Machine-personnel assignment with training and interim worker requirements

Hatice Çalık · Pieter Smet · Everton Fernandes Silva

Received: date / Accepted: date

Abstract We introduce a machine-personnel assignment problem where personnel must be trained to operate certain groups of machines, whereas others can also be operated by interim workers hired at an additional cost. In order to achieve a feasible or a minimum-cost assignment for long-term planning, it may be necessary to cross-train employees on different machines. However, this requires switching to different machines frequently, which is not desirable for personnel. Therefore, each switch incurs a penalty cost. This specific characteristic makes the problem unique and complex when compared to well-known related problems in the literature. The problem requires assigning each machine to a minimum number of operators for each day of a given planning horizon while minimizing the total cost of hiring interim workers and switching machines. We provide integer programming formulations of the problem and develop an iterated local search method to solve instances with longer planning horizons. A comparison of the introduced methods on randomly generated instances indicate that the iterated local search algorithm is capable of finding high quality solutions within a reasonable time limit.

Keywords Cross-training · Staff assignment · Integer programming · Iterated local search

1 Introduction

The new problem studied in this paper, which we refer to as the *machine-personnel assignment problem with training and interim worker requirements* (MPATI), is motivated by a real-world application from the food and drink industry. Although we were introduced to this problem by a company operating in this specific sector, it is likely that similar applications exist in other sectors involving production or process optimization with possibilities of personnel training and hiring interim workers.

The MPATI deals with a division within the company which contains a fixed number of machines to be operated on a daily basis and permanent workers (personnel) whose shifts and working days in the planning horizon are predetermined. Although the majority of the machines must be operated only by personnel and under the supervision of a qualified worker, it is possible, and sometimes necessary, to hire interim workers to operate the others. Each machine requires a minimum number of operators (personnel or interim workers) assigned to every day, while an operator can only work on exactly one machine within a working day.

At the beginning of the planning horizon, only a subset of the workers is qualified to supervise or operate each machine independently. The set of qualified workers is not necessarily identical for different machines and, therefore, a worker qualified on one machine might be unqualified for another. Certain groups of workers may obtain the necessary qualification to operate certain machines after a training period. Training is possible only under the supervision of a qualified employee dedicated to work on those machines during the training days. In order to preserve their qualification, workers must avoid not operating the same machine for too many consecutive days. Otherwise, they lose their qualification and must undertake a short training period

H. Çalık - Corresponding author
KU Leuven, Department of Computer Science, CODeS; Leuven.AI, Belgium
E-mail: hatice.calik@kuleuven.be

P. Smet
KU Leuven, Department of Computer Science, CODeS; Leuven.AI, Belgium
E-mail: pieter.smet@kuleuven.be

E. Fernandes Silva
KU Leuven, Department of Computer Science, CODeS; Leuven.AI, Belgium
E-mail: everton.fernandesdasilva@kuleuven.be

which requires working on that machine under the supervision of a qualified person. Assigning workers to the same machines every day implies having only single-skilled personnel, which can result in infeasibilities in the long term. On the other hand, workers generally prefer not to switch machines frequently, given its additional adjusment period required. Therefore, a penalty is incurred every day a worker changes their machine assignment.

Learning and forgetting effects have been well-studied in scheduling problems. In these problems, job processing times are affected by the number of times a resource has performed a certain job [1,2]. The impact of these effects has been analyzed on the performance of both the system as a whole and individual workers [3,8,9].

Given the aforementioned restrictions, the MPATI requires a schedule which assigns workers to machines for each day of the planning horizon such that the total cost of hiring interim workers and the penalty cost of switching machines is minimized.

There are numerous examples of practical applications in the literature in which an accurate representation of employee skills is essential [5]. While there exist many papers on staff scheduling with fixed skills [10], the problem setting considered in this paper is rather uncommon due to the presence of decisions concerning employee cross-training. In one of the few related studies, Wirojanagud et al. [11] use an integer programming model to decide when to hire, fire or cross-train employees to minimize the costs incurred by production losses and training. In contrast to our problem setting, decisions are made at an aggregate level, without considering individual workers who may have varying preferences. De Bruecker et al. [4] propose a three-phase integer programming approach for optimizing the shift and training schedule of aircraft maintenance workers. Emphasis is placed on finding the optimal trade-off between low-cost schedules that require highly cross-trained workers and the training costs required to establish such a workforce. The impact of long-term staffing decisions has been studied by Komarudin et al. [7] who introduce a methodology for evaluating the effect on operational costs associated with staff allocation when exists multiple skills among the workforce being scheduled. For a comprehensive review of staff scheduling and personnel rostering literature, we refer to [6].

