.5	1948	2038	2148	600	600	600
	0.7	1742	1943	2237	600	600	600
5	0.3	2354	2367	2578	600	600	600
	0.5	2572	2973	3167	600	600	600
	0.7	2463	2759	3274	600	600	600
6	0.3	4578	5079	4582	600	600	600
	0.5	4673	4384	4978	600	600	600
	0.7	4075	4775	4475	600	600	600
7	0.3	3517	4324	4739	600	600	600
	0.5	3838	4933	4723	600	600	600
	0.7	4436	4426	4919	600	600	600
8	0.3	6900	7590	8315	600	600	600
	0.5	-	-	8283	600	600	600
	0.7	8382	-	8678	600	600	600
9	0.3	5521	5608	5582	600	600	600
	0.5	5998	5519	5679	600	600	600
	0.7	4899	5506	4665	600	600	600
10	0.3	11734	13169	13568	600	600	600
	0.5	13148	12309	12661	600	600	600
	0.7	13040	13345	13461	600	600	600
11	0.3	11158	11161	10347	600	600	600
	0.5	10457	11050	11156	600	600	600
	0.7	11058	11249	10468	600	600	600
12	0.3	18231	18499	19002	600	600	600
	0.5	19332	18540	17936	600	600	600
	0.7	18648	18242	18426	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	9576	9563	9678	600	600	600
	0.5	8419	8039	9449	600	600	600
	0.7	-	8626	-	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.11: Results of method 3 with the relaxation fitness function, pool size = 20 and second stopping criterion

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	910	-	812	38.4035	102.789	177.909
	0.5	806	-	907	73.1978	136.771	145.036
	0.7	912	807	-	57.9802	113.164	306.79
2	0.3	-	-	-	600	600	600
	0.5	2665	-	-	600	600	600
	0.7	-	-	-	600	600	600
3	0.3	2534	2526	2231	600	600	600
	0.5	-	-	2433	600	600	600
	0.7	1517	2516	2538	600	600	600
4	0.3	-	1941	2042	597.395	600	600
	0.5	-	2137	2243	600	600	600
	0.7	2034	2139	2138	600	600	600
5	0.3	2256	2170	3272	600	600	600
	0.5	2359	2772	3076	600	600	600
	0.7	2476	2582	2973	600	600	600
6	0.3	4257	4284	4993	600	600	600
	0.5	4066	3958	5086	600	600	600
	0.7	3965	4598	4990	600	600	600
7	0.3	4617	4731	5032	600	600	600
	0.5	4435	4436	4919	600	600	600
	0.7	4414	4318	4643	600	600	600
8	0.3	8186	8475	7893	600	600	600
	0.5	8486	7981	7187	600	600	600
	0.7	8310	8511	-	600	600	600
9	0.3	5099	5593	5400	600	600	600
	0.5	5675	5324	5920	600	600	600
	0.7	4553	4989	5567	600	600	600
10	0.3	13350	13341	13029	600	600	600
	0.5	12942	13371	13084	600	600	600
	0.7	13255	12553	13357	600	600	600
11	0.3	10338	11355	11046	600	600	600
	0.5	11047	10954	9927	600	600	600
	0.7	10552	11067	11271	600	600	600
12	0.3	17854	18404	18047	600	600	600
	0.5	19132	18415	18635	600	600	600
	0.7	17512	18713	18736	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	9328	9720	10175	600	600	600
	0.5	9418	9349	7446	600	600	600
	0.7	10242	9349	-	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.12: Results of method 3 with the relaxation fitness function, pool size = 20 and third stopping criterion.

