PATAT 2010 manuscript No.
(will be inserted by the editor)

An Evolutionary Algorithm in a Multistage Approach for an Employee Rostering Problem with a High Diversity of Shifts

Bäumelt Zdeněk · Šůcha Přemysl · Hanzálek Zdeněk

Received: date / Accepted: date

Abstract This work deals with the problem of rostering employees at an airport. There are about a hundred different shifts in order to handle the irregular coverage constraints. Together, with the strict constraints, given by the collective agreement, the problem becomes difficult to solve. Common one stage algorithms, applied to this problem, produce rosters containing too many isolated days-on and days-off which makes the roster unusable. This paper suggests a three stage approach for the employees rostering problem where a set of different shifts is needed to satisfy the coverage requirements. The solution is based on the problem transformation to a simpler problem, thereupon, an evolutionary algorithm is used to determine a rough position of the shifts in the roster. The maximal weighted matching in the bipartite graph is used as the inverse transformation of the problem and the final roster is obtained by the optimization based on a Tabu Search algorithm.

Keywords high diversity of shifts · multistage approach · evolutionary algorithm · employee rostering

1 Introduction

This paper deals with a problem from the traffic sphere belonging to the domain of employee timetabling/employee rostering/personnel scheduling problems (ETPs). The main difference to the classical Nurse Rostering Problem (NRP) [1], [14] is in the shift coverage demand. A typical NRP considers a small set of shifts, e.g. {day, night} [9] or {early, late, night} [13]. On the contrary, in the ETP, motivated by real problems from the public transport (e.g.

Z. Bäumelt · P. Šůcha · Z. Hanzálek
Department of Control Engineering, Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2, 166 27, Prague 6, Czech Republic
E-mail: baumezde@fel.cvut.cz, suchap@fel.cvut.cz, hanzalek@fel.cvut.cz

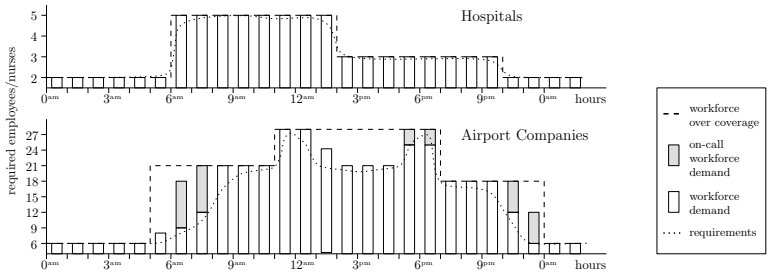

Fig. 1 The coverage function f_C examples

airport companies), the set of shifts can be quite large. This is caused by the fact that the coverage constraints are given by a so called *coverage function* f_C [6] determining the number of employees required each hour (see Fig.1). The f_C reflects the changes in the workforce demand during the day, that are caused by the traffic peaks and, in our case, it also fluctuates for different days and different seasons of the year. There are two basic possibilities how to cover peaks in the f_C. Either it can be fit using the 'classical' set of shifts, or it can be satisfied with an *extended set of shifts* with the size of dozens or hundreds of shifts. The extended set of shifts consists, not only, of shifts with different start and finish times, but also contains split shifts and on-call shifts. The split shifts facilitate coverage of the traffic peaks during the day, while the on-call shifts are used as an alternative for employees' sick leaves and other unanticipated causes. This approach makes the ETP more difficult, but it allows one to minimize the personnel expenses caused by the over coverage of the workforce. Two examples of f_C, corresponding to real data, are shown in Fig. 1. The first one is f_C typical for NRPs considering three shifts {early, late, night}. The second one is f_C typical for airport companies, where coverage requirements are depicted by a dotted curve. A dashed line corresponds to the over coverage of the workforce demand when a set of shifts with a small size is used. The gray blocks at the second f_C represent another feature – *on-call hours*. There are also *before shift on-calls*, when the employee is on the phone and comes to work one or two hours earlier if necessary. On the other hand, there are also *after shift on-calls* where the head of the ward decides whether the employee stays longer or leaves at the regular end of the shift. These before and after on-calls can resolve unpredictable changes in the f_C, e.g. traffic peak delays caused by bad weather conditions.

It is obvious that a high diversity of shifts is needed to cover the f_C in the second problem depicted in Fig. 1. We denote this problem as the Employee Timetabling Problem with a High Diversity of shifts (ETPHD).

The ETPHD is not only specific with a large variety of shifts but also through its set of constraints. The constraints that make this problem more complex are so called *block constraints*. These constraints are an extension of the 'classical' constraints limiting the number of consecutive days and are described in detail in Section 2.1.

1.1 Related Works

Summaries of the approaches for solving problems from the timatebling/rostering domain are published in [1], [4]. The most reviewed part of the ETPs belongs to the health care branch [2], [3]. In the ideal case, i.e. a small set of shifts, a small set of employees and a simplified set of constraints, the problem can be solved by *Integer Linear Programming* (ILP) [9] leading to the optimal solution.

