



An evolutionary approach for the nurse rerostering problem

Broos Maenhout^{a,*}, Mario Vanhoucke^{a,b}

^a Faculty of Economics and Business Administration, Ghent University, Tweekerkenstraat 2, 9000 Gent, Belgium

^b Operations and Technology Management Centre, Vlerick Leuven Gent Management School, Reep 1, 9000 Gent, Belgium

ARTICLE INFO

Available online 31 December 2010

Keywords:

Nurse rostering

Rerostering

Genetic algorithms

ABSTRACT

The personnel scheduler constructs a deterministic personnel roster that determines the line-of-work for each personnel member. When unexpected events disrupt this roster, the feasibility needs to be restored by constructing a new workable roster. The scheduler must reassign the set of employees in order to cover the disrupted shift such that the staffing requirements and the time-related personnel constraints remain satisfied. In this paper, we propose an evolutionary meta-heuristic to solve the nurse rerostering problem. We show that the proposed procedure performs consistently well under many different circumstances. We test different optimisation strategies and compare our procedure with the existing literature on a dataset that is carefully designed in a controlled and varied way.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

Operating systems typically have to operate in a dynamic and uncertain environment where unexpected events may occur. When these events lead to schedule disruptions and infeasibilities, rescheduling is necessary to update the activity schedule. A disruption in a personnel schedule is defined as an occurrence when an employee cannot be planned to work a specific task is unavailable to work the specified task due to e.g. unplanned absences or staff turnover.

In a report of the National Health Service (UK, 1999) it is indicated that in the health sector, disruptions have a dramatic effect on the budget for workforce staffing. In total 4% of the total resources spent on staffing are lost due to schedule disruptions [1]. Since uncertainty is considerable in healthcare applications and disruptions cannot be eliminated, decision support systems should be developed that adequately react to unexpected events. The rerostering problem is a scheduling type of problem with which most hospitals are confronted. In case of schedule disruptions, hospital units should resort to their own nursing resources to solve any schedule infeasibility. Only when the hospital unit cannot cope with the problem(s), they can typically call upon more expensive nursing staff resources external to the unit (e.g. hospital reserve pool, external interim nurses).

When contingencies and schedule disruptions occur, employees cannot perform their duties in accordance with the postulated

schedule. In that case, deviations are required to the original schedule as tasks cannot be operated below a minimum number of required staff. Rerostering nursing staff is a reactive approach to cope with unexpected schedule disruptions in personnel rosters such that the schedule remains valid in accordance with the schedule requirements, i.e. the staffing requirements and the time-related constraints which guarantee the roster quality of the single personnel members. Typically, the goal of rerostering is to rebuild the schedule while minimising the number of deviations to the original schedule as deviations may not be very well accepted by the workforce.

Due to the frequency of schedule disruptions and the importance of rerostering in personnel rostering, we propose an optimisation tool for the nurse rerostering problem that revises and re-optimises a schedule for a set of heterogeneous nurses. In this paper, we gain insights and understanding in solving the nurse rerostering problem. We focus on the application of optimisation concepts that were found to be successful for the traditional nurse rostering problem and adapt these to meet the peculiarities (problem settings, objectives and constraints) of the rerostering problem. The personnel scheduling optimisation methods found in the literature provide excellent solutions to the nurse rostering problem. However, as uncertainty and absenteeism are inherent when dealing with personnel, these roster solutions may no longer be valid at certain moments. We discuss how to tailor these methods to the nurse rerostering problem from an algorithmic point-of-view.

The remainder of the paper is organised as follows. In Section 2, we give an overview of the relevant literature on heuristic optimisation procedures involving nurse (re-)rostering problems. A precise problem description and formulation is provided in Section 3, in which we discuss the typical settings, objectives and

* Corresponding author.

E-mail addresses: Broos.Maenhout@UGent.be (B. Maenhout), Mario.Vanhoucke@ugent.be (M. Vanhoucke).

constraints of the problem. In Section 4, we discuss the proposed heuristic approach to solve the nurse rostering problem. In Section 5, we discuss the algorithmic performance of all problem-specific features of our approach and compare the proposed optimisation procedure with the existing literature. In Section 6, conclusions are drawn and directions for future research are given.

2. Literature review

Most research papers in the personnel scheduling literature involve exact or heuristic procedures for the construction of a deterministic personnel schedule. An extensive overview of personnel shift scheduling problems can be found in Ernst et al. [2,3] and for the nurse rostering problem specifically in Cheang et al. [4] and Burke et al. [5]. In the following, we first go through the nurse rostering literature and then we discuss the relevant (meta-)heuristic optimisation principles found in the general nurse rostering literature.

In contrast to the nurse rostering problem, the *nurse rerostering problem* has received limited attention in the staff scheduling literature. Moz and Pato [6] were the first to formulate the nurse rerostering problem as an integer multi-commodity flow model with the single objective of minimising the number of deviations from the original schedule. This single-objective version has been classified as NP-hard in Moz and Pato [7]. Moz and Pato [6] developed a constructive heuristic to reconstruct nurse rosters instantly. Furthermore, they developed various mathematical ILP formulations with additional constraints for the nurse rerostering problem in a multi-level acyclical network that are solved using CPLEX [6,8]. Recently, Moz and Pato [7] developed an indirect genetic algorithm approach to the nurse rerostering problem and tested their procedure on real-life data instances from Portuguese public hospitals. Their algorithm is composed of an improved constructive heuristic and a genetic algorithm. The constructive heuristic entails a sequential re-assignment of all tasks to nurses in various ways after a disruption has occurred. In the genetic algorithm, the encoding consists of two permutations, i.e. a permutation of the list of tasks and a permutation of the list of nurses. A new nurse roster is then generated by trying to assign each task to one of the nurses following the order of the nurse permutation and based on various rules of thumb which promote the feasibility and the quality of the nurse roster as best as possible. The individuals are then scored based on their similarity to the original schedule. A genetic algorithm (using selection, crossover and mutation operators) is then applied to the population of individuals to produce a new population. The authors developed several versions of the genetic algorithm, whose differences lay in the encoding of the permutations and in the genetic operators used for each encoding. Pato and Moz [9] formulated and solved a multiple objective approach to the nurse rerostering problem in order to map the trade-off between schedule similarity and the fairness among nurses. In their paper, they designed a bi-objective genetic heuristic based on the best performing optimisation principles found in Moz and Pato [7]. In Pato and Moz [9], however, the evolutionary approach operates on the Pareto optimal set of solution elements and the fitness function complies with the Pareto ranking of the respective decoded solutions. Knighton [10] proposed a network-based mathematical programming approach that also incorporates multiple objectives. The goal of this methodology is to respond to disruptions in a workforce schedule such that the number of deviations with respect to the original schedule is minimised and the management and employee preferences to work a particular shift on a particular day are satisfied as best as possible. In this respect, a decomposition-based heuristic algorithm is developed

for the rerostering of a heterogeneous workforce over a multi-week period. A schedule is composed of combining multiple weekly nurse rostering problems that are constructed using a network-based linear program.

Related to the nurse rostering problem, which is a shift scheduling problem, there are different other papers in the literature that cope with reconstructing personnel schedules e.g. the task-based crew recovery problem in the airline industry (for an overview see Clausen et al. [11]), the tour-of-duty rescheduling problem in the railway sector (e.g. Huisman [12]), the general days on/days off personnel replanning problem under annualised hours [13].

In the *nurse rostering problem* literature, problem descriptions and models vary drastically and depend on the characteristics and policies of the particular business environment. Hence, many objective function possibilities subject to a huge variety of constraint combinations are explored. Since nurse rostering problems have this multitude of formulations, many (meta-)heuristic procedures have been proposed to solve the nurse rostering problem in an acceptable computation time. Instigated by their flexible design, meta-heuristic approaches are considered to tackle most appropriately the nurse rostering problem in a real-world problem setting. Recent meta-heuristics described in the literature implemented successfully the ideas of creating and maintaining diversity in the population elements and the idea of exploiting variable neighbourhoods to diversify the search in the solution space. We can discern several local search-based and evolutionary meta-heuristic frameworks that are successfully implemented to solve the nurse rostering problem. In this respect, variable neighbourhood search is an important concept in heuristic optimisation that is successfully explored in recent years for the nurse rostering problem and boils down to the systematic change of neighbourhoods within a (local) search algorithm. Different papers combine different local search heuristics within a meta-heuristic framework and/or study the use of hyperheuristics that implement neighbourhoods to enhance the diversification step in the meta-heuristic search. We explore these principles in this paper to tackle the nurse rerostering problem adequately.