	HMCR	Best objective value 0.3	0.5	0.7	Total solving time (s) 0.3	0.5	0.7
Instance	PAR						
1	0.3	811	-	806	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
2	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
3	0.3	-	-	3246	600	600	600
	0.5	2326	3347	3032	600	600	600
	0.7	1830	-	2234	600	600	600
4	0.3	1831	1945	2246	600	600	600
	0.5	-	1842	2135	600	600	600
	0.7	1940	2143	2250	600	600	600
5	0.3	2362	3065	3195	600	600	600
	0.5	2354	2570	2886	600	600	600
	0.7	2259	2477	2986	600	600	600
6	0.3	4372	4570	4686	600	600	600
	0.5	3771	4699	4690	600	600	600
	0.7	4482	4492	4797	600	600	600
7	0.3	3615	4539	3829	600	600	600
	0.5	4609	3830	5142	600	600	600
	0.7	4116	4019	4927	600	600	600
8	0.3	8091	7510	7391	600	600	600
	0.5	7686	7093	8922	600	600	600
	0.7	-	8081	8892	600	600	600
9	0.3	5997	5909	5908	600	600	600
	0.5	5520	4785	5717	600	600	600
	0.7	6137	5470	5404	600	600	600
10	0.3	13052	13236	13038	600	600	600
	0.5	12254	13067	12859	600	600	600
	0.7	13173	13251	13262	600	600	600
11	0.3	11367	10852	11574	600	600	600
	0.5	11244	12361	10933	600	600	600
	0.7	10847	10048	11335	600	600	600
12	0.3	17545	17721	19356	600	600	600
	0.5	18024	18100	18339	600	600	600
	0.7	18933	18302	18913	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	9852	8446	600	600	600
	0.5	-	9670	9748	600	600	600
	0.7	-	8841	9355	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.13: Results of method 3 with the heuristic fitness function, pool size = 10 and first stopping criterion.

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	806	909	1008	396.36	414.7	600
	0.5	1013	911	814	558.04	414.25	600
	0.7	815	810	911	544.56	534.23	600
2	0.3	1634	1642	2246	600	600	600
	0.5	1838	1754	2040	600	600	600
	0.7	2038	1935	2255	600	600	600
3	0.3	-	3274	2845	600	600	600
	0.5	2735	-	3865	600	600	600
	0.7	2731	3582	3876	600	600	600
4	0.3	3789	3675	3686	600	600	600
	0.5	3468	3669	3765	600	600	600
	0.7	3758	3673	3870	600	600	600
5	0.3	4196	3968	4102	600	600	600
	0.5	3987	4121	3887	600	600	600
	0.7	3789	4509	4295	600	600	600
6	0.3	5906	5604	5399	600	600	600
	0.5	5500	5894	5596	600	600	600
	0.7	5705	5295	6001	600	600	600
7	0.3	5448	5345	4744	600	600	600
	0.5	5831	5119	5538	600	600	600
	0.7	4731	5631	4638	600	600	600
8	0.3	-	-	8707	600	600	600
	0.5	-	9416	8606	600	600	600
	0.7	7899	8275	-	600	600	600
9	0.3	5506	6006	6223	600	600	600
	0.5	6217	6506	6008	600	600	600
	0.7	5724	5993	6609	600	600	600
10	0.3	13195	13547	14174	600	600	600
	0.5	13746	13569	12968	600	600	600
	0.7	13246	13587	13735	600	600	600
11	0.3	12044	11169	11656	600	600	600
	0.5	11263	12036	10829	600	600	600
	0.7	11368	10850	10749	600	600	600
12	0.3	18534	17729	17249	600	600	600
	0.5	19600	18234	17909	600	600	600
	0.7	19324	18252	19025	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	-	9364	9843	600	600	600
	0.5	11396	9755	-	600	600	600
	0.7	10684	9940	-	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.14: Results of method 3 with the heuristic fitness function, pool size = 10 and second stopping criterion

	HMCR	Best objective value 0.3	0.5	0.7	Total solving time (s) 0.3	0.5	0.7
Instance	PAR						
1	0.3	910	807	912	191.13	369.29	342.1
	0.5	904	907	709	351.41	354.8	393.09
	0.7	907	1008	815	385.96	555.28	286.99
2	0.3	1656	1837	1935	600	599.76	327.18
	0.5	2455	2047	2143	294.33	397.46	532.47
	0.7	1547	2345	2235	600	383.43	600
3	0.3	2843	3685	3463	600	600	600
	0.5	3977	2930	2847	600	600	600
	0.7	2650	3465	2842	600	600	600
4	0.3	3367	3571	3566	600	600	600
	0.5	3565	3271	3563	600	600	600
	0.7	3466	3567	3559	600	600	600
5	0.3	3377	3976	3897	600	600	600
	0.5	3600	4202	3894	600	600	600
	0.7	4598	4202	4297	600	600	600
6	0.3	5587	5607	5611	600	600	600
	0.5	5495	5713	5497	600	600	600
	0.7	5291	5803	5801	600	600	600
7	0.3	5864	5427	5941	600	600	600
	0.5	5555	5239	5644	600	600	600
	0.7	5361	5439	5930	600	600	600
8	0.3	-	8607	7786	600	600	600
	0.5	8303	-	-	600	600	600
	0.7	-	8498	-	600	600	600
9	0.3	5901	6511	6104	600	600	600
	0.5	6210	6115	5900	600	600	600
	0.7	5988	6613	6726	600	600	600
10	0.3	14251	13262	13381	600	600	600
	0.5	13056	13767	12561	600	600	600
	0.7	13559	12733	13675	600	600	600
11	0.3	11675	11380	12356	600	600	600
	0.5	10839	11750	10855	600	600	600
	0.7	12062	11779	11854	600	600	600
12	0.3	17717	19332	18840	600	600	600
	0.5	18323	18346	19341	600	600	600
	0.7	20466	18413	18615	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	11355	-	-	600	600	600
	0.5	-	10245	9665	600	600	600
	0.7	-	-	-	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.15: Results of method 3 with the heuristic fitness function, pool size = 10 and third stopping criterion.