The ETPs can also be modeled as *Constraints Satisfaction Problems* (CSP), solved by constraint programming techniques [8]. A hybrid approach from the domain of the declarative programming was presented in [7] on a simplified NRP where the authors proposed an automatically implied constraint generation. Through this hybrid technique, the ratio of the solved NRPs can be increased. Furthermore, this approach allows one to discover non-solvable problems before search, for some instances.

It is impracticable to use the optimal approaches like ILP when more difficult ETPs are considered. In this case, heuristic approaches are applied or the solved ETP is separated into its subproblems. These two possibilities are sometimes joined together to attain suboptimal solutions.

One of the most applied metaheuristic approaches for ETPs is a *Tabu Search Algorithm* (TSA). A two stage approach is described in [10] where, in the first step, a feasible solution with respect to hard constraints is found and, in the second step, a TSA based optimization is used. Similar stage separation is described in [5] where the comparison of two approaches (TSA and *Memetic Algorithm* (MA)) for the optimization stage were introduced. In a general way, TSA is faster than MA, but its computation time considerably depends on the previous initialization stage.

The nearest problem to the problem described in this paper, from the coverage constraints point of view, is described in [6]. The coverage is expressed as a varying number of staff needed for each grade throughout the day. The presented method is a two stage approach, previously described in [11] where a MA is used in the initialization stage and a TSA is employed in the optimization stage. The time interval coverage constraints are met by different combinations of shifts applied in the second stage with TSA. However, the number of shifts (not their combinations) considered in this work for one grade is less than ten.

In terms of the NRP classification proposed in [12], the presented ETPHD can be categorized as ASBI|TVNO|PLGM.

1.2 Contribution and Outline

In this paper, we introduced a multistage approach for handling the ETPHD specified by the high diversity of shifts. The basic idea lies in a transformation of the extended set of shifts to a simpler one. The transformed timetable is initialized by an evolutionary algorithm (the first stage) and the problem instance is transformed back by an algorithm based on matching in the bi-

partite graph (the second stage). The objective of these stages is to determine the rough position of the blocks of shifts. The final roster is obtained during the optimization based on the TSA (the third stage). This stage uses our adaptation of the TSA suggested in [5]. The contributions of the paper are:

a) a transformation allowing one to solve the ETPHD described in Sections 3 and 5
b) an ILP model presented in Section 3
c) an algorithm for the first stage based on a evolutionary algorithm (EA) shown in Section 4

The paper is organized as follows: Section 2 outlines the motivation problem at the airport. Section 3 explains the problem transformation to a problem with a reduced set of shifts and shows its ILP model. The transformed problem is solved in Section 4 by an EA. The inverse transformation is described in Section 5. Experiments and performance evaluation are summarized in Section 6 and the last section concludes the work.

2 Problem Statement

The problem solved in this paper is inspired by a real ETPHD from the traffic sphere. This problem is outstanding through its extended set of shifts where the shifts differ, not only, in the starting and finish times. There are also different split shifts and shifts prolonged by several before and after on-call hours.

The goal of the ETPHD is the same as in the NRP, to assign the requested shifts from the set of shifts to the employees with respect to the given constraints that are discussed below in detail.

2.1 Constraints

Constraints considered in the ETPHD are divided into two groups. The first group is stated as *hard constraints* that have to always be fulfilled. On the other hand, *soft constraints* can be violated, but their non-fulfillment is penalized in the objective function. However, it is necessary to notice that the constraints separation is not firm. It is more flexible to move some hard constraints to the group of soft and assign them large penalties. With a higher count of hard constraints it is more difficult to search the state space of the ETPHD.

The hard constraints considered in this problem are:

(c_1) An employee cannot be assigned to more than one shift per day.
(c_2) Shifts requiring a certain grade has to be covered by employees with this grade.
(c_3) Over coverage of shifts is not allowed.
(c_4) Under coverage of shifts is not allowed.
(c_5) The minimal time gap of free between shifts must be kept.

(c_6) Personnel requests must be considered – like fixed shift assignment, day-off requests, partial day-off requests (e.g. an employee is able to work to 5pm).
(c_7) The maximum and minimum number of consecutive days-on and maximum hours have to be kept.
(c_8) The minimal block rest between the blocks have to be fulfilled.
(c_9) Valid blocks of shifts must be respected, e.g. no more than one split shift is allowed in the block.
(c_{10}) The minimal block rest after 2 consecutive night shifts must be kept, e.g. 70 hours free.

The constraint (c_5), which is considered differently than normal is remarkable for this restriction. It defines a *minimal time gap* between two shifts equal to 12 hours. This minimal time gap can be shortened down to 10 hours subject to a condition that the following minimal time gap will be prolonged by the time equal to the previous shortage. The hard constraint (c_6) keeps the *fixed shifts* in the roster from unacceptable assignments, e.g. a planned business trip or holidays must be respected.