3. Problem definition and formulation

The problem under study is the nurse rerostering problem and can be stated as follows. A set of heterogeneous nurses N (i.e. set of nurses, index i ($i=1, \dots, n$)) is scheduled on a periodically basis within a pre-defined period D (i.e. set of days in the scheduling period, index j ($j=1, \dots, d$)). More precisely, these nurses are assigned to one of a set of possible shifts S (i.e. set of shifts, index k ($k=1, \dots, s$), with the last shift s as the free day assignment). This assignment should satisfy on the one hand some (hard) time-related constraints that typically guarantee the (social) quality of a schedule for a single nurse. On the other hand, multiple (conflicting) goals can be postulated, e.g. minimising understaffing and overstaffing costs, minimising labour costs, maximising the nurse preferences and constructing employee schedules as fair as possible. The decision variables used for the nurse rerostering problem are the following:

$$x_{ijk} \quad \begin{cases} 1 & \text{if nurse } i \text{ is scheduled to work on day } j \text{ shift } k, \\ 0 & \text{otherwise.} \end{cases}$$

Combining the decisions for all these assignment variables, each employee of the available nursing staff is assigned to an individual schedule, which is in line with the applicable nurse rostering policies and objectives. When coping with the nurse rerostering problem, the nurse rostering problem has already been solved. We denote these assignments stated in the original nurse

roster as

x'_{ijk} 1 if nurse i was scheduled to work on day j shift k in the original nurse roster,
0 otherwise.

The problem of rerostering nurse schedules arises when a set of schedule disruptions V occurs which is defined as follows:

$$V = \{(i, j, k) : x_{ijk} = 0 | x'_{ijk} = 1, \forall i \in N, \forall j \in D, \forall k \in S \setminus \{s\}\}.$$

In other words, V is the set of all schedule disruptions for which nurse i is unable to perform the originally assigned tasks k on one or more future work days j for unexpected reasons. Hence, the nurse rerostering problem involves rebuilding duty timetables by assigning a set of heterogeneous nurses N to one of a set of shifts S over a mid-term or short-term period of D days. Fig. 1 is an example nurse roster. The left roster displays the original nurse assignments of five nurses over a period of four days. Each day the nurses are assigned to the early (E, 7am–3pm), late (L, 2pm–10pm), night (N, 9pm–7am) or free (F) shift. The crossed assignments designate the disruptions in this left nurse roster. In this example, the schedule assignments for nurse 2 on day 3 and for nurse 5 on day 2 are disrupted. Similar to the traditional nurse rostering problem, the required reconstruction of nurse rosters is undertaken along with various constraints and objectives, which are described in Sections 3.1 and 3.2 respectively. The right roster in Fig. 1 designates a feasible solution and solves these two conflicts. The shaded assignments are the assignments that are adapted compared to the original roster in order to attain feasibility. The period to reroster, which is typically to be decided by the management, is the complete planning horizon in this example. In Moz and Pato [6], the period to reroster is determined to endure from the first day of changes until the end of the planning horizon.

3.1. Scheduling constraints and requirements

Staffing requirements: One of the pre-requisites in nurse rerostering is that the set of duties associated with the set of disruptions V are disrupted and must be performed by other nurses such that the staffing requirements are continued to be met, i.e.

$$\sum_{i \in N} x_{ijk} - n_{jk}^o + n_{jk}^u = R_{jk} \quad \forall j \in D, \forall k \in S \setminus \{s\} \quad (1)$$

with R_{jk} is the required number of nurses to work shift k on day j , n_{jk}^o is the surplus number of nurses scheduled to work shift k on day j , n_{jk}^u the deficient number of nurses scheduled to work shift k on day j .

These coverage requirements express the required number of nurses R_{jk} per shift and per day and are inherent to any shift scheduling problem.

Time-related constraints: The new roster should meet not only these staffing requirements but should also be conform to all established time-related requirements as for the original nurse rostering problem, i.e. national legislation, institutional conditions (collective union agreement, hospital regulations and policies), unit requirements and personal contract stipulations. These rules define acceptable schedules for the individual nurses and reduce the set of feasible individual roster lines.

The first two constraints are very typical and inherent to continuous shift scheduling problems:

- Each nurse can only be assigned to one working shift per day or to the free shift.
- Forward rotation: A minimum rest time should be respected between working shifts that prohibits the succession of certain shifts. This constraint implies the forbidden successive assignments between e.g. a night and early/late shift and between a late and early shift on the next day.

Additionally, apart from these two constraint types, Burke et al. [5,14] identified different other constraint types and their appearance in literature. In search for a general and robust rostering model, we incorporated the following four classes of frequently occurring constraints both with a minimal and maximal constraint:

- Number of working assignments, i.e. number of non-free shift assignments per scheduling period.
- Number of consecutive working assignments, i.e. number of consecutive non-free shift assignments.
- Number of assignments per shift type, i.e. number of identical shift assignments per scheduling period.
- Number of consecutive assignments per shift type, i.e. number of consecutive identical shift assignments.

Disruption constraints: Besides all these constraints of the traditional nurse rostering problem that must be satisfied, a new constraint is imposed on the nurse rerostering problem that stipulates that some nurses cannot be assigned to particular shifts due to the encountered unplanned absences:

$$x_{ijk} = 0 \quad \forall (i, j, k) \in V \quad (2)$$

In this paper, the time-related rules are set as hard constraints as done by Moz and Pato [6]. The staffing requirements and the prohibitive assignment constraint due to schedule disruptions are dealt with as soft constraints that can be violated at a certain penalty cost expressed in the objective function using the auxiliary variables n_{jk}^o and n_{jk}^u (cf. goal Z_1) and parameter Q_{ijk}^1 (cf. goal Z_2) respectively.

	Day 1	Day 2	Day 3	Day 4
Nurse 1	E	E	N	N
Nurse 2	L	L	L	L
Nurse 3	N	N	F	F
Nurse 4	F	F	E	E
Nurse 5	E	E	E	E
$\sum_i x'_{ijE}$	2	2	2	2
$\sum_i x'_{ijL}$	1	1	1	1
$\sum_i x'_{ijN}$	1	1	1	1

	Day 1	Day 2	Day 3	Day 4
Nurse 1	E	E	L	L
Nurse 2	L	L	F	F
Nurse 3	N	N	N	N
Nurse 4	E	E	E	E
Nurse 5	F	F	E	E
$\sum_i x_{ijE}$	2	2	2	2
$\sum_i x_{ijL}$	1	1	1	1
$\sum_i x_{ijN}$	1	1	1	1

Fig. 1. An example nurse roster with schedule disruptions (left) and a possible feasible solution (right).

3.2. Objectives

Apart from complying with all these hard constraints, the nurses should be assigned to shifts such that the quality of the reconstructed timetable is maximised. The schedule quality is measured by multiple objectives with different priority levels which represent the hospital's policies and the nurses' preferences.

Service continuity and overtime: The hospital objectives typically consist of ensuring a minimum level of care in terms of the number of nurses for each shift, while avoiding additional costs as a result of overstaffing and/or unplanned absences, i.e. overtime, inefficient staffing, the use of external or temporary nurses, etc. [15,16]. These objectives are strived for by minimising respectively the shortage and surplus slack variables of constraint (1), i.e.

$$\text{Min } Z_1 = \sum_{j \in D} \sum_{k \in S \setminus \{s\}} (C_{jk}^o n_{jk}^o + C_{jk}^u n_{jk}^u) \quad (3)$$

with C_{jk}^o is the penalty cost of scheduling a surplus number of nurses to work shift k on day j , C_{jk}^u is the penalty cost of scheduling a deficient number of nurses to work shift k on day j .