Our contribution in this paper is threefold. First, we introduce a new, relevant machine-personnel assignment problem combined with disaggregated cross-training and interim worker hiring decisions which have never been considered in the literature. Second, we provide mathematical formulations of this newly introduced problem with several valid inequalities. Third, we present a heuristic method which produces very high quality solutions within the imposed time limit.

The remainder of the paper is organized as follows: Section 2 formally describes the problem with an integer programming (IP) formulation. This section also introduces a mixed integer programming (MIP) variant of this IP and several valid inequalities obtained by exploiting practical properties of the problem. Section 3 provides an Iterated Local Search (ILS) algorithm for solving the MPATI. The performance of both mathematical models and the heuristic algorithm are then evaluated on randomly generated instances in Section 4. Finally, we conclude and offer directions for future research in Section 5.

2 Problem formulation

Given a set of workers P working the same shift of a process unit in a factory or a company, the problem consists of operating a set of machines M during a planning horizon denoted by a set of days D. While a subset M^N of machines M can only be operated by workers in P, the remaining set $M - M^N = M^I$ of machines can also be operated by interim workers who must be hired at additional cost f per person per day. Each machine $m \in M$ requires a minimum number of operators n_{md} on day $d \in D$.

At the beginning of the planning horizon, not every worker is qualified to independently operate every machine. This information is available via a skill matrix $S = [s_{pm}]$ where $s_{pm} = 1$ if worker $p \in P$ is qualified to operate machine $m \in M$ without supervision and $s_{pm} = 0$ otherwise. A worker $p \in P$ can undergo a training period of l_{pm} days (not necessarily consecutive) to become qualified to operate machine $m \in M$ without supervision. While trainees are considered as operating workers, they can only be trained on a machine if there is a qualified worker assigned to the same machine on the same day. A worker $p \in P$ must be assigned to exactly one machine on each working day $d \in D_p$ where $D_p \subseteq D$ is the set of days that worker p is available. Worker p who is not assigned to a machine for C_p consecutive working days of D_p must undergo a short retraining of R_p days (not necessarily consecutive) on that machine to reacquire their qualification. We refer to C_p as 'skill memory' throughout the remainder of the paper.

Although working on different machines on different days might be inevitable for workers, it is undesirable and therefore a penalty cost π is incurred per machine switch per worker. The objective is to find a machine-worker assignment for the entire planning horizon while minimizing the total cost of hiring interim workers and switching penalties.

In order to formulate this problem as an IP model, we define the following additional parameters:

e_{0p}: the index of the first working day of worker p.
e_{dpk}: the index of the k-th previous working day of worker p for day $d \in D_p$.
h_{pm}: the number of consecutive working days worker $p \in P$ has not been working on machine $m \in M$.
t_{pm}: the number of training days of worker $p \in P$ on machine $m \in M$.

Moreover, we introduce the following decision variables:

$x_{pmd} = 1$ if worker $p \in P$ is assigned to machine $m \in M$ on day $d \in D$, 0 otherwise.
$\gamma_{pmd} = 1$ if worker $p \in P$ is qualified to work on machine $m \in M$ on day $d \in D$, 0 otherwise.
$w_{pmd} = 1$ if worker $p \in P$ is qualified and working on machine $m \in M$ on day $d \in D$, 0 otherwise.
$\alpha_{pmd} = 1$ if worker $p \in P$ lost their qualification for machine $m \in M$ on day $d \in D$, 0 otherwise.
$\tau_{pmd} = 1$ if worker $p \in P$ is being trained on machine $m \in M$ on day $d \in D$, 0 otherwise.
$y_{pd} = 1$ if worker $p \in P$ switches to a different machine on day $d \in D$, 0 otherwise.
z_{md}: the number of interim workers assigned to machine $m \in M$ on day $d \in D$.