What makes this ETPHD problem difficult are the hard constraints (c_7) and (c_8) dealing with the so called *block of shifts* (c_9). This block is defined as a sequence of consecutive shifts where block rest between each two shifts in the block does not exceed the defined *minimal block rest* covered by (c_8), e.g. 45 hours. The hard constraint (c_7) defines that the count of working shifts in each *block* is less than or equal to the maximal shift count. Likewise, the number of working hours in the block is limited. The last block constraint (c_9) limits the number of certain shifts in the block. These constraints make the situation more complex since the position of the blocks is crucial for the quality of the resulting schedule. Therefore, in our opinion, it rules out the majority of the single stage approaches since the rough position of the block should be determined in the first stage respecting the fixed shifts in the roster (e.g. planned holidays, planned business trips, etc.).

The soft constraints, considered in the ETPHD, are:

(c_{11}) The maximum number of shifts of a given type performed in the planning period should not be exceeded, e.g. a max. of 7 night shifts during 5 weeks.
(c_{12}) Overtime hours should be balanced according to an employee workload.
(c_{13}) Weekend and night working hours should be in balance according to an employee workload.
(c_{14}) Hours of a certain shift kind (early, late, night, split and on-call shifts) should be balanced.

In order to simplify the benchmarks, only the constraints (c_1), (c_3)–(c_9) and (c_{12}) are taken into account in the rest of the paper. Constraints (c_2) and (c_{10}) can be easily incorporated into the mathematical model as well.

2.2 Problem Formalization

Let E be a *set of employees*, D represents a *set of days* from the whole planning period and S denotes the *extended set of shifts*. Consequently, the roster is

represented by R, a binary matrix such that $\forall i \in E, \forall j \in D, \forall s \in S$

$$R_{ijs} = \begin{cases} 1, & \text{the shift } s \text{ is assigned to the employee } i \text{ on the day } j \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

When the roster contains a *fixed shift* s, defined due to (c_6), the corresponding $R_{ijs} = 1$ is a constant and another shift can not be assigned to this position.

The coverage constraints (c_3) and (c_4) from Section 2.1 are expressed by a binary matrix RS where $RS_{sj} = 1$ iff the shift $s \in S$ is required on the day $j \in D$. Subsequently, in relationship to constraint (c_5), we can define a binary matrix with *shift precedences* SP so that

$$SP_{s_1 s_2} = \begin{cases} 1, & \text{the shift } s_1 \text{ can be followed by } s_2 \text{ on the subsequent day} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $s_1, s_2 \in S$.

Even though there is a large number of different shifts, set S can still be joined into groups given by a mapping $\mathcal{M} : S \longmapsto K$ where $K = \{\mathcal{F}, \mathcal{H}, \mathcal{E}, \mathcal{L}, \mathcal{N}, \mathcal{S}, \mathcal{O}\}$ is a *set of shift kinds*. The set of shift kinds consists of {free $\mathcal{F}$, required free or holiday $\mathcal{H}$, early shifts $\mathcal{E}$, late shifts $\mathcal{L}$, night shifts $\mathcal{N}$, split shifts $\mathcal{S}$ and on-call shifts $\mathcal{O}$}. Let K_W, K_F and K_S be subsets of K defined as follows

$$\begin{aligned} K_W &= \{\mathcal{E}, \mathcal{L}, \mathcal{N}, \mathcal{S}, \mathcal{O}\} \\ K_F &= \{\mathcal{F}, \mathcal{H}\} \\ K_S &= \{\mathcal{S}, \mathcal{O}\}. \end{aligned} \tag{3}$$

In other words, K_W is a subset of *working shift kinds*, where K_F represents *free shift kinds*. The last subset K_S consists of *split shift kinds* and *on-call shift kinds*. Furthermore, for each $k \in K$, let L_k be an *average shifts length* of kind k so that $L_k = \text{avg}_{s \in S | \mathcal{M}(s) = k} |s|$ where $|s|$ is the length of the shift $s \in S$.

Finally, *workloads* of all employees E are defined by a non-negative vector W according to the length of planning period.

3 Mathematical Model of the Transformed Problem

The goal of the first stage of the algorithm is to design the rough position of the blocks where the shifts should be placed in an accordance to the given constraints. The *rough blocks* of shifts can be modeled as blocks of days-on separated by days-off. The output of the first stage, presented in this paper, gives extra information in the form of which kind of shift $k \in K$ should be assigned on which day in the block. The first stage is described in this and the following section. This section presents a mathematical model based on a transformation used in the first stage.