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1007	1210	711	600	600	600
	0.5	1004	813	913	600	600	600
	0.7	710	914	911	600	600	600
2	0.3	1827	1933	1746	600	600	600
	0.5	1648	1940	2050	600	600	600
	0.7	1942	2144	2248	600	600	600
3	0.3	3783	3364	2527	600	600	600
	0.5	2549	3245	3475	600	600	600
	0.7	-	3862	3140	600	600	600
4	0.3	3186	3368	3361	600	600	600
	0.5	3461	3370	3577	600	600	600
	0.7	3983	3661	3671	600	600	600
5	0.3	3898	3800	4293	600	600	600
	0.5	4500	4193	4288	600	600	600
	0.7	3679	4105	4383	600	600	600
6	0.3	5604	5100	5695	600	600	600
	0.5	5508	5695	5802	600	600	600
	0.7	5781	5588	5694	600	600	600
7	0.3	5832	5545	5230	600	600	600
	0.5	5449	5554	5544	600	600	600
	0.7	5434	5645	5635	600	600	600
8	0.3	-	8084	8598	600	600	600
	0.5	7991	8880	8675	600	600	600
	0.7	-	10116	8881	600	600	600
9	0.3	6014	6303	6417	600	600	600
	0.5	6017	5539	5909	600	600	600
	0.7	6373	6000	6091	600	600	600
10	0.3	14474	12753	13437	600	600	600
	0.5	13070	13150	14369	600	600	600
	0.7	13632	13150	13029	600	600	600
11	0.3	10360	10657	11345	600	600	600
	0.5	11362	11463	11588	600	600	600
	0.7	12170	11436	12276	600	600	600
12	0.3	18538	18012	17536	600	600	600
	0.5	19141	19023	18550	600	600	600
	0.7	18832	19120	18222	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	11082	10145	9643	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	9742	-	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.16: Results of method 3 with the heuristic fitness function, pool size = 20 and first stopping criterion.

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1010	1111	909	384.17	262.55	425.81
	0.5	908	711	816	521.97	442.95	600
	0.7	814	815	811	600	600	600
2	0.3	1741	1735	1644	600	600	600
	0.5	1752	1934	1933	600	600	600
	0.7	1843	1933	1944	600	600	600
3	0.3	2633	-	3147	600	600	600
	0.5	2729	2835	3476	600	600	600
	0.7	4183	-	3782	600	600	600
4	0.3	3481	3665	3456	600	600	600
	0.5	3366	3669	3167	600	600	600
	0.7	3468	3580	3477	600	600	600
5	0.3	3573	4205	4321	600	600	600
	0.5	3801	4198	3680	600	600	600
	0.7	4393	3596	3796	600	600	600
6	0.3	5798	5186	5700	600	600	600
	0.5	5706	6008	5788	600	600	600
	0.7	5702	5895	5487	600	600	600
7	0.3	5647	4833	5140	600	600	600
	0.5	5749	5368	5450	600	600	600
	0.7	4824	5348	4826	600	600	600
8	0.3	-	8003	7767	600	600	600
	0.5	8089	9088	8494	600	600	600
	0.7	7905	9487	-	600	600	600
9	0.3	5705	5504	5947	600	600	600
	0.5	5390	5808	5803	600	600	600
	0.7	5985	5724	5596	600	600	600
10	0.3	13069	13301	13070	600	600	600
	0.5	13043	13355	12852	600	600	600
	0.7	13378	13528	13654	600	600	600
11	0.3	10948	10966	11149	600	600	600
	0.5	10935	10959	10035	600	600	600
	0.7	11230	10840	11175	600	600	600
12	0.3	18550	17834	18653	600	600	600
	0.5	16796	18341	17992	600	600	600
	0.7	18357	18050	18820	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	8974	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	10162	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.17: Results of method 3 with the heuristic fitness function, pool size = 20 and second stopping criterion