The following is an IP formulation for the MPATI.

$$(IP) \quad \min \sum_{m\in M^I}\sum_{d\in D} f z_{md} + \sum_{p\in P}\sum_{d\in D_p} \pi y_{pd} \tag{1}$$

$$\text{s.t.} \sum_{m\in M} x_{pmd} = 1, \qquad \forall p \in P, d \in D_p, \tag{2}$$

$$w_{pmd} + \tau_{pmd} = x_{pmd}, \qquad \forall p \in P, m \in M, d \in D_p, \tag{3}$$

$$\sum_{p\in P: d\in D_p} x_{pmd} \geq n_{md}, \qquad \forall m \in M^N,\ d \in D, \tag{4}$$

$$z_{md} + \sum_{p\in P: d\in D_p} x_{pmd} \geq n_{md}, \qquad \forall m \in M^I,\ d \in D, \tag{5}$$

$$\tau_{pmd} \leq \sum_{q\in P: d\in D_q} w_{qmd}, \qquad \forall p \in P, m \in M, d \in D_p, \tag{6}$$

$$x_{pmd} - x_{pme_{dp1}} \leq y_{pd}, \qquad \forall p \in P, m \in M, d \in D_p, \tag{7}$$

$$x_{pme_{dp1}} - x_{pmd} \leq y_{pd}, \qquad \forall p \in P, m \in M, d \in D_p, \tag{8}$$

$$w_{pmd} \leq \gamma_{pmd}, \qquad \forall p \in P, m \in M, d \in D_p, \tag{9}$$

$$\gamma_{pme_{0p}} \leq s_{pm}, \qquad \forall p \in P, m \in M, \tag{10}$$

$$\gamma_{pmd} \leq \sum_{k=1}^{C_p - h_{pm}} x_{pme_{dpk}}, \qquad \forall p \in P, m \in M, d \in D_p : e_{dp(C_p - h_{pm})} = 0, \tag{11}$$

$$\gamma_{pmd} \leq \sum_{k=1}^{C_p} x_{pme_{dpk}}, \qquad \forall p \in P, m \in M, d \in D_p : e_{dpC_p} \geq 0, \tag{12}$$

$$\sum_{j=0: j\in D_p}^{d} \tau_{pmj} \geq (l_{pm} - t_{pm})\gamma_{pmd}, \qquad \forall p \in P, m \in M, d \in D_p : s_{pm} = 0, \tag{13}$$

$$\gamma_{pme_{dp1}} - \gamma_{pmd} \leq \alpha_{pmd}, \qquad \forall p \in P, m \in M, d \in D_p, \tag{14}$$

$$\sum_{j=i: j\in D_p}^{d} \tau_{pmj} \geq R_p\gamma_{pmd} + R_p(\alpha_{pmi} - 1), \qquad \forall p \in P, m \in M, d, i \in D_p : i \leq e_{dpR_p}, \tag{15}$$

$$x_{pmd}, \gamma_{pmd}, w_{pmd}, \alpha_{pmd}, \tau_{pmd} \in \{0,1\}, \qquad \forall p \in P, m \in M, d \in D_p, \tag{16}$$

$$y_{pd} \in \{0,1\} \qquad \forall p \in P, d \in D_p, \tag{17}$$

$$z_{md} \in \{0\} \cup \mathbb{Z}^+ \qquad \forall m \in M, d \in D. \tag{18}$$

Objective function (1) minimizes the total cost of hiring interim workers and switching machines. Constraints (2) assign a machine to each worker on each of their working days, while Constraints (3) make sure that this worker is either training or qualified (but not both) for that machine on that day. Constraints (4) and (5) assign a sufficient number of operators to each machine for each day. Constraints (6) ensure that trainees are assigned to a machine only if a qualified worker is also assigned to the same machine. For each working day of each worker, Constraints (7) and (8) introduce a switch to be penalized in the objective function if on this day this worker works on a different machine than the one they worked on their previous working day.