3.1 Transformation $\mathcal{SK}$

Let $\mathcal{SK}$ be a transformation following from the mapping $\mathcal{M} : S \longmapsto K$. The $\mathcal{SK}$ transforms the ETPHD to ETPHD^K, specifically R to R^K where $R^K_{ijk} = 1$ iff the shift kind k is assigned to the employee i on the day j. In the same way, RS becomes RS^K where RS^K_{kj} is number of required shifts of kind k for the day j. Finally, SP becomes SP^K such that $SP^K_{k_1k_2}$ expresses, whether the shifts of kind k_1 can be followed by the shifts of kind k_2. The transformation is defined by equations (4), (5) and (6).

$$R^K_{ijk} = \begin{cases} 1, & \exists s \in S \mid R_{ijs} = 1 \wedge \mathcal{M}(s) = k \\ 0, & \text{otherwise} \end{cases}, \quad \forall i \in E, \forall j \in D, \forall k \in K \tag{4}$$

$$RS^K_{kj} = \sum_{s \in S \mid \mathcal{M}(s)=k} RS_{sj}, \quad \forall k \in K, \forall j \in D \tag{5}$$

$$SP^K_{k_1k_2} = \begin{cases} -1, & \forall s_1, s_2 \in S \;\middle|\; \begin{array}{l} SP_{s_1s_2} = 0 \\ \mathcal{M}(s_1) = k_1 \\ \mathcal{M}(s_2) = k_2 \end{array} \\ 0, & \forall s_1, s_2 \in S \;\middle|\; \begin{array}{l} SP_{s_1s_2} = 1 \\ \mathcal{M}(s_1) = k_1 \\ \mathcal{M}(s_2) = k_2 \end{array} \\ a, & \text{otherwise} \end{cases}, \quad \forall k_1, k_2 \in K \tag{6}$$

The positive penalty cost a of $SP^K_{k_1k_2}$ reflects the cases, when the precedence of the shift kinds $k_1, k_2 \in K$ is not obvious in general, but for most of the combinations of shifts $s_1, s_2 \in S \mid \mathcal{M}(s_1) = k_1 \wedge \mathcal{M}(s_2) = k_2$ is permitted, e.g. $SP^K_{\mathcal{L},\mathcal{E}}$.

3.2 Integer Linear Programming Model of ETPHD^K

The roster formed by rough blocks can be stated by an ILP model. The model uses a multicriteria objective function Z considering a linear combination of the constraints (c_3), (c_4) and (c_{12}) fulfillment, where $\alpha, \beta > 0$ are weights of the criterions. These criterions are evaluated by the piecewise linear functions (e.g. absolute value function) penRS and penW. The penRS function reflects the over and under coverage of the assigned shift kinds, while the penW function corresponds to the coverage of the employees' workloads. These functions are represented in the ILP model by a set of auxiliary variables that are not incorporated into equations (7)–(13) in order to make the model more readable.

$$\min Z = \min \left\{ \alpha \cdot \sum_{k \in K_W} \sum_{j \in D} \text{penRS} \left(RS^{K}_{kj} - \sum_{i \in E} R^{K}_{ijk} \right) + \beta \cdot \sum_{i \in E} \text{penW} \left(W_i - \sum_{j \in D} \sum_{k \in K} L_k \cdot R^{K}_{ijk} \right) \right\} \tag{7}$$

subject to

$$\sum_{k \in K} R^{K}_{ijk} = 1, \quad \forall i \in E, \forall j \in D \tag{8}$$

$$R^{K}_{ijk_1} + R^{K}_{i,j+1,k_2} - SP^{K}_{k_1 k_2} \leq 2, \quad \forall i \in E, \forall j = \langle 1, |D| - 1 \rangle, \forall k_1, k_2 \in K \tag{9}$$

$$\sum_{j=t}^{t+B_{maxL}} \sum_{k \in K_W} R^{K}_{ijk} \leq B_{maxL}, \quad \forall i \in E, \forall t = \langle 1, |D| - B_{maxL} \rangle \tag{10}$$

$$\sum_{k \in K_W} \left(R^{K}_{ijk} - R^{K}_{i,j+1,k} + R^{K}_{i,j+t,k} \right) \geq 0, \quad \forall i \in E, \forall t = \langle 2, B_{minL} \rangle, \forall j = \langle 1, |D| - t \rangle \tag{11}$$

$$\sum_{k \in K_W} \left(R^{K}_{ijk} - R^{K}_{i,j+t-1,k} + R^{K}_{i,j+t,k} \right) \leq 1, \quad \forall i \in E, \forall t = \langle 2, BR_{minL} \rangle, \forall j = \langle 1, |D| - t \rangle \tag{12}$$

$$\sum_{j=t}^{t+d} \sum_{k \in K_S} R^{K}_{ijk} \leq 1 + M \cdot \sum_{j=t}^{t+d} \sum_{k \in K_F} R^{K}_{ijk}, \quad \forall i \in E, \forall d = \langle B_{minL}, B_{maxL} \rangle, \forall t = \langle 1, |D| - d \rangle \tag{13}$$

The constraints of the ILP model are stated by equations (8) – (13). The first constraint equation (8) corresponds to the constraint (c_1), i.e. one shift per day is assigned. Similarly, equation (9) matches the constraint (c_5) represented by $SP^{K}_{k_1 k_2}$.