Nurse preferences and infeasibilities due to schedule disruptions: The quality of a personnel roster is typically measured in the way the schedule satisfies the individual nurse preferences [3]. Azaiez and Al Sharif [15] conducted an extensive survey to gain understanding on real-world nurses' preferences encountered in traditional nurse rostering. All of the identified kinds of preferences can be modelled by defining nurse preferences of working a particular shift on a particular day and sequence dependent preferences (e.g. involving the number of consecutive working days). When rostering nursing staff, a survey by Moz and Pato [6] revealed that the nurse preferences consist primarily of retaining the current nurses individual shift assignments as much as possible. Other nurse preferences for nurse rostering problem are the shift assignments nurses cannot be assigned due to schedule disruptions (i.e. the disruption constraints, which are handled as soft constraints using the parameter Q_{ijk}^1). Both kinds of preferences are expressed as follows:

$$\text{Min } Z_2 = \sum_{i \in N} \sum_{j \in D} \sum_{k \in S} (p_{ijk} x_{ijk}) \quad (4)$$

with p_{ijk} is the preference penalty cost of scheduling nurse i to work shift k on day j with $p_{ijk} = p'_{ijk} + Q_{ijk}^1 + Q_{ijk}^2$, with $Q_{ijk}^1 > 0$ if $(i, j, k) \in V$ and $Q_{ijk}^1 = 0$ otherwise $Q_{ijk}^2 > 0$ if $x_{ijk} \neq x'_{ijk}, \forall (i, j, k) \in (N \times D \times S) \setminus V$ and $Q_{ijk}^2 = 0$ otherwise.

The preference penalty costs for the nurse rostering problem are composed of the sum of three penalty costs, i.e. p'_{ijk} , Q_{ijk}^1 and Q_{ijk}^2 . p'_{ijk} indicates the preference penalty cost of scheduling nurse i to work on day j shift k as taken into account for the original nurse roster construction. The penalty cost Q_{ijk}^1 is associated with the set of schedule disruptions V ($Q_{ijk}^1 > 0, \forall (i, j, k) \in V$). The penalty cost Q_{ijk}^2 is related to nurse assignments that deviate from the original roster ($Q_{ijk}^2 > 0$ if $x_{ijk} \neq x'_{ijk}, \forall (i, j, k) \in (N \times D \times S) \setminus V$). Hence, this objective boils down to determine a new workable roster that minimises the number of shift changes with regard to the current one or to find a new feasible roster as similar as possible to the previously announced roster for the same period.

Fairness: In nurse rostering and, hence, in nurse rostering it is important to evenly distribute the workload over the nurses. This fairness objective assures that nurses, whose schedule needs to be changed due to unplanned absences, have to catch up with these unperformed duties such that the fairness is maintained (or even improved) after reconstructing the nurse roster. This objective is formulated as follows:

$$\text{Min } Z_3 = \sum_{i \in N} (\delta_i^- + \delta_i^+) \quad (5)$$

s.t.

$$\sum_{j \in D} \sum_{k \in S \setminus \{s\}} x_{ijk} - \delta_i^+ + \delta_i^- = \bar{a} \quad \forall i \in N \quad (6)$$

with \bar{a} is the number of duties averaged over the nurses ($= \text{minimum}(\sum_{j \in D} \sum_{k \in S \setminus \{s\}} R_{jk} / n, a^{\max})$ with a^{\max} the maximum number of duties over a period D).

This objective ensures that there is a buffer between the number of scheduled duties and the maximum number of duties that can be performed during the planning horizon for each nurse.

4. A heuristic solution procedure for the nurse rostering problem

In this section, we propose a heuristic procedure for the nurse rostering problem. As discussed in Section 2, many successful exact and heuristic procedures were developed for the traditional nurse rostering problem that generate a workable baseline schedule, which assumes complete information and a static and deterministic environment. In this paper, we focus on the application of established techniques that have proven to be successful for the nurse rostering problem and adapt these to meet the specific peculiarities of the nurse rostering problem. In the following, we discuss an evolutionary algorithm and consider how we deal with nurse rostering from an algorithmic point-of-view as we incorporate problem-specific information.

4.1. Basic structure

Evolutionary algorithms (EAs) are population-based meta-heuristics imitating the evolutionary ideas of natural selection and genetic processes. The basic concept of EAs consists in simulating evolutionary processes by combining solutions to yield better solutions thriving on the principle of the survival of the fittest. As such, they represent an intelligent exploitation of a random search within a defined search space to solve the problem under study. Meta-heuristic optimisation methods and, hence, evolutionary algorithms provide only a conceptual optimisation framework that is very flexible, since each of its elements can be implemented in a variety of ways and degrees of sophistication. Their effectivity and flexibility have inspired many researchers to use EAs for solving various (real-world) scheduling problems in many application areas. The application and success of the exploitation of EAs, however, depend on the incorporation of problem-specific information. The basic structure of our algorithm is as follows:

```

Initialisation procedure
While stop criterion not met
    Chromosome selection
    Combination mechanism
    Mutation operator
    Improvement method
    Update population
Endwhile

```

In the following, we highlight how problem-specific information of the nurse rostering problem is incorporated.

4.2. Representation of a chromosome

In evolutionary heuristics, each generated individual solution is encoded in a data structure, called a chromosome. In contrast to the bipartite chromosome structure used by Moz and Pato [7], we utilise the nurse–day view [4] as representation form, which is a

common notation for solving the traditional nurse rostering problem. The nurse–day view is a direct representation of the two-dimensional nurse duty roster and directly reflects respectively the shifts or tasks the nurse is assigned to over the scheduling period. These chromosomes serve as input for a fitness function which evaluates the quality of the solution encoded in each chromosome (cf. Section 3).

4.3. Initialisation procedure

The evolutionary algorithm starts with the initialisation of a population of individuals. Since useful information about the structure of optimal solutions is typically contained in a suitably diverse collection of elite solutions, the initial population is generated and constructed in such a manner a critical level of diversity is guaranteed [17,18]. Based on this consideration, we construct solution elements in two different ways, i.e. in a completely random manner and using a constructive heuristic. The constructive heuristic, able to find high-quality solution early in the search process, schedules the nurses nurse per nurse in a random sequence taking the different objective function components into account. Based on this initial set of solution elements, only a subset is designated to be population elements. In order to promote the concept of diversity even more, we store only the non-dominated Pareto optimal solutions in the population. Due to the multi-objective nature of the nurse rerostering problem, we do not retain the set of best solutions based on an aggregation of the different goals but we apply the principle of Pareto optimality and maintain all solution elements that are not dominated in one of the dimensions of the objective function. In Fig. 2, the principle of dominance is exemplified in the objective space for a minimisation problem of only two different objectives. The solution points on the dashed line are Pareto optimal whereas the solution points above the line are dominated solution points. After the initialisation phase, the evolutionary process of selection, combination (or diversification), mutation, intensification and updating the population set is instigated. Guided by the principles in these mechanisms, the population evolves over time admitting new solution points and ruling out old solutions. This process improves the quality of the best known solution elements during the search process. The search heuristic is performed until a maximum number of schedules is evaluated.

4.4. Chromosome selection

In each evolution cycle, pairs of chromosomes are randomly selected and exchange information in such a way a new individual is created with attributes of both the parent chromosomes. The evolutionary algorithm operates on this reference set by combining pairs of reference solutions in a controlled and structured way. Based on the ranking of the population elements according to the different objective function criteria, we randomly

select for each ranking one solution element. This selection is guided by probabilities that are proportional to the fitness of the elements with respect to the objective function component the ranking is based on. When, due to randomness, two elements are chosen that are highly ranked for one objective function component, intensification is stimulated. Diversification will be stimulated in case the chosen elements score high on different objective function components. Moreover, a distance threshold is imposed which avoids that highly resembling solution elements are paired. There are as many solution elements chosen as there are different objective function components. All two-element subsets are investigated out of this set of elements.

4.5. Diversification

Combination mechanism: Depending on the data encoding, different crossover operators have been applied in the literature concerning the nurse rostering problem. Using the day-based view most crossovers define the crossover point based on the nurses (i.e. whole individual nurse schedules are copied to the child chromosomes) [19–22] or based on the days (i.e. whole day rosters are copied to the child chromosomes) [22,23]. The first crossover operators are characterised by the fact that the child rosters encounter feasibility problems with respect to the coverage constraints. The latter are confronted with infeasibilities with respect to the individual time-related constraints as individual nurse schedules are broken into pieces. Burke et al. [14] define another type of crossover operator that combines the parent chromosomes by breaking these rosters in both dimensions, i.e. the nurse and day-dimension, in order to incorporate the best elements of both rosters. This crossover is confronted with infeasibilities in both dimensions. In this paper, we propose a crossover mechanism that guarantees the feasibility with respect to the time-related nurse constraints and tries to obtain coverage feasibility as best as possible.