Constraints (9) forbid a worker to work on a machine without supervision unless they are qualified for that machine on that working day. Constraints (10) indicate whether or not a worker is qualified to work on a machine independently on their first working day. Constraints (11) and (12) restrict the number of consecutive working days that a qualified worker can avoid working on a given machine and yet remain qualified for it. Constraints (13) ensure that a worker obtains the qualification to work on a machine without supervision only after sufficient training. The minimum number of required training days for this first qualification varies among workers and depends on their training level at the beginning of the planning horizon. By Constraints (14), each day a worker loses their qualification on a machine is captured and Constraints (15) ensure that upon losing the qualification this worker can become qualified on this machine again only after receiving the short retraining. Finally, Constraints (16)-(18) are binary and integer restrictions.

Note that the integer restrictions on $\mathbf{y}$ variables can be relaxed as the right hand sides of (7) and (8) are always integral and the coefficients of these variables are nonnegative in the minimization-type objective function. Given that n_{md} values are integer, a similar reasoning allows us to also relax the integrality restrictions on $\mathbf{z}$ variables as well, leading to an MIP variant. Additionally, the requirements satisfied by Constraints (4) and (5) can also be expressed as in Constraints (19) and (20), respectively, since an assigned personnel is either a trainee or a qualified worker. Utilizing (19) and (20) further enables replacing equality (3) with an inequality as in (21).

$$\sum_{p\in P:d\in D_p} w_{pmd} + \sum_{p\in P:d\in D_p} \tau_{pmd} \geq n_{md}, \qquad \forall m \in M^N,\ d\in D, \tag{19}$$

$$z_{md} + \sum_{p\in P:d\in D_p} w_{pmd} + \sum_{p\in P:d\in D_p} \tau_{pmd} \geq n_{md}, \qquad \forall m \in M^I,\ d\in D, \tag{20}$$

$$w_{pmd} + \tau_{pmd} \leq x_{pmd}, \qquad \forall p\in P, m\in M, d\in D_p, \tag{21}$$

These modifications lead us to the following MIP for the EMPATI:

$$\begin{aligned} (MIP) \qquad & \min\ (1) \\ & \text{s.t. } (2), (6)-(16), (19)-(21) \\ & 0 \leq y_{pd} \leq 1 && \forall p\in P, d\in D_p, && (22) \\ & z_{md} \geq 0 && \forall m\in M, d\in D. && (23) \end{aligned}$$

We obtain several valid inequalities by further analyzing the problem characteristics. These inequalities can be classified into two groups. The first group ensures that the binary variables corresponding to the following decision pairs cannot cannot take value one at the same time for a worker-machine combination:

Pair 1: lose qualification and be qualified on the same day (24)
Pair 2: lose qualification and work as a qualified worker on the same day (25)
Pair 3: lose qualification on a day and work on the previous day (26), (27)

$$\alpha_{pmd} + \gamma_{pmd} \leq 1, \qquad \forall p\in P, m\in M, d\in D_p, \tag{24}$$

$$\alpha_{pmd} + w_{pmd} \leq 1, \qquad \forall p\in P, m\in M, d\in D_p, \tag{25}$$

$$\alpha_{pmd} + \tau_{pme_{dp1}} + w_{pme_{dp1}} \leq 1, \qquad \forall p\in P, m\in M, d\in D_p, \tag{26}$$

$$\alpha_{pmd} + x_{pme_{dp1}} \leq 1, \qquad \forall p\in P, m\in M, d\in D_p. \tag{27}$$

Similarly, the second group ensures that the binary variables corresponding to the following decision pairs cannot take value one at the same time for a worker-machine combination:

Pair 1: train and be qualified on the same day (28)
Pair 2: train on a day and be qualified on the previous day (29)
Pair 3: train on a day and work as a qualified worker on the previous day (30)

$$\tau_{pmd} + \gamma_{pmd} \leq 1, \qquad \forall p\in P, m\in M, d\in D_p, \tag{28}$$

$$\tau_{pmd} + \gamma_{pme_{dp1}} \leq 1, \qquad \forall p\in P, m\in M, d\in D_p, \tag{29}$$

$$\tau_{pmd} + w_{pme_{dp1}} \leq 1, \qquad \forall p\in P, m\in M, d\in D_p. \tag{30}$$

3 Iterated local search

The MPATI is typically solved for a planning horizon of multiple months as the required training for a worker to qualify for a new machine typically takes several weeks. Preliminary experiments have shown that while integer programming may be used to find solutions when considering planning horizons of a few weeks, large-scale problems cannot be solved with available solvers. To address these problem instances, an Iterated Local Search (ILS) algorithm is introduced. Algorithm 1 outlines the main components of the proposed ILS, where S_0 is the initial solution and $f(S)$ is an evaluation function which returns the cost of solution S.