The constraints (c_7), (c_8) are given by (10), (11), (12). A maximal block length B_{maxL} of the working shift kinds is constrained by (10), while the following equation (11) considers the minimal length of the blocks B_{minL}. The last inequality from the block constraints (12) defines the minimal block rest length BR_{minL} between the block of shifts. Since the ILP model of the first stage is formulated on shift kinds, constraints (10)–(12) consider the average shift length L_k. Therefore, these equations limit the number of consecutive shift kinds instead of the sum of hours. Typical values for $(B_{maxL}, B_{minL}, BR_{minL})$

used in the solved ETPHD^K are $(5, 3, 2)$. In the last stage these constraints (c_7), (c_8) are reflected in the complete form.

The last equation (13) stands for (c_9) to avoid more shifts of the same kind in one block, e.g. it is not feasible to have more than one split shift, i.e. $k \in K_{\mathcal{S}}$, in one block. The term $M \cdot \sum_{j=t}^{t+d} \sum_{k \in K_F} R^K_{ijk}$ on the right side of (13) eliminates the equation in effect, when d consecutive days contain a free shift kind $k \in K_F$, i.e. it is not a block of the consecutive shifts. M is a big integer number.

4 Solution of the First Stage by an Evolutionary Algorithm

The transformation and the subsequent solution of the ILP model from Sectio 3.2 is the output of the algorithm's first stage. Due to enormous size of the ILP model it is not possible to find its feasible solution in a reasonable amount of time. Therefore, the solution of the first stage is found heuristically by the EA described below.

The roster is represented in the EA so that a couple of consecutive days are joined together. It reduces the overhead with the roster constraints' verification and the roster evaluation.

4.1 Modified Mathematical Model for the EA

The evolutionary algorithm, solving the first stage, uses the roster representation where the shift kinds assigned to the fixed number of consecutive days constitute a *gene*. The number of days representing the gene is called a *gene length* denoted as l. The consecutive genes are placed into *gene slots* GS indexed by p such that $p \in GS = \{\frac{j}{l} \mid j \in D \wedge (j \bmod l) = 0\}$. Thereafter, the gene $R^K[i,p]$ is a submatrix of R^K given by $R^K[i,p] = R^K_{ijk} \mid (p-1) \cdot l < j \leq p \cdot l$.

Both soft and hard constraints are taken into account as penalties in the objective function Z^E.

$$\begin{aligned} \min Z^E = \min \Bigg\{ & \alpha \cdot \sum_{k \in K_W} \sum_{j \in D} \mathrm{penRS}\left(RS^K_{kj} - \sum_{i \in E} R^K_{ijk} \right) + \\ & \beta \cdot \sum_{i \in E} \mathrm{penW}\left(W_i - \sum_{j \in D} \sum_{k \in K} L_k \cdot R^K_{ijk} \right) + \\ & \gamma \cdot \sum_{i \in E} \sum_{p \in GS} \mathrm{penUnsuit}\left(R^K[i,p] \right) + \\ & \delta \cdot \sum_{i \in E} \sum_{p \in \langle 1, |GS|-2 \rangle} \mathrm{penPrec}\left(R^K[i,p], R^K[i,p+1], R^K[i,p+2] \right) \Bigg\} \end{aligned} \tag{14}$$

The first two elements correspond to the objective function Z mentioned in (7), i.e. penalties of the under and over coverage of the required shift kinds and penalties of the unbalanced workload of the employees. The next two terms follow from the roster encoding. The third element of Z^E penalizes the suitability of the gene $R^K[i,p]$ given by the constraints expressed in (10)–(13). The last element is focused on the gene precedences. It is necessary to take into account the borders of the genes after the recombination, i.e. the kind precedences SP^K and the block constraints related to the neighborhood genes have to be checked and updated. Furthermore, the other constraints, e.g. (c_{10}) can be easily appended through this criterion.

4.2 Evolutionary Algorithm

The `Preprocessing` function of the EA (shown in Alg. 1) contains the described transformation $\mathcal{SK}$. Consequently, in the `GeneratePopulation` function, all permutations with a repetition of shift kinds $k \in K$ of length l are generated and evaluated with respect to the considered constraints. Similarly, the precedences of genes are assessed with respect to (9)–(13). This static part of EA accelerates the evaluation of Z^E.

The roster R^K containing the genes $R^K[i,p]$ represents an individual $\mathcal{I}$ of a population $\mathcal{P}$. The initial population $\mathcal{P}_0$, containing $pSize_0$ individuals, is created randomly by the `GeneratePopulation` function in the following way. For each individual $\mathcal{I} \in \mathcal{P}_0$, the employees $i \in E$ are selected according to their count of the fixed shifts. The selection itself is similar to the rank selection of