In this combination method new solution points are the result of combining sets of two rosters out of the Pareto optimal reference set. This process is characterised as generating paths between an initiating and a guiding solution [24]. A path between solutions will generally yield new solutions that contain characteristics of both the parent solutions. The crossover mechanism introduces attributes contributed by the guiding solution. The goal is to capture the assignments that frequently or significantly occur in high-quality solutions and then to introduce some of these compatible assignments into solutions that are generated by the heuristic combination mechanism. The relinking process between the two parent solutions out of the reference set is based on moves in the nurse neighbourhood.

For these moves in the nurse neighbourhood, the schedule of a subset of the nurses of the initiating solution is directed towards the schedules of the corresponding nurses in the guiding solution nurse per nurse. This is accomplished by defining a minimum cost flow problem for each nurse with an underlying activity-on-the-node network structure as in Maenhout and Vanhoucke [25]. This problem of finding a feasible individual nurse schedule with minimum cost is a resource constrained shortest path problem. In order to calculate the costs of the assignments in the network, the proposed combination method relies on problem-specific information of both the initiating and the guiding solution element to create a new roster and takes multiple criteria into account. The first three of these criteria incorporate objective function related data as follows:

- Vertical roster quality (Z_1)—In order to minimise the penalty cost associated with the violation of the minimal staffing

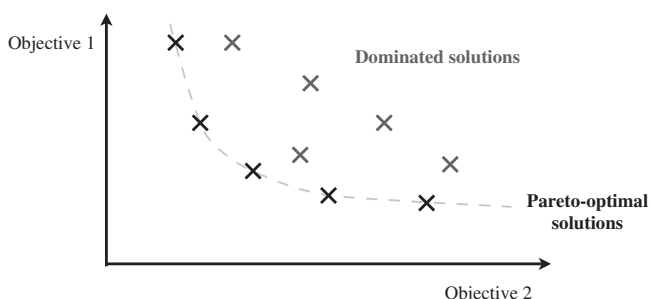


Fig. 2. The principle of Pareto optimality.

requirements, the algorithm penalises (only) those shifts where the staffing requirements are violated. In doing so, the solution combination method biases the initiating solution towards a (more) feasible solution.

- Horizontal roster quality: Nurse preferences (Z_2)—This component involves the objective to satisfy the nurses preferences and to solve infeasibilities due to schedule disruptions as best as possible.
- Horizontal roster quality: Nurse workload (Z_3)—This component involves the objective to distribute the workload as fair as possible over the nurses.

The other two criteria incorporate information to maintain the good characteristics of both the initiating and the guiding roster as follows:

- Criticality of shifts—In directing the initiating solution towards the guiding solution, the algorithm prevents the removal of critical shifts from the initiating solution, which, in case of removal, would lead to an additional violation of the staffing requirements. In doing so, the algorithm aims at the construction of a new solution point that does not encounter any (additional) violations of the staffing requirements.
- Bias to the guiding solution—The algorithm guides the initiating solution to the assignments of the guiding solution, in order to insert attributes of the guiding solution into the initiating solution.

These elements are carefully taken into account for each move combining an initiating solution and a guiding solution. The proposed heuristic combination mechanism thrives on the idea that the parent chromosomes will pass their good characteristics on to the newly created solution points. Hence, the algorithm preserves or even improves the good characteristics of the parent chromosomes.

The subset of nurses that is subject to the combination method consists primarily of the nurses encountering schedule disruptions. The individual schedules of other nurses are combined only if the individual nurse schedule in the guiding solution has a significantly better objective function quality than the corresponding individual nurse schedule in the initiating solution. The latter introduces some controllable flexibility on the size of the roster that is subject to the combination mechanism. This implementation restricts the roster size that is subject to diversification. Guided by the five different cost criteria, the combination method is conceived as such that the number of schedule changes to non-disrupted nurse schedules is as limited as possible and that these changes are only prompted by objective function considerations. Fig. 3 displays the crossover operator for a single nurse illustrating how the construction of the child solution is influenced by the different cost criteria. The non-shaded assignments are assignments different than the initiating and guiding solution and are introduced by the three objective function-related criteria. The shaded assignments are retained from the initiating solution or introduced by the guiding solution.

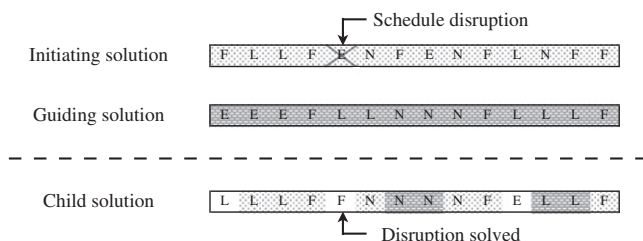


Fig. 3. Illustration of the crossover operator for the nurse rostering problem.

Mutation operator: Each new chromosome can undergo another transformation by assigning random schedules to nurses, called a mutation, to prevent convergence during the search process. This is carried out by defining a minimum cost flow problem with an underlying network structure as for the combination method with random costs.

4.6. Improvement method

The applied local search heuristics explore neighbouring solution points of the newly generated solution point in order to improve the different objective function components. To that purpose, we introduced a variable neighbourhood search based on the principles described in Maenhout and Vanhoucke [26] and adapted these to the specific problem characteristics of the nurse rostering problem. The three local search algorithms are sequentially solved and decompose the original problem into smaller problems that are optimally solved. They can be perceived as complementary as each local search concentrates on a different part of the scheduling matrix, i.e. on the shift-pattern of a single nurse, on a single day and on the complete matrix. Due to the large size of the neighbourhood, the neighbourhoods are searched in an efficient manner using network flow techniques and dynamic programming. The neighbourhoods of the different local search mechanisms are illustrated in Fig. 5.

Compared to the traditional nurse rostering problem, the performance of these local search methods is heavily dependent on the order according to which the specific roster areas (i.e. nurse schedule, day roster or combination of days) are optimised. For each of the local searches, we propose a dynamic guiding order in which the associated roster area to select next is dependent from the roster at hand and its impact on the entire nurse roster, which is determined by an improvement cost factor. This approach is beneficial for the computational performance (cf. Section 5.2) as only a subpart of the roster and associated search space needs to be investigated to attain the local optimum in each iteration. Fig. 4 displays a schematic overview of the generic functioning of the applied local search methods.

In order to select the next roster area to optimise, we calculate each time the improvement costs for the remaining non-optimised roster areas. The method of this cost calculation is dependent on the local search at hand (cf. infra). The roster areas that are already optimised in the current iteration of a specific

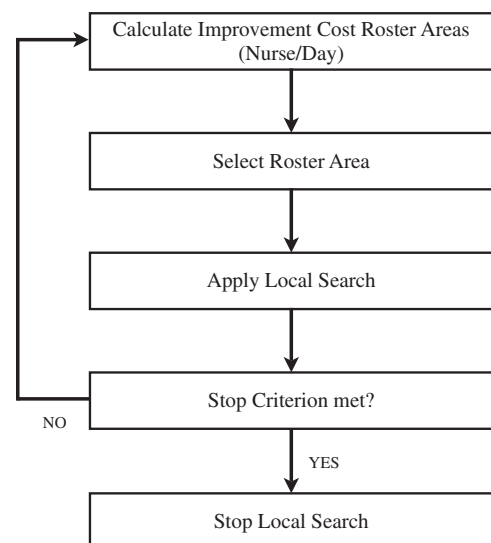


Fig. 4. Schematic overview of the local search heuristics.

local search run are not considered when determining the next roster area to optimise. Based on these calculations, we select the roster area that has the worst quality or that is thought to have the highest potential to improve and apply the improvement method. This process is continued until the stop criterion is met. The stop criterion is twofold. Firstly, instigated by the objective function structure it is guaranteed that the roster areas that contain disruptions (i.e. the shaded areas in Fig. 5) or that have an improvement cost factor larger than a certain threshold value are optimised. Secondly, controlled by a percentage parameter, only a subset of the other roster areas is further subject to the local search whereas for the traditional nurse rostering problem it is required that almost all roster areas are examined.