Algorithm 1: Iterated local search

Data: S_0, $f(S)$
Result: S
$S \leftarrow localSearch(S_0)$;
while *time limit not exceeded* **do**
  $S' \leftarrow perturb(S)$;
  $S' \leftarrow localSearch(S')$;
  if $f(S') \leq f(S)$ **then**
    $S \leftarrow S'$;
return S;

To obtain an initial solution S_0, a constructive heuristic first assigns one qualified worker to each machine in M^N. Then, randomly selected workers who are available are assigned until the minimum number of operators n_{md} is reached. Finally, any workers still available are assigned to the machines in M^I. These steps are repeated for each day in the planning horizon.

3.1 Solution representation

The proposed ILS operates on a direct solution representation consisting of a two dimensional matrix $S = (P \times D)$ whose values correspond to the machine assigned to worker $p \in P$ on day $d \in D$. To avoid feasibility issues, two hard constraints are relaxed during the search: machine staffing requirements (Constraints (4) and (5)) and trainee supervision (Constraints (6)). Violations of these constraints are penalized in the evaluation function $f(S)$. Possible compensation between the constraint penalties and the problem's real objective function (Equation (1)) is avoided by using a two-level lexicographic evaluation function. The first level sums all violations of the relaxed hard constraints while the second corresponds to the problem's original weighted sum function objective.

To identify violations of the trainee supervision constraint, an auxiliary datastructure is employed which maintains a training label $T_{pmd} = (l, c, \delta, s)$ for each worker $p \in P$, machine $m \in M^N$ and day $d \in D$. The label indicates the current skill level $l \in \{\text{none}, \text{trainee}, \text{qualified}\}$, the number of days c the worker has been level l, the previous day δ the worker was assigned to machine m and a boolean s which indicates whether or not the worker is eligible for a short retraining on this machine. Algorithm 2 sets the training labels for worker p on machine m from day d' onwards. The procedure updateWorking() adjusts the training label on day d after working an additional day on the machine. It is possible for a worker to transition from trainee to qualified if the required number of training days is reached. However, if the last time the worker was assigned to the machine was too long ago, that is, $d - \delta > C_p$, it is also possible to transition back to the trainee status. Similarly, updateIdle() sets the training label on day d when the worker is not assigned to the machine. All transitions are possible in this case, even to the *none* status when considering the situation where a worker was previously qualified but has lost it on this day.

Algorithm 2: Setting training labels

Data: p, m, d'
foreach $d \in D : d \geq d'$ **do**
  if $S(p, d) = m$ **then**
    $T_{pmd} \leftarrow$ updateWorking();
  else
    $T_{pmd} \leftarrow$ updateIdle();

As Algorithm 2 only updates the training labels from day d' onwards, delta evaluation of the trainee supervision constraint is straightforward to implement. Similarly, identifying violations of Constraints (4) and (5) and calculating the problem's real objective may be accelerated by only re-calculating the parts of the solution which have been changed.

3.2 Local search and perturbation

The main drivers of ILS' performance are the local search and perturbation procedures. In the local search performed at lines 1 and 4 in Algorithm 1, two parametrized neighborhood structures are used:

- *change*(k): assigns a worker to a machine for k consecutive days on which the worker is available,
- *swap*(k): swaps the assigned machines of two workers for k consecutive days on which these workers are available.

The local search procedure is instantiated with two variants of each neighborhood structure: one in which k is randomly chosen in each iteration from the interval $[1, 30]$ and another with $k = 1$. Rather than performing a full neighborhood search in each iteration, one new solution is randomly sampled and evaluated.