Input : $ETPHD$ instance
Output: Roster R^K

0 $ETPHD^K \leftarrow$ `Preprocessing`$(ETPHD)$;
1 $\mathcal{P}_0 \leftarrow$ `GeneratePopulation`$(ETPHD^K, pSize_0)$;
2 **foreach** $\mathcal{I} \in \mathcal{P}_0$ **do** `Evaluate`$(\mathcal{I})$;
3 $\mathcal{P}_0 \leftarrow \mathcal{P}$;
4 **while** *stop condition is not met* **do**
5 $\quad \mathcal{P} \leftarrow$ `Select`$(\mathcal{P}, pSize)$; `// select the` $pSize$ $\mathcal{I} \in \mathcal{P}$
6 $\quad \mathcal{P}_N \leftarrow \emptyset$; `// clear population` $\mathcal{P}_N$
7 $\quad$ **for** $i \leftarrow 1$ **to** $oCount$ **do** `// breed` $oCount$ `offsprings`
8 $\quad\quad [\mathcal{I}_1, \mathcal{I}_2] \leftarrow$ `ChooseParents`$(\mathcal{P})$;
9 $\quad\quad \mathcal{I}_{ofs} \leftarrow$ `Crossover`$(\mathcal{I}_1, \mathcal{I}_2)$;
10 $\quad\quad \mathcal{I}_{ofs} \leftarrow$ `Mutate`$(\mathcal{I}_{ofs})$ with probability p_M;
11 $\quad\quad \mathcal{P}_N \leftarrow \mathcal{P}_N \cup \mathcal{I}_{ofs}$; `// add offspring` $\mathcal{I}_{ofs}$
12 $\quad$ **end**
13 $\quad$ **foreach** $\mathcal{I} \in \mathcal{P}_N$ **do** `Evaluate`$(\mathcal{I})$; `// evaluate` $\mathcal{P}_N$
14 $\quad \mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{P}_N$; `// merge populations`
15 **end**
16 $R^K \leftarrow \mathcal{I} \in \mathcal{P}$ with the lowest value of Z^E;
17 **return** R^K

Algorithm 1: An Evolutionary Algorithm pseudo-code

the EA. The top employees have the higher probability to be selected earlier than the others. When the employee is selected, it is necessary to choose the gene slot $p \in GS$ where the gene could be assigned. The gene slots selection is similar to the employees selection, i.e. according to the position of the fixed shifts in slot $R^K[i,p]$.

The multicriteria objective function Z^E, given by (14), is used in the `Evaluate` function. The stop condition of the evolutionary algorithm is based on the ratio of the Z^E improvement to Z^E during the last 20 populations. If this ratio is under a given threshold, the algorithm stops.

The rank selection is used in the `Select` function to reduce the number of individuals of $\mathcal{P}$ to $pSize$. Furthermore, the applied selection keeps the population unique, i.e. $\forall \mathcal{I}_1, \mathcal{I}_2 \in \mathcal{P}$ holds $\mathcal{I}_1 \neq \mathcal{I}_2$. Finally, the elitism set is supplemented during the computation to keep the best $\mathcal{I} \in \mathcal{P}$ alive.

There are two basic possibilities, how to perform the `Crossover` function in the ETPHD^K. A uniform crossover in a horizontal dimension, i.e. with the certain probability p_{CH}, the better roster of employee i with respect to Z^E, is chosen from the individuals $\mathcal{I}_1, \mathcal{I}_2$. A typical value of $p_{CH} = 0.6$ makes the better roster from two individuals more favorable. An advantage is that there is no need to apply any repair operators for the horizontal crossover, because the whole rosters of the employees are copied to the offspring, i.e. all constraints related to precedences (kind precedence, gene precedence) are satisfied. The impact of the crossover to the 'vertical' constraints, e.g. shift kinds coverage, is incorporated to the objective function Z^E.

On the contrary, for a vertical dimension, it is better to apply the 1-point crossover instead of the uniform one. For each point of the crossover, it follows that the objective function Z^E increases rapidly due to precedence constraints violations. Then, the repair operators have to be applied to these violations. The application of repair operators is time consuming in comparison with the rest of the evolutionary algorithm. Therefore, this type of crossover is suitable to use as a diversification only, when the algorithm is caught in the local optima.

The mutation operator is applied on the offspring $\mathcal{I}_{ofs}$ after the crossover a with probability p_M. The value of p_M is typically 0.2. The mutation is focused on randomly selected employees from the roster. For each employee, a few genes $R^K[i,p]$ are changed in a random way, too.

5 The Second Stage – Inverse Transformation $\mathcal{KS}$

The objective of the first stage is to determine the rough position of the blocks and to assign a shift kind $k \in K$ to each position of the block. The second stage transforms the roster back when the shift kinds K are substituted by the required shifts $s \in S \mid RS_{sj} = 1$ for the day $j \in D$. This inverse transformation is based on the maximum weighted matching in a bipartite graph G_j where $j \in D$ is an index of the day.