	HMCR	Best objective value 0.3	0.5	0.7	Total solving time (s) 0.3	0.5	0.7
Instance	PAR						
1	0.3	719	1007	618	161.18	259.06	549.19
	0.5	908	911	814	267.88	600	600
	0.7	711	816	808	547.18	452.49	494.68
2	0.3	1939	1842	1943	600	600	600
	0.5	2046	1948	1934	600	600	600
	0.7	1752	1839	1948	600	600	600
3	0.3	-	2940	3964	600	600	600
	0.5	2745	3578	3569	600	600	600
	0.7	4175	3149	3136	600	600	600
4	0.3	3373	3364	3272	600	600	600
	0.5	3566	3265	3473	600	600	600
	0.7	3569	3565	3475	600	600	600
5	0.3	3774	3389	3801	600	600	600
	0.5	3978	4287	3283	600	600	600
	0.7	3907	3698	4595	600	600	600
6	0.3	5699	5803	5398	600	600	600
	0.5	5899	5489	5494	600	600	600
	0.7	5910	5491	5606	600	600	600
7	0.3	4620	5042	5356	600	600	600
	0.5	5335	5231	5167	600	600	600
	0.7	5031	5330	5244	600	600	600
8	0.3	8400	8891	-	600	600	600
	0.5	-	8492	-	600	600	600
	0.7	7962	-	-	600	600	600
9	0.3	6109	6404	4995	600	600	600
	0.5	6600	6096	6121	600	600	600
	0.7	5615	5790	5505	600	600	600
10	0.3	13500	12447	13125	600	600	600
	0.5	13271	13268	13047	600	600	600
	0.7	13660	13444	13127	600	600	600
11	0.3	11068	11262	11363	600	600	600
	0.5	11445	11569	11764	600	600	600
	0.7	10934	10964	10952	600	600	600
12	0.3	18816	18800	19232	600	600	600
	0.5	18062	18036	18226	600	600	600
	0.7	18314	18533	18834	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	11595	-	-	600	600	600
	0.5	10878	-	-	600	600	600
	0.7	9858	10261	10171	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

Table D.18: Results of method 3 with the heuristic fitness function, pool size = 20 and third stopping criterion.

		Best objective value			Total solving time (s)		
	HMCR	0.3	0.5	0.7	0.3	0.5	0.7
Instance	PAR						
1	0.3	1007	1006	908	600	600	600
	0.5	905	814	808	600	600	600
	0.7	815	907	915	600	600	600
2	0.3	1853	1837	2036	600	600	600
	0.5	1935	1948	1951	600	600	600
	0.7	1853	2243	2053	600	600	600
3	0.3	3575	2840	3672	600	600	600
	0.5	4059	3453	-	600	600	600
	0.7	-	-	-	600	600	600
4	0.3	3675	3279	3565	600	600	600
	0.5	3471	3463	3560	600	600	600
	0.7	3150	3766	3270	600	600	600
5	0.3	3995	4204	4005	600	600	600
	0.5	4102	3811	4120	600	600	600
	0.7	3991	4083	3990	600	600	600
6	0.3	5890	5594	5596	600	600	600
	0.5	5694	5792	4995	600	600	600
	0.7	5696	5896	5518	600	600	600
7	0.3	5551	5450	5664	600	600	600
	0.5	5243	4839	5341	600	600	600
	0.7	5249	5148	5550	600	600	600
8	0.3	7973	9100	8912	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	8488	600	600	600
9	0.3	5294	5814	6287	600	600	600
	0.5	6299	6303	6409	600	600	600
	0.7	5517	5700	5742	600	600	600
10	0.3	13044	13354	12556	600	600	600
	0.5	13466	13242	13161	600	600	600
	0.7	12832	12958	12452	600	600	600
11	0.3	11774	11267	9148	600	600	600
	0.5	11160	11153	11571	600	600	600
	0.7	12014	10644	10644	600	600	600
12	0.3	18436	18543	18169	600	600	600
	0.5	18414	18531	17119	600	600	600
	0.7	17706	18829	17937	600	600	600
13	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
14	0.3	9390	11260	-	600	600	600
	0.5	-	9063	9681	600	600	600
	0.7	-	-	-	600	600	600
15	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
16	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
17	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
19	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
20	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
21	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
22	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
23	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600
24	0.3	-	-	-	600	600	600
	0.5	-	-	-	600	600	600
	0.7	-	-	-	600	600	600