Pattern-based local search: The pattern-based local search optimises the schedule of a particular single nurse given the (fixed) schedules of all other nurses. In order to determine the next individual nurse schedule to optimise the quality of the remaining individual nurse schedules is calculated as follows:

- Vertical roster quality: Overstaffing (Z_1)—We account a penalty cost to the shift assignments of the nurse that is overstaffed.
- Horizontal roster quality: Nurse preferences (Z_2)—We account a penalty cost for the deviations of the current nurse schedules from the original ones.
- Infeasibility: Nurse preferences (Z_2)—We account an infeasibility penalty cost (a very large number that is proportional to the encountered number of infeasibilities) to the individual schedules that are infeasible due to unplanned absences.
- Horizontal roster quality: Nurse workload (Z_3)—Nurse schedules for which the workload deviates from the average workload are accounted for a penalty cost proportional with this deviation.

These costs and the total cost are calculated for each nurse. A minimum cost flow problem is defined in the same way as done for the combination method (cf. Section 4.5). The associated costs with the node assignments take all the objective function components into account (Z_1 , Z_2 and Z_3).

Day-based local search: The day-based local search optimises a single day of the nurse roster given the (fixed) assignments of the nurses on all other days. This decomposition heuristic optimises the nurse roster day-by-day by optimally assigning a feasible shift assignment to each nurse. This local search is formulated as such that under- and overstaffing of the staffing requirements are allowed. The costs in the assignment cost matrix consist of the costs associated with the entire schedule the nurse is assigned to taking the different objective function components into account (Z_1 , Z_2 and Z_3). In order to select the day to optimise next, the improvement cost of the remaining non-optimised days is calculated taking the following criteria into account:

- Vertical roster quality (Z_1)—We account a penalty cost to the shifts that are over- or understaffed.

- Horizontal roster quality: Nurse preferences (Z_2)—We account a penalty cost for the number of deviations of the current day rosters compared to the original day rosters.
- Infeasibility: Nurse preferences (Z_2)—We account an infeasibility penalty cost to the day rosters that embody an infeasible shift assignment due to schedule disruptions.
- Horizontal roster quality: Nurse workload (Z_3)—We compare the total workload of each nurse with the average workload. We add only a penalty cost to the quality of a certain day when there is potential to improve this objective:
 - if $\delta_i^- > 0$, penalty costs are assigned to all days nurse i is assigned a day off.
 - if $\delta_i^- = \delta_i^+ = 0$, no penalty costs are assigned.
 - if $\delta_i^+ > 0$, penalty costs are assigned to all days nurse i is assigned a working shift.

Schedule-based local search: The schedule-based local search focuses on the whole schedule for all nurses by swapping (sub-parts of) schedules between nurses. To that purpose, we define a linear assignment problem that optimally redistributes (sub-parts of) the shift patterns of the current population element among the nurses. This redistribution mechanism improves the objectives Z_2 and Z_3 , while Z_1 cannot be improved. In order to determine the combination of days to optimise next, we sum the improvement cost for different days that is calculated likewise for the day-based local search except for the vertical roster quality component (Z_1) which is left out the calculation.

4.7. Update population

After the application of the diversification and intensification process, the population is maintained and dynamically updated according to the principle of Pareto optimality. Child solutions are only added if the solution point is not dominated regarding one of the objective function components and the distance between the new solution and any solution of the population is larger than some threshold value. The latter prevents the duplication of solution points in the reference set and/or the entrance of highly resembling solutions.

5. Computational experiments

In this section, we provide computational insights into our procedure that is developed for solving the nurse rostering problem. In Section 5.1, we describe the test design and parameter settings used in the experimental analysis. In Section 5.1.1 we explain the methodology based on which the new benchmark dataset is constructed. We explain in detail how the input data for the nurse rostering problem, i.e. the original nurse roster instances and the schedule disruptions, are generated. Further, we explain the parameter settings and weights of all objectives and constraints of the nurse rostering problem. In Section 5.1.2 we discuss how the test framework and parameters of the algorithmic components of the procedure are finetuned to obtain satisfactory

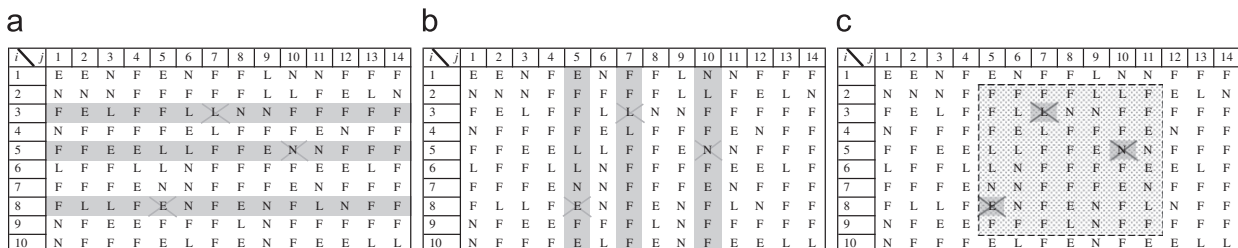


Fig. 5. Overview of the three complementary local search heuristics: (a) Pattern-based local search; (b) day-based local search; (c) schedule-based local search.

results. In Section 5.2, we validate the beneficial performance of the proposed problem-specific operators for the nurse rostering problem. In Section 5.3, we compare the performance of the proposed procedure with the existing literature. All tests were carried out on a Dell Dual Core processor 2.8 GHz and 2 Gb RAM.

5.1. Test design

5.1.1. Test dataset: problem instances and parameter settings

Original nurse roster instances: In order to test the performance of our newly developed evolutionary algorithm rigorously, we compose a dataset based on artificially generated complexity not only with respect to the schedule disruptions but also with respect to the original nurse rosters. It is our goal to test our algorithm under a wide range of varying assumptions in order to verify the robustness of the proposed solution approach. To that purpose, we downloaded the nurse rostering problem instances generated by NSPGen [27] from www.projectmanagement.ugent.be/nsp.php. These files have been artificially constructed as the nurse preferences and coverage requirement information have been generated under a controlled design based on six complexity indicators presented in Vanhoucke and Maenhout [27]. This dataset contains different subsets with different problem sizes and characteristics. We have tested our procedure on the subset with 30 full-time nurses of the so-called realistic set where the nurses have to be assigned to one of the four shifts (i.e. the early, late, night or free shift) over a planning horizon of 28 days. All the involved nurses have the same contract stipulations and skill competences. This implies perfect substitutability between the nurses. Based on different values for the complexity indicators, this N30 subset consists of in total 96 complexity classes with each containing 10 instances. We have randomly selected one instance from each complexity class which results in 96 problem instances. Further, each problem instance has been extended by a set of different (hard) time-related constraints (cf. Section 3.1) apart from the staffing requirements per day and per shift, i.e.

- (i) Number of working assignments: min 10, max 20.
- (ii) Number of consecutive working assignments: min 2, max 5.
- (iii) Number of assignments per shift type: min 0, max 20.
- (iv) Number of consecutive assignments per shift type: min 1, max 5.
- (v) Each nurse can only be assigned to one working shift per day or to the free shift.
- (vi) Forward rotation: A minimum rest time of 11 h should be respected between working shifts (e.g. a late shift cannot be followed by an early shift).

We have solved these 96 problem instances to optimality using the branch-and-price procedure of Maenhout and Vanhoucke [25]. The incorporated nurse scheduling objectives are the following (ordered according to their degree of importance):

- Minimise the number of shifts that are understaffed (Z_1).
- Minimise the number of shifts for which too many nurses are assigned to (Z_1).
- Construct a roster as fair as possible with respect to the workload (i.e. the number of assigned shifts) (Z_3).
- Satisfying the nurses' wishes and requests as much as possible (Z_2).

In this way we constructed an artificial set of original nurse rosters founded on different complexity characteristics for the nurse rostering problem.