For a day $j \in D$, let G_j be a bipartite graph with bipartition $V(G_j) = S(j) \cup E$, where $S(j)$ is a set of shifts required for the day j, so that $S(j) = \{s \in S \mid RS_{sj} = 1\}$. Furthermore, let $c : E(G_j) \to \mathbb{R}$ be the weights on the edges. There is an edge $(i, s) \in E(G_j)$ with $c\big((i, s)\big) = 1$ iff $\big(R^{K}_{ijk} = 1 \mid \mathcal{M}(s) = k \wedge k \in K_W\big) \wedge \big(SP_{s_{prev},s} = 1 \mid s_{prev} \in S(j-1) \wedge R_{i,j-1,s_{prev}} = 1\big)$ where s_{prev} is a shift assigned to $i \in E$ on the previous day. Moreover, there are also edges $(i, s) \in E(G_j)$ with lower weight $c\big((i, s)\big) = \epsilon(s, s_{prev}) < 1$ iff $\big(\exists k \in K_W \mid R^{K}_{ijk} = 1\big) \wedge \big(SP_{s_{prev},s} = 1 \mid s_{prev} \in S(j-1) \wedge R_{i,j-1,s_{prev}} = 1\big)$. These edges represent an assignment which is still possible but it is not preferred, i.e. $\mathcal{M}(s)$ is not the kind assigned to this position in the block. Thereafter, the weight $\epsilon(s, s_{prev})$ reflects how the shift s fits into the block when s_{prev} is placed on the day before.

The algorithm of the inverse transformation consecutively, for $j = 1$ to $j = |D|$, generates graphs G_j and looks for the maximal weighted matching M_W. When $(i, s) \in M_W$ the shift $s \in S(j)$ is assigned to the employee $i \in E$ on the day j, i.e. $R_{ijs} = 1$.

The result of the second stage (R) is optimized in the third stage which can be based on common techniques, e.g. a Tabu Search algorithm [5] or other heuristic approaches.

6 Experiments

The presented experiments show results obtained on real data from the airport. A one stage approach where the algorithm is the adaptation of the algorithm presented in [5] is used for comparison. The three stage approach uses the same algorithm in the third stage, but its execution is preceded by two initialization stages described in this paper.

The length of the gene used for our instances was tuned during the experiments and the best results were reached with the gene length $l = 4$. This value correlates with the constants B_{maxL}, B_{minL} and BR_{minL} of the block constraints.

The computational results are summarized in Tab. 1. For each test period the table shows the period length $|D|$, the number of employees $|E|$, the number of shifts $|S|$ considered in the period and the ratio of the fixed shifts F. The approaches are compared on the number of unplaced shifts $|\overline{S}|$, the number of

Table 1 Experiments

period	$\|D\|$	$\|E\|$	$\|S\|$	F[%]	1 stage approach		3 stage approach		ΔZ[%]
					$\|\overline{S}\|$	C_S/C_D	$\|\overline{S}\|$	C_S/C_D	
nov09	28	90	79	33.71	16	37/25	2	9/8	15.5
dec09	28	90	82	32.16	13	35/21	0	8/6	20.3
jan10	35	90	89	32.97	12	39/22	0	9/6	17.9
mar09	35	94	103	41.64	24	43/28	5	11/9	13.8

isolated days-on/days-off C_S and the number of two consecutive days-on C_D. The overall objective function improvement is presented in the last column ΔZ.

The last instance in Tab. 1 with the largest set of shifts and employees is remarkable. For this instance, a few of the shifts have not been assigned, but in the objective function and all other measures point of view, the obtained roster for this problematic planning period is much better. This period was made more difficult by the additional mandatory courses (reflected in the value of fixed assignments of shifts F) that all of the employees should have passed, if at all possible, during this period.

The CPU time of the three stage approach is higher due to the first stage of the algorithm where the EA determines the rough position of the shifts. The average CPU time of the first stage is around 5 minutes depending on the number of the fixed shifts. On the other hand, the extra 5 minutes given to the algorithm used in the one stage approach does not lead to a significant objective function improvement. The computation time of the second stage is negligible with respect to the first stage and it is smaller than 6 seconds.

7 Conclusion

In this paper, we introduced a three stage heuristic algorithm for the employee timetabling domain. The solved problem, motivated by a real employee rostering at the airport, is characterized in that the coverage function typically consists of two peaks and the level of the workforce demand differs from hour to hour by up to five employees. In order to satisfy the coverage requirements and to minimize the personnel expenses it is necessary to cover the requirements by an extended set of shifts. In our case, it is more than one hundred shifts. This fact together with the strict roster constraints, given by the collective agreement, makes the employee rostering very difficult. Therefore, the problem solution was decomposed into three stages, where the first one designs a rough position of the shift kinds (i.e. early shifts, late shifts, etc.), the second stage assigns shifts into the roster and the final stage fine-tunes the final roster.

The experiments have shown that the common single stage approaches are not applicable on the ETPHD. The resulting roster suffers from the presence of isolated days-on and days-off, which makes the roster unusable.

On the contrary, the improvement of the objective function reaches about 15 percent with the proposed three stage approach. The rosters produced by our algorithm are more compact, the count of isolated days-on and days-off is suitable and all the required shifts are assigned to the roster.

Acknowledgements This work was supported by the Ministry of Education of the Czech Republic under the Research Programme MSM6840770038.