E

Results of solving random solutions to the first subproblem with the three difference fitness functions

Table E.1: Objective values and solving times of solving 10 random solutions with the exact, relaxation and heuristic fitness function for each instance

Instance	Solution	Objective value	Solving time (s)	Objective value	Solving time (s)	Objective value	Solving time (s)
1	1	**2227.00**	1.06	**2227.00**	0.02	**2227.00**	0.02
	2	**2230.00**	0.09	**2230.00**	0.05	**2230.00**	0.05
	3	**1924.00**	0.11	**1924.00**	0.05	**1924.00**	0.03
	4	**2336.00**	0.12	**2336.00**	0.06	**2336.00**	0.03
	5	**2231.00**	0.12	**2231.00**	0.05	**2231.00**	0.05
	6	**1923.00**	0.11	**1923.00**	0.05	**1923.00**	0.06
	7	**2426.00**	0.09	**2426.00**	0.98	**2426.00**	0.02
	8	**1923.00**	0.05	**1923.00**	0.02	**1923.00**	0.02
	9	**2035.00**	0.09	**2035.00**	0.05	**2035.00**	0.05
	10	**2126.00**	0.11	**2126.00**	0.03	**2126.00**	0.05
	Mean	*2138.10*	*0.20*	*2138.10*	*0.14*	*2138.10*	*0.04*
2	1	**3158.00**	0.31	**3158.00**	0.11	3984.00	0.05
	2	**2744.00**	0.31	**2744.00**	0.11	3873.00	0.06
	3	**3156.00**	0.23	**3156.00**	0.11	4387.00	0.03
	4	**3059.00**	1.06	**3059.00**	0.11	4394.00	0.06
	5	**3050.00**	0.33	**3050.00**	0.11	4481.00	0.08
	6	**3248.00**	0.33	**3248.00**	0.11	3366.00	0.08
	7	**2451.00**	0.25	**2451.00**	0.09	3375.00	0.05
	8	**3242.00**	0.23	**3242.00**	0.08	4064.00	0.05
	9	**2558.00**	0.18	**2558.00**	0.06	3876.00	0.03
	10	**2853.00**	1.11	**2853.00**	0.14	3983.00	0.08
	Mean	*2951.90*	*0.43*	*2951.90*	*0.10*	*3978.30*	*0.06*
3	1	**4649.00**	0.92	**4649.00**	0.22	5684.00	0.11
	2	**4051.00**	0.91	**4051.00**	0.15	4976.00	0.06
	3	**3758.00**	0.87	**3758.00**	0.12	4895.00	0.05
	4	**3860.00**	0.41	**3860.00**	0.09	4577.00	0.05
	5	**3655.00**	0.33	**3655.00**	0.09	5289.00	0.03
	6	**4456.00**	0.30	**4456.00**	1.02	5087.00	0.11
	7	**3754.00**	1.27	**3754.00**	0.22	5198.00	0.09
	8	**3953.00**	0.53	**3953.00**	0.14	4777.00	0.06
	9	**4056.00**	0.53	**4056.00**	0.12	5491.00	0.05