Schedule disruption characteristics: Based on this set we constructed a problem set for the nurse rostering problem by specifying the occurrence of disruptions in a systematically varied

and controlled way. The number of schedule disruptions on day j ($j \in D$) is expressed as v_j . We use $\bar{v} = \sum_{j \in D} v_j / |D|$ to denote the average number of schedule disruptions per day. The occurrence of schedule disruptions for the nurse rostering problem is generated by means of two input parameters, i.e.

- Total number of disruptions (TND): This factor gives an indication of the number of unplanned schedule absences over the complete scheduling horizon as follows:
 - If TND = 1, the number of disruptions on each day j and each shift k is equal to the staffing requirements R_{jk} which signifies that all scheduled duties are disrupted.
 - If TND = 0, no scheduled duties are disrupted.

This measure is expressed as a fraction of the required duties over the planning horizon ($= \sum_{j \in D} v_j / \sum_{j \in D} \sum_{k \in S} R_{jk}$).

- Spread of disruptions (SD): This factor measures the distribution of the number of disruptions over all days as follows:
 - If SD = 0, the number of disruptions are evenly distributed over the days of the planning horizon.
 - If SD = 1, the number of schedule disruptions are maximal for one or several days, and zero for all remaining days.

The maximum number of schedule disruptions that can occur for a single day is equal to the number of required duties that day. In order to define the spread of the disruptions, we use a general measure of variance ($SD = \sum_{j \in D} |v_j - \bar{v}| / \alpha_{max}$ with α_{max} as the maximum possible deviation). This is a variant of the variance measure that has been proposed by Vanhoucke et al. [28].

Table 1 indicates the test settings we employed to construct our set of problem instances for the computational analysis. We utilise a 3×3 factorial design to generate schedule disruptions in order to investigate the performance of the algorithm in the most robust way possible. This implies that we have a problem set of $96 \times 9 = 864$ problem instances in total. We have opted to keep the number of disruptions very low in comparison with the total number of required duties in line with real-life situations (note that the traditional nurse rostering problem is solved when the number of disruptions becomes too large). The spread, however, is varied as large as possible.

Objective function weights and constraint parameters: Based on these original nurse rosters and generated schedule disruptions a 4-week nurse roster is re-constructed from the day of the first disruptions to the end of the planning horizon in line with Moz and Pato [6]. Additional constraints for the nurse rostering problem, apart from constraints (i)–(vi), that are imposed in our test design are the following disruption constraints (cf. Section 3.1):

- (vii) nurses may not be assigned to work tasks on the days they are absent due to e.g. vacation;
- (viii) nurses may not be assigned to work tasks on the days their schedule is disrupted.

In this paper, the scheduling rules (i)–(vi) are applied as hard constraints for our computational tests unless otherwise stated.

Table 1

Parameter settings characterising the schedule disruptions generated in the problem library.

Parameter values			
TND	0.01	0.03	0.05
SD	0	0.5	1

Constraints (vii) and (viii) are handled as soft constraints. This list is very similar to the list of constraints as described in Pato and Moz [9] who state that these rules are the most important nurse rescheduling policies and rules applied in practice.

Respecting all the hard rules and regulations, a nurse roster is comprised based on the multiple objectives which have different priorities (cf. Section 3) as provided in Table 2 with a higher weight denoting a higher priority.

5.1.2. Algorithmic parameter calibration

Based on experimental tests, we apply a stop criterion of 1000 and 5000 evaluated schedules. Within the limits of this stop criterion, we need to carefully design our algorithm such that we find the right balance between diversification and intensification. To that purpose, we apply a strict stop criterion to each local search heuristic and combine the three heuristics into a variable neighbourhood search in a particular order. We apply first the pattern-based local search. The number of nurses whose schedule is optimised is dependent from the number of schedule disruptions (1%, 3% and 5%) and amounts 10%, 20% and 25% respectively. Then the day-based local search is applied, which optimises 30% of the day rosters in the schedule. Last, the schedule-based local search is exercised which is limited to re-distributing only 20% of all possible three- and two-day roster combinations.

Other parameters that have to be calibrated are the population size (20 for all problem settings), the percentage of nurses subject to the combination step (25%, 40% and 50% dependent on the number of schedule disruptions (for 1%, 3% and 5% resp.)) and the

mutation percentage. Mutation is carried out on 30% of the newly generated solution elements. This step involves the assignment of completely random schedules to 20% of all nurses.

5.2. Algorithmic performance

In this section, we analyse the effect of the meta-heuristic principles implemented in our procedure. More precisely, different search strategies and operation modes are implemented and the outcomes of the different runs of our algorithm are compared. For each strategy we display the solution quality averaged over the solved problem instances (Z), the percentage deviation from the best performing heuristic procedure ($\%Dev$) and the required CPU time (in seconds) (CPU). In order to test the effect of the different strategies we start from the best performing heuristic procedure and implement a certain strategy or characteristic of the procedure. In this way, we can analyse unambiguously the impact and contribution of the different strategies in terms of solution quality.

Population: In Section 4.3, we indicated that we operate on the Pareto optimal set of solution elements (*Pareto set*). However, the traditional heuristic optimisation theory learns that evolutionary algorithms typically operate on a population that evolves according to the principle survival of the fittest, i.e. only the best set of solution elements in terms of solution quality are retained (with a threshold distance that avoids duplication in the population) (*Best set*). The computational results in Table 3 reveal that exploiting a Pareto optimal set of solution elements outperforms the strategy operating on the best solution elements by 11.46%. The main reason for these better results is that maintaining a Pareto optimal set of solutions artificially advocates diversity in the search process.

Selection: We compare the proposed chromosome selection method (cf. Section 4.4) (*fitness probability*) with a strategy that chooses two elements randomly with all population elements having an equal probability to be selected (*equal probability*). Table 3 reveals that incorporating some intelligence in the selection system leads to a better computational performance. The second strategy outperforms the first by 10.75%.

Table 2
Relative priorities and weights for the different objective function components.

	Objective	Weight
Z_2	Infeasibilities	100,000
Z_1	Understaffing	10,000
Z_1	Overstaffing	5000
Z_2	Nurse preferences	100
Z_3	Fairness	50

Table 3
Computational results for the different optimisation strategies (1000 schedules).

Optimisation strategy		Z	$\%Dev$	CPU
Population	Pareto set	27,640	0.00	15.986
	Best set	30,809	11.46	28.756
Selection	Equal probability	30,612	10.75	16.401
	Fitness probability	27,640	0.00	15.986
Combination	1COP	33,432	20.95	16.280
	Best tournament	30,189	9.22	16.250
	Random tournament	32,835	18.79	15.984
	Random	36,176	30.88	21.074
	Proposed combination method	27,640	0.00	15.986
Mutation	Yes	27,640	0.00	15.986
	No	28,063	1.53	15.985
Improvement—contribution	No improvement method	306,164	1007.67	268.035
	No PBLS	30,007	8.56	16.763
	No DBLS	129,348	367.97	22.675
	No SBLS	70,835	156.28	14.827
	All improvement methods	27,640	0.00	15.986
Improvement—optimisation order	Fixed order	33,056	19.59	13.401
	Random order	31,898	15.40	26.667
	Static order worst-best	28,505	3.13	16.595
	Dynamic order worst-best	27,640	0.00	15.986

Combination: In this paper, we have proposed a combination method (cf. Section 4.5) that constructs combinations between an initiating and a guiding solution. In order to validate the performance of this mechanism we compare this crossover operator with established crossover operators from the personnel rostering literature:

- The nurse-based one-point crossover of Aickelin and Dowsland [19] randomly selects a crossover point between 1 and the number of nurses, such that the individual nurse schedules before the crossover point are copied from the one parent and the individual nurse schedules after the crossover point are copied from the other parent (1COP).
- The nurse-based crossover with best tournament selection is based on Burke et al. [14] and creates a child schedule that combines the best individual nurse schedules from both parents (*Best tournament*).
- The randomly selected nurse-based crossover is used by Aickelin and Dowsland [19], Burke et al. [14] and Dias et al. [21] and boils down to a complete random pairwise selection of the nurses' schedules from both parents (*Random tournament*).
- The random diversification mechanism embodies the multiple construction of a random solution and the application of the different local search heuristics (*Random*). This version is characterised in the literature as a multistart heuristic relying on a variable neighbourhood search.