References

1. Burke, E.K., De Causmaecker, P., Vanden Berghe, G., Landeghem, H.V.: The State of the Art of Nurse Rostering. Journal of Scheduling 7(6), 441-499 (2004)
2. Hung, R.: Hospital Nurse scheduling. Journal of Nursing Administration, 25(7/8), 21-23 (1995)
3. Cheang, B., Li H., Lim A. and Rodrigues B.: Nurse Rostering Problems – A Bibliographic Survey. European Journal of Operational Research, 151, 447-460 (2003)
4. Ernst, A.T., Jiang, H., Krishnamoorthy, M. and Sier, D.: Staff Scheduling and Rostering: A Review of Applications, Methods and Models. European Journal of Operational Research, 153(1), 3-27 (2004)
5. Berghe, G.V.: An Advanced Model and Novel Meta-heuristic Solution Methods to Personnel Scheduling in Healthcare, PhD Thesis, 276, University of Gent (2002)
6. Burke, E.K., De Causmaecker, P., Petrovic, S., Berghe, G.: Metaheuristics for Handling Time Interval Coverage Constraints in Nurse Scheduling. Applied Artificial Intelligence: An International Journal, 20(9), 743-766 (2006)
7. Wong, G.Y.C., Chun, A.H.W.: Nurse Rostering Using Constraint Programming and Meta-Level Reasoning. Engineering Applications of Artificial Intelligence, 17(6), 599-610 (2004)
8. Cheng, B.M.W., Lee, J.H.M. and Wu, J.C.K.: A Nurse Rostering System Using Constraint Programming and Redundant Modeling. IEEE Transactions on Information Technology in Biomedicine, 1, 44-54 (1997)
9. Azaiez, M.N., Al Sharif, S.S.: A 0-1 Goal Programming Model for Nurse Scheduling. Computers & Operations Research, 32(3), 491-507 (2005)
10. Burke, E.K., De Causmaecker, P. and Berghe G. V.: A Hybrid Tabu Search Algorithm for the Nurse Rostering Problem. In: B. McKay et al., Editors, Simulated Evolution and Learning, Selected Papers from the 2nd Asia-Pacific Conference on Simulated Evolution and Learning, SEAL 98, Springer Lecture Notes in Artificial Intelligence, vol. 1585, Springer, 187-194 (1999)
11. Burke, E.K., Cowling, P., De Causmaecker, P. and Berghe, G.V.: A Memetic Approach to the Nurse Rostering Problem. Applied Intelligence 15, 199-214 (2001)
12. De Causmaecker, P.: Towards a Reference Model for Timetabling and Rostering. Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2008)
13. Valouxis, C. and Housos, E.: Hybrid Optimization Techniques for the Workshift and Rest Assignment of Nursing Personnel. Artificial Intelligence in Medicine, 20, 155-175 (2000)
14. Staff Rostering Benchmark Data Sets, http://www.cs.nott.ac.uk/∼tec/NRP/

List of Variables

f_C	coverage function
d, i, j, k, p, s, t	indices
$\alpha, \beta, \gamma, \delta$	positive weights of the objective function
Z	objective function
S	set of shifts
$S(j)$	set of shifts required on the day $j \in D$
D	set of days from the planning period
E	set of employees
K	set of shift kinds, i.e. early, late, night, etc.

L_k	average shift length of shift kind k
B_{minL}	minimal block length (in days)
B_{maxL}	maximal block length (in days)
BR_{minL}	minimal block rest length (in days)
M	big M – big integer number
K_W	set of working shift kinds
K_F	set of non-working shift kinds
K_S	set of split shift kinds
W	vector of employees workloads
R	binary matrix representing the roster, where $R_{ijs} = 1$ iff shift $s \in S$ is assigned to the employee $i \in E$ on day $j \in D$
RS	matrix of requested shifts, where $RS_{sj} = 1$ iff the shift $s \in S$ is requested on day $j \in D$
SP	matrix of the shift precedences, so that $SP_{s_1 s_2} = 1$ iff the shift $s_1 \in S$ can be followed by $s_2 \in S$ on the consecutive days
a	penalty cost of shift precedence $SP^K_{k_1 k_2}$
R^K	binary matrix representing the roster, where $R^K_{ijk} = 1$ iff shift of kind $k \in K$ is assigned to the employee $i \in E$ on day $j \in D$
RS^K	matrix of the requested shifts, where RS^K_{kj} is the number of the required shifts of kind $k \in K$ on day $j \in D$
SP^K	matrix representing the feasibility of two consecutive shifts precedence
$R^K[i, p]$	submatrix of R^K – a gene
l	gene length (in days)
$\mathcal{P}$	population in the evolutionary algorithm
$pSize, pSize_0$	size of the population, size of the initial population
$oCount$	count of the offsprings breed in each iteration of the EA
$\mathcal{I}$	individual of the population $\mathcal{P}$
p_{CH}	probability of the better roster acceptance in the horizontal crossover
p_M	probability of mutation
G_j	bipartite graph
$V(G_j)$	set of vertices of G_j
$E(G_j)$	set of edges of G_j
c, ϵ	weights of the edges $E(G_j)$
M_W	maximal weighted matching $M_W \subseteq E(G_j)$