	10	**3850.00**	0.37	**3850.00**	0.11	4774.00	0.05
	Mean	*4004.20*	*0.64*	*4004.20*	*0.23*	*5074.80*	*0.07*
4	1	**4063.00**	0.23	**4063.00**	0.08	5823.00	0.09
	2	**5069.00**	0.22	**5069.00**	0.06	6527.00	0.42
	3	**4276.00**	0.61	**4276.00**	0.22	6937.00	0.20
	4	**4172.00**	0.59	**4172.00**	0.20	6240.00	0.19
	5	**3966.00**	0.42	**3966.00**	0.14	5833.00	0.12
	6	**4162.00**	0.29	**4162.00**	0.12	6439.00	0.12
	7	**4077.00**	0.30	**4077.00**	0.11	5737.00	0.09
	8	**5195.00**	0.25	**5195.00**	0.08	6866.00	0.09
	9	**4575.00**	0.89	**4575.00**	0.20	7047.00	0.27
	10	**4368.00**	0.47	**4368.00**	0.17	6649.00	0.22
	Mean	*4392.30*	*0.43*	*4392.30*	*0.14*	*6409.80*	*0.18*
5	1	**5703.00**	0.70	**5703.00**	0.17	7270.00	0.18
	2	**5093.00**	0.50	**5093.00**	0.16	6551.00	0.17
	3	**4899.00**	0.39	**4899.00**	0.12	6272.00	0.14
	4	**5190.00**	0.37	**5190.00**	0.12	7363.00	0.11
	5	**5202.00**	0.31	**5202.00**	0.73	6749.00	0.30
	6	**4710.00**	1.16	**4710.00**	0.23	7470.00	0.23
	7	**4501.00**	0.52	**4501.00**	0.17	5851.00	0.16
	8	**5497.00**	0.41	**5497.00**	0.12	6673.00	0.14
	9	**5301.00**	0.41	**5301.00**	0.12	6666.00	0.14
	10	**5100.00**	0.33	**5100.00**	0.11	7490.00	0.11
	Mean	*5119.60*	*0.51*	*5119.60*	*0.21*	*6835.50*	*0.17*
6	1	**7021.00**	0.52	**7021.00**	0.16	10916.00	0.30
	2	**6902.00**	1.10	**6902.00**	0.25	11401.00	0.17
	3	**6505.00**	0.73	6404.00	0.25	10099.00	0.12
	4	**6608.00**	0.59	**6608.00**	0.19	10403.00	0.12
	5	**6508.00**	0.53	6407.00	0.11	11200.00	0.08
	6	**6605.00**	0.37	**6605.00**	0.14	11311.00	0.08
	7	**5912.00**	0.39	**5912.00**	0.11	10732.00	0.11
	8	**7202.00**	1.04	7130.00	0.25	12925.00	0.17
	9	**6696.00**	0.67	**6696.00**	0.19	12025.00	0.16
	10	**6706.00**	0.59	**6706.00**	0.16	11540.00	0.12
	Mean	*6666.50*	*0.65*	*6639.10*	*0.18*	*11255.20*	*0.14*
7	1	**5734.00**	0.98	5633.00	0.14	8755.00	0.08
	2	**7537.00**	0.52	7536.00	0.12	10173.00	0.09
	3	**5953.00**	0.46	**5953.00**	0.11	8257.00	0.09
	4	**7475.00**	0.48	7374.00	0.33	10083.00	0.19
	5	**6953.00**	1.03	6852.00	0.20	9764.00	0.16
	6	**6962.00**	0.78	**6962.00**	0.19	9372.00	0.12
	7	**6644.00**	1.39	6543.00	0.12	9391.00	0.11
	8	**6982.00**	0.50	**6982.00**	0.11	9173.00	0.09
	9	**7061.00**	0.48	**7061.00**	0.11	9683.00	0.09
	10	**6666.00**	0.48	**6666.00**	0.12	8689.00	0.19
	Mean	*6796.70*	*0.71*	*6756.20*	*0.16*	*9334.00*	*0.12*
8	1	**-**	1.72	**-**	0.36	**-**	1.91
	2	**-**	1.11	**-**	0.33	**-**	1.87
	3	**-**	1.09	**-**	0.35	**-**	2.12
	4	**-**	1.05	**-**	0.34	**-**	2.16