Table 3 reveals that the variable neighbourhood search that starts each time from a completely random solution yields the worst results. This is an encouraging result as introducing intelligence in combining existing solutions returns beneficial results. Further, the results show that the one-point crossover operator is outperformed by the more disruptive crossover operator (i.e. the crossover operator based on random tournament selection) and the combination mechanisms incorporating a higher degree of problem-specific information (i.e. the proposed combination mechanism and the crossover operator based on best tournament selection). The proposed combination mechanism outperforms all other diversification mechanisms by more than 9.22%.

Mutation: We test also the effect of mutation when solving the nurse rostering problem (*Mutation vs No mutation*). The results reveal only a limited beneficial effect when incorporating the mutation operator as the procedure incorporating the mutation operator outperforms the procedure without the mutation operator with 1.53%.

Improvement: In analysing the improvement method we examine the role of each local search mechanism and the role of the order in which the involved nurse schedules and day rosters are optimised.

Improvement—contribution: The beneficial effect of the variable neighbourhood search can be clearly discerned when leaving the improvement method out of the algorithm (*No improvement method*). In order to establish the contribution of the different local search methods, we omitted each improvement method leading to three different versions of the algorithm, i.e. without the pattern-based local search (*No PBLS*), without the day-based local search (*No DBLS*) and without the schedule-based local search (*No SBLs*). Comparing the three local searches, the approach 'No DBLS' yields the worst results. This implies that this singular local search has the highest contribution in the improvement method. The schedule-based local search has a smaller but still significant contribution. The results reveal further that the pattern-based local search has only a limited contribution to the search process. These results contrast the observed effects for the traditional nurse rostering problem where the

pattern-based local search yields the highest contribution. A last version incorporates the three local search mechanisms (*All improvement methods*) which gives overall the best results. This result provides proof of the complementarity of the three local searches.

Improvement—optimisation order: Furthermore, we test how the order in which the nurse schedules and day rosters are optimised in these local search heuristics affect the solution quality. The following order strategies are examined:

- The nurses' schedules and day rosters are optimised in a fixed recurring increasing order starting from the first day of the rostering horizon (*Fixed order*).
- The nurses' schedules and day rosters are optimised each time in a new, completely random order (*Random order*).
- The nurses' schedules and day rosters are optimised each time in a new order that is composed before the application of the improvement heuristic. The order is based on the calculation of the improvement costs and ranks the worst nurse or day first and the best nurse or day last (*Static order worst-best*).
- The nurses' schedules and day rosters are optimised following a dynamic order as proposed in Fig. 4 (*Dynamic order worst-best*).

The computational results reveal the beneficial effect of incorporating problem-specific information. The third and fourth strategies clearly outperform the first two generic strategies. These results can be explained by the fact that implementing the third and especially the fourth strategy leads to a better balance between diversification and intensification. The parameter calibration reveals that the percentage of nurses and/or days to optimise in a single run of the local search heuristics is far larger for the 'Fixed order' and 'Random order' strategies (resp. 90% and 100%) as these orderings do not incorporate problem-specific information. These two strategies spend too much iterations on intensification to attain good quality results and less on diversification while the 'Static order worst-best' and 'Dynamic order worst-best' strategies need to spend less iterations on intensification (resp. 60% and 20%) to obtain the local optima and have a more diversified search which yields better results. This effect is even increased when dynamically adapting the ordering in which the nurses and/or days are improved as the proposed fourth strategy outperforms the third strategy.

These findings confirm the need and benefit of using a dedicated algorithm tailored to the rostering problem over applying traditional rostering algorithms to the nurse rostering problem. Traditional nurse rostering algorithms investigate typically all roster areas whereas the proposed approach examines only a limited part of the roster area. These findings are exemplified in two alternative computational experiments that are carried out on a different generated set of problem instances for validation purposes and to avoid problems of overfitting. We compare the rostering approach that optimises the complete roster area in a single local search run (100%) with different rostering approaches that examine only a limited part of the roster area. The results are displayed in Table 4. Experiment 1 returns the solution quality Z obtained after 5000 schedule iterations ($\# Iter$). Experiment 2 provides the number of iterations required for the rostering approaches to obtain a similar solution quality as for the rostering approach that optimises all areas of the entire roster.

The table reveals that it is more beneficial that only a part of the roster is subject to the improvement methods compared with optimise each time the complete roster as the 100% approach is outperformed by the three other approaches. The best results are obtained when only a very limited fraction 10–30% of the roster is

Table 4
The effect of limiting the roster area subject to the improvement methods.

Strategy		Experiment 1		Experiment 2	
		Z	# Iter	Z	# Iter
Rostering	100%	24,297.22	5000	24,297.22	5000
Rerostering	70–90%	22,979.63	5000	24,297.22	1593.49
	40–60%	22,183.33	5000	24,297.22	1090.32
	10–30%	20,915.74	5000	24,297.22	619.25

Table 5
Computational comparison with Pato and Moz [9] varying the minimum number of consecutive working days.

Constr. Par.	Procedure	Z	CPU	%Feas solution	%Feas rosters
1	Pato and Moz (2008)	16,541.63	4.35	89.93	91.15
	<i>This procedure</i>	5653.99	8.43	100	100
2	Pato and Moz (2008)	25,658.17	12.23	35.42	63.18
	<i>This procedure</i>	13,087.91	12.45	100	100

investigated. Additionally, this approach obtains the solution quality of the rostering approach already after 619.25 schedule iterations which gives proof of the better convergence of applying rerostering methods over the traditional rostering methodology.

5.3. Benchmarking and comparison with the existing literature

In this section, we present detailed computational benchmark results of the best version of the evolutionary algorithm identified in Table 3 and compare the obtained results with another procedure out of the nurse rostering literature. As indicated in our literature overview, Pato and Moz [9] developed a comparable procedure for the nurse rostering problem that explicitly incorporates multiple objectives. Their genetic algorithm uses a constructive heuristic to rebuild a nurse roster from a chromosome representation. However, in their paper it is indicated that infeasibilities in the individual nurse schedules can occur when using this constructive heuristic. These infeasibilities are encountered due to the presence of succession and consecutiveness constraints in nurse (re-)rostering (e.g. the minimal rest time between working shifts). The personnel scheduling literature (cf. the overview of Burke et al. [29]) and practice, where cyclic scheduling is still exercised in many application areas, emphasise the importance and due presence of these constraints. In the computational experiments of Pato and Moz [9] the constraint parameter for the minimal number of consecutive working days is one, which partly explains why they do not pay (much) attention to solve these infeasibilities. We compare in Table 5 our procedure with Pato and Moz [9] when a minimal number of consecutive working days of one day is imposed. In our test experiments we also solve our set of problem instances by further increasing this minimal number of consecutive working days to two. In order to observe the resulting effects on the functioning of both procedures, we display the solution quality (Z), the required CPU time (CPU), the percentage of problem instances for which a feasible solution is found after 1000 iterations (% Feas solution) and the percentage of times the solution elements are feasible during the search process (% Feas rosters).

The results reveal that the procedure of Pato and Moz [9] leads increasingly to infeasible solutions as the minimum number of consecutive working days increases. When the minimum number of consecutive working days is one, the procedure leads for

89.93% of the problem instances to a feasible solution. This percentage of feasible solutions decreases rapidly to only 35.42% as the minimum number of consecutive working days is two. For the files for which a feasible solution is obtained the heuristic decoder leads resp. in only 91.15% and 63.18% of the iterations to a feasible nurse roster. The proposed heuristic always leads to feasible solutions with respect to the hard constraints. In order to compare unambiguously the performance of the proposed procedure, we selected for both runs the subset of instances for which feasible solutions are obtained by the procedure of Pato and Moz [9]. The results reveal that in both cases the proposed algorithm outperforms clearly the approach of Pato and Moz [9] (Z).