	5	**-**	1.08	**-**	0.36	**-**	2.11
	6	**-**	1.11	**-**	0.37	**-**	2.09
	7	**-**	1.42	**-**	0.37	**-**	2.36
	8	**-**	1.05	**-**	0.34	**-**	2.08
	9	**-**	1.07	**-**	0.33	**-**	1.95
	10	**-**	1.06	**-**	0.38	**-**	1.94
	Mean	-	*1.18*	-	*0.35*	-	*2.06*
9	1	**6615.00**	1.90	**6615.00**	0.26	11293.00	0.27
	2	**7396.00**	1.32	**7396.00**	0.23	11536.00	0.27
	3	**7108.00**	1.36	**7108.00**	0.36	13102.00	0.25
	4	**7746.00**	1.16	**7746.00**	0.25	10723.00	0.22
	5	**7217.00**	1.16	7216.00	0.25	10439.00	0.23
	6	**7085.00**	1.03	**7085.00**	0.25	11374.00	0.17
	7	**7401.00**	1.31	**7401.00**	0.24	13544.00	0.13
	8	**8459.00**	1.14	**8459.00**	0.27	13739.00	0.14
	9	**8106.00**	1.47	8005.00	0.25	13057.00	0.19
	10	**7223.00**	1.22	7220.00	0.27	12896.00	0.19
	Mean	*7435.60*	*1.31*	*7425.10*	*0.26*	*12170.30*	*0.21*
10	1	**15197.00**	2.25	**15197.00**	0.61	25150.00	0.25
	2	**14581.00**	2.56	**14581.00**	0.64	23679.00	0.31
	3	**14682.00**	2.37	**14682.00**	0.62	25815.00	0.49
	4	**16595.00**	2.28	**16595.00**	0.58	26990.00	0.30
	5	**14452.00**	2.41	**14452.00**	0.75	26853.00	0.62
	6	**14273.00**	3.08	**14273.00**	0.67	27223.00	0.37
	7	**15689.00**	2.96	**15689.00**	0.58	25624.00	0.34
	8	**14479.00**	2.42	**14479.00**	0.63	22923.00	0.67
	9	**14497.00**	2.34	**14497.00**	0.64	24035.00	0.37
	10	**13858.00**	2.35	**13858.00**	0.80	25783.00	0.30
	Mean	*14830.30*	*2.50*	*14830.30*	*0.65*	*25407.50*	*0.40*
11	1	**13374.00**	3.42	**13374.00**	0.72	33626.00	0.31
	2	**12651.00**	3.39	**12651.00**	0.72	32921.00	0.31
	3	**13891.00**	3.44	**13891.00**	0.75	37355.00	0.29
	4	**13172.00**	3.78	**13172.00**	0.71	34580.00	0.31
	5	**12352.00**	3.17	**12352.00**	0.71	34513.00	0.31
	6	**13982.00**	3.47	**13982.00**	0.73	35422.00	0.33
	7	**12670.00**	3.05	**12670.00**	0.73	33343.00	0.30
	8	**12259.00**	3.58	**12259.00**	0.77	33518.00	0.31
	9	**12856.00**	3.44	**12856.00**	0.72	36437.00	0.31
	10	**13989.00**	3.03	**13989.00**	0.74	34491.00	0.31
	Mean	*13119.60*	*3.38*	*13119.60*	*0.73*	*34620.60*	*0.31*
12	1	**20731.00**	8.19	**20731.00**	1.86	41965.00	0.44
	2	**18927.00**	8.22	**18927.00**	1.97	43066.00	0.47
	3	**18432.00**	9.28	**18432.00**	2.12	42138.00	0.45
	4	**20552.00**	9.69	**20552.00**	1.95	41569.00	0.44
	5	**19443.00**	10.35	**19443.00**	1.94	43540.00	0.41
	6	**19766.00**	8.30	**19766.00**	1.89	43379.00	0.42
	7	**19738.00**	10.21	**19738.00**	2.01	45598.00	0.52
	8	**20663.00**	9.78	**20663.00**	2.04	44581.00	0.48
	9	**18966.00**	8.90	**18966.00**	2.19	44358.00	0.52
	10	**19537.00**	10.18	**19537.00**	2.27	42346.00	0.50
	Mean	*19675.50*	*9.31*	*19675.50*	*2.02*	*43254.00*	*0.47*