In order to improve the comparability between both procedures, we extend the algorithm of Pato and Moz [9] to include the presence of succession constraints (in particular, a minimal number of two consecutive working days) based on the propositions of Beasley and Chu [30]. These authors propose two ways of dealing with the incorporation of additional constraints, i.e. penalty functions and repair functions:

- A penalty function penalises infeasible solutions without distorting the fitness landscapes of feasible solutions. The extension of the objective function to penalty functions to avoid succession infeasibilities in individual nurse schedules does not have any influence on the fundamental approach of their algorithm as mentioned in Pato and Moz [9]. To implement this approach, we add an objective function component to avoid individual nurse schedule infeasibilities with respect to the hard time-related constraints. Each infeasible nurse schedule is penalised with a cost of 100,000.
- A repair function is a heuristic operator that transforms infeasible solutions into feasible solutions. This repair function is used to reinstate the schedule of a single nurse by searching the feasible schedule which resembles most to the nurse's schedule constructed by the crossover operator violating hard constraints. The repair function is implemented by defining a minimum cost flow problem on the same activity-on-the-node network as done for the combination method and the pattern-based local search with appropriate costs.

In Table 6 we compare our algorithm with the approach of Pato and Moz [9] with penalty function (Run 3) and with repair function (Run 4).

The results reveal that incorporating a penalty function to avoid schedule infeasibilities improves only slightly the percentage of feasible solutions. Using a repair function, however, leads to feasible solutions for all problem instances, like the proposed procedure. The repair operator is called upon in 78.71% of the iterations to restore feasibility after application of the heuristic decoder. As this repair function leads to a new evaluation of the nurse rosters, basically these repairs should be counted as an iteration which is not done in Table 6 (in the advantage of Pato and Moz [9]). Despite these functions to avoid infeasibilities and these uncounted iterations the results reveal that the obtained solution quality of both approaches (Z) is outperformed by the proposed procedure.

Table 6
Computational comparison with Pato and Moz [9] with penalty and repair function.

Procedure	Z	CPU	%Feas solution
Moz and Pato (2008)—penalty fct	237,036	4.428	43.40
Moz and Pato (2008)—repair fct	91,478	22.221	100
<i>This procedure</i>	27,640	15.986	100

6. Conclusion

In this paper, we proposed a heuristic optimisation tool for the nurse rostering problem that adequately revises and re-optimises a schedule of a set of heterogeneous nurses. The heuristic procedure comprehends an evolutionary meta-heuristic that operates on a Pareto optimal set of solutions. Moreover, the computational results show that the procedure yields a right balance between a well-performing diversification mechanism and three complementary intensification mechanisms. We provide computational insights involving the performance of our procedure that is tested on a new benchmark dataset. Computational experiments reveal that our procedure presents state-of-the-art results as we outperform the existing literature.

The idea of reconstructing personnel rosters has received only very limited attention in the literature. In this paper we applied a traditional evolutionary optimisation method to the nurse rostering problem. In recent years, many other methods in multiple objective optimisation are developed. An interesting topic for future research is to examine which method is most suitable for solving the nurse rostering problem. The introduction of the dataset in this paper provides an interesting starting platform for this algorithmic comparison. The optimisation literature has shown that hybridisation of meta-heuristic principles is another important and promising concept. A skilled combination of concepts of different meta-heuristics can provide a more efficient behaviour and a higher flexibility when dealing with real-world and large-scale problems. We believe that our optimisation method and the standard test design on a benchmark dataset are good starting points to work towards these future research directions.

Acknowledgements

We acknowledge the support for the postdoctoral research project fundings by the Fonds voor Wetenschappelijk Onderzoek (FWO), Vlaanderen, Belgium under contract number 3E009808T.

References

- [1] Ritchie K, Macdonald E, Gilmour W, Murray K. Analysis of sickness absence among employees of four NHS trusts. *Occupational Environmental Medicine* 1999;56:702–8.
- [2] Ernst A, Jiang H, Krishnamoorthy M, Owens B, Sier D. An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research* 2004;127:21–144.
- [3] Ernst A, Jiang H, Krishnamoorthy M, Sier D. Staff scheduling and rostering: a review of applications, methods and models. *European Journal of Operational Research* 2004;153:3–27.
- [4] Cheang B, Li H, Lim A, Rodrigues B. Nurse rostering problems—a bibliographic survey. *European Journal of Operational Research* 2003;151:447–60.
- [5] Burke E, De Causmaecker P, Vanden Berghe G, Van Landeghem H. The state of the art of nurse rostering. *Journal of Scheduling* 2004;7:441–99.
- [6] Moz M, Pato M. An integer multicommodity flow model applied to the rostering of nurse schedules. *Annals of Operations Research* 2003;119:285–301.
- [7] Moz M, Pato M. A genetic algorithm approach to a nurse rostering problem. *Computers & Operations Research* 2007;34:667–91.
- [8] Moz M, Pato M. Solving the problem of rostering nurse schedules with hard constraints: new multicommodity flow models. *Annals of Operations Research* 2004;128:179–97.
- [9] Pato M, Moz M. Solving a bi-objective nurse rostering problem by using a utopic Pareto genetic heuristic. *Journal of Heuristics* 2008;14:359–74.
- [10] Knighton S. An optimal network-based approach to scheduling and rostering continuous heterogeneous workforces. PhD thesis, Arizona State University, Tempe, AZ; 2005.
- [11] Clausen J, Larsen A, Larsen J, Rezanova N. Disruption management in the airline industry concepts, models and methods. *Computers & Operations Research* 2010;37:809–21.
- [12] Huisman D. A column generation approach for the rail crew re-scheduling problem. *European Journal of Operational Research* 2007;180:163–73.
- [13] Corominas A, Pastor R. Replanning working time under annualised working hours. *International Journal of Production Research* 2010;48:1493–515.
- [14] Burke E, De Causmaecker P, Petrovic S, Vanden Berghe G. Fitness evaluation for nurse scheduling problems. In: *Proceedings of congress on evolutionary computation, CEC 2001*, 2001. p. 1139–46.
- [15] Azaiez M, Al Sharif S. A 0–1 goal programming model for nurse scheduling. *Computers & Operations Research* 2005;32:491–507.
- [16] Bard J, Purnomo H. Preference scheduling for nurses using column generation. *European Journal of Operational Research* 2005;164:510–34.
- [17] Glover F, Laguna M, Marti R. Fundamentals of scatter search and path relinking. *Control and Cybernetics* 2000;29:653–84.
- [18] Marti R, Laguna M, Glover F. Principles of scatter search. *European Journal of Operational Research* 2006;169:359–72.
- [19] Aickelin U, Dowland K. Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem. *Journal of Scheduling* 2000;3:139–53.
- [20] Burke E, Cowling P, De Causmaecker P, Vanden Berghe G. A memetic approach to the nurse rostering problem. *Applied Intelligence* 2001;15:192–214.
- [21] Dias T, Ferber D, de Souza C, Moura A. Constructing nurse schedules at large hospitals. *International Transactions in Operational Research* 2003;10:245–65.
- [22] Maenhout B, Vanhoucke M. Comparison and hybridization of crossover operators for the nurse scheduling problem. *Annals of Operations Research* 2008;159:333–53.
- [23] Inoue T, Furuhashi T, Maeda H, Takaba M. A proposal of combined method of evolutionary algorithm and heuristics for nurse scheduling support system. *IEEE Transactions on Industrial Electronics* 2003;50:833–8.
- [24] Glover F, Laguna M. *Tabu search*. Kluwer Academic Publishers; 1997.
- [25] Maenhout B, Vanhoucke M. Branching strategies in a branch-and-price approach for a multiple objective nurse scheduling problem. *Journal of Scheduling* 2009;13:77–93.
- [26] Maenhout B, Vanhoucke M. An electromagnetic meta-heuristic for the nurse scheduling problem. *Journal of Heuristics* 2007;13:359–85.
- [27] Vanhoucke M, Maenhout B. On the characterization and generation of nurse scheduling problem instances. *European Journal of Operational Research* 2009;196:457–67.
- [28] Vanhoucke M, Coelho J, Debels D, Maenhout B, Tavares L. An evaluation of the adequacy of project network generators with systematically sampled networks. *European Journal of Operational Research* 2008;187:511–24.
- [29] Burke E, De Causmaecker P, Petrovic S, Vanden Berghe G. Variable neighbourhood search for nurse rostering problems. In: Resende M, Pinho de Sousa J, editors. *Metaheuristics: Computer Decision-Making*. Boston, MA: Kluwer Academic Publishers; 2004. p. 153–72.
- [30] Beasley J, Chu P. A genetic algorithm for the set covering problem. *European Journal of Operational Research* 1996;94:392–404.