13	1	-	179.00	31221.00	136.71	-	3.13
	2	-	448.51	35951.00	421.32	-	4.94
	3	-	899.64	34234.00	268.34	-	6.00
	4	-	673.63	39672.00	242.86	-	4.75
	5	-	605.42	37616.00	211.20	-	5.47
	6	-	206.49	31583.00	208.58	-	4.06
	7	-	118.57	35167.00	333.17	-	2.39
	8	-	572.34	36543.00	220.36	-	5.17
	9	-	670.89	37427.00	259.61	-	6.50
	10	-	242.74	33762.00	311.11	-	6.36
	Mean	-	*461.72*	*35317.60*	*261.33*	-	*4.88*
14	1	-	3.41	-	1.37	-	11.78
	2	-	4.58	-	1.41	-	7.69
	3	-	3.36	-	0.77	-	7.66
	4	-	1.61	-	0.56	-	4.66
	5	-	1.52	-	0.61	-	3.75
	6	-	1.61	-	0.64	-	3.94
	7	-	1.12	-	0.48	-	4.67
	8	-	1.69	-	0.53	-	5.05
	9	-	1.64	-	0.53	-	4.58
	10	-	1.72	-	0.59	-	4.56
	Mean	-	*2.23*	-	*0.75*	-	*5.83*
15	1	-	4.80	-	13.51	-	4.14
	2	-	4.49	-	33.58	-	4.92
	3	-	4.70	-	28.16	-	4.90
	4	-	4.73	-	23.71	-	7.34
	5	-	5.14	-	60.50	-	5.15
	6	-	8.00	-	49.55	-	9.58
	7	-	5.01	-	41.34	-	4.27
	8	-	3.37	-	3.93	-	4.63
	9	-	4.69	-	39.16	-	4.45
	10	-	3.31	-	2.58	-	4.51
	Mean	-	*4.82*	-	*29.60*	-	*5.39*
16	1	-	1.11	-	0.33	-	6.17
	2	-	1.16	-	0.34	-	6.09
	3	-	1.13	-	0.33	-	5.90
	4	-	1.14	-	0.33	-	5.87
	5	-	1.11	-	0.34	-	5.88
	6	-	1.11	-	0.34	-	5.86
	7	-	1.12	-	0.36	-	6.46
	8	-	1.16	-	0.33	-	6.71
	9	-	1.22	-	0.32	-	6.70
	10	-	1.19	-	0.33	-	6.00
	Mean	-	*1.15*	-	*0.34*	-	*6.16*
17	1	-	3.04	-	1.28	-	7.33
	2	-	2.86	-	1.18	-	7.77
	3	-	2.95	-	1.25	-	6.71
	4	-	2.92	-	1.30	-	5.66
	5	-	3.28	-	1.73	-	6.16
	6	-	2.90	-	1.43	-	6.21

	7	-	2.97	-	1.14	-	6.13
	8	-	2.87	-	1.23	-	6.18
	9	-	3.03	-	1.20	-	7.14
	10	-	3.33	-	1.14	-	6.47
	Mean	-	*3.02*	-	*1.29*	-	*6.58*
18	1	-	2.17	-	0.58	-	11.49
	2	-	1.95	-	0.68	-	11.56
	3	-	1.97	-	0.58	-	11.63
	4	-	1.97	-	0.72	-	10.85
	5	-	2.26	-	0.62	-	12.13
	6	-	2.17	-	0.75	-	11.17
	7	-	1.98	-	0.50	-	11.41
	8	-	2.02	-	0.77	-	11.66
	9	-	2.15	-	0.61	-	11.31
	10	-	1.98	-	0.56	-	11.48
	Mean	-	*2.06*	-	*0.64*	-	*11.47*
19	1	-	7.68	-	3.97	-	12.89
	2	-	9.13	-	3.46	-	15.46
	3	-	10.10	-	5.59	-	15.10
	4	-	10.81	-	3.45	-	13.71
	5	-	7.52	-	2.20	-	14.28
	6	-	8.33	-	2.94	-	14.31
	7	-	10.28	-	3.14	-	15.30
	8	-	10.61	-	3.47	-	17.52
	9	-	7.67	-	2.50	-	13.30
	10	-	8.06	-	2.65	-	12.13
	Mean	-	*9.02*	-	*3.34*	-	*14.40*
21	1	-	94.70	-	1039.25	-	90.99
	2	-	137.86	-	932.17	-	72.42
	3	-	89.27	-	741.67	-	80.50
	4	-	102.86	-	757.60	-	54.83
	5	-	76.71	-	596.13	-	22.64
	6	-	72.11	-	549.30	-	45.07
	7	-	65.95	-	796.62	-	85.05
	8	-	133.29	-	866.78	-	72.25
	9	-	110.73	-	713.98	-	74.36
	10	-	96.06	-	864.38	-	66.13
	Mean	-	*97.95*	-	*785.79*	-	*66.42*