The perturbation procedure at line 3 in Algorithm 1 is used to escape from local optima reached by the local search procedure. This is achieved through a random walk of γ steps in the *change*(k) and *swap*(k) neighborhoods. Based on preliminary experiments, this parameter was set to $\gamma = 100$.

4 Computational study

In order to evaluate the performance of our methods under different planning settings, we randomly generated five sets of instances based on the real data provided by the company. In all instances, the interim cost $f = 264$, the switching cost $\pi = 60$, the number of workers $|P| = 40$ and the number of machines $|M| = 15$. Only three of these machines can be operated by interim workers, that is, $|M^I| = 3$. Each instance set corresponds to a fixed-length planning horizon, as shown in Table 1. For all instances in the same set, the number of days required for training is identical for all machine-personnel pairs, that is, $l_{pm} = \lambda$ for all $m \in M$ and $p \in P$. However, not every worker can be trained on a machine; this 0-1 (binary) parameter value is chosen randomly and may vary among instances of the same set. Similar to training, $C_p = C$ and $R_p = R$, $\forall p \in P$ in all instances of the same set. In addition to $|D|$, Table 1 reports the l, C and R values for each instance set.

Table 1 Planning horizon, training, skill memory and short retraining lengths (in days) for each instance set.

Instance set	Horizon $\|D\|$	Training λ	Skill memory C	Short training R
1	5	1	∞	0
2	10	2	5	1
3	20	4	10	2
4	60	12	20	3
5	260	60	20	3

Each set contains four instances which can be subdivided into two groups. For each group, a unique combination of random data generation probabilities is employed. These probability combinations imply that workers in the first group have fewer skills and less past training (yet to get the qualification) than those in the second group at the beginning of the planning horizon. Moreover, in the long run, for the first group fewer machines are available and a higher number of workers are needed.

We implement all mathematical models and the ILS algorithm in Java and use CPLEX 12.8 to solve the mathematical formulations. The experiments are run on a cluster of Intel Xeon CPU E5-2860 @2.50GHz with 24 cores and 64GB of RAM. The time limit for each run is set to five hours for each method. At most four threads are allowed for CPLEX runs. The ILS is run ten times per instance with different seed values for the algorithm's random number generator.

The first column of all the remaining tables in this section provides the instance code in format '$\#^1\_\#^2$' where $\#^1 = |D|$ indicates the length of the planning horizon and $\#^2$ is a unique identifier to associate the instance with the aforementioned groups and seeds. More specifically, instances with $\#^2 = 1, 2$ belong to the first group while those with $\#^2 = 3, 4$ belong to the second group. Columns 'Obj' provide the value of the solution obtained from the corresponding method whereas 'g%' indicates the gaps reported by CPLEX at the end of the time limit.

Table 2 Comparison of mathematical models with and without valid inequalities

		IP		MIP		IP + (24)-(30)		MIP + (24)-(30)	
Inst.	Best Obj	Obj	g%	Obj	g%	Obj	g%	Obj	g%
5_1	360	**360**	73.3	**360**	73.3	**360**	**72.8**	**360**	73.3
5_2	120	**120**	65.6	**120**	**50.0**	**120**	**50.0**	**120**	**50.0**
5_3	10044	**10044**	1.8	**10044**	1.8	**10044**	2.7	**10044**	**1.2**
5_4	8952	**8952**	6.0	**8952**	5.4	**8952**	**4.7**	**8952**	6.0
Avg.	4869.0	**4869.0**	36.7	**4869.0**	32.6	**4869.0**	**32.5**	**4869.0**	32.6
10_1	540	**540**	**65.4**	**540**	67.3	**540**	67.3	**540**	67.3
10_2	360	**360**	**68.6**	**360**	70.1	420	74.4	420	74.4
10_3	16344	**16344**	**17.3**	16644	18.8	**16344**	**17.3**	18084	25.2
10_4	12972	**12972**	22.2	**12972**	**21.7**	**12972**	22.7	**12972**	22.2
Avg.	7554.0	**7554.0**	**43.4**	7629.0	44.5	7569.0	45.4	8004.0	47.3
20_1	13044	**13044**	**94.8**	27912	97.6	41556	98.4	40152	98.3
20_2	12384	14700	96.7	34608	98.6	**12384**	**96.1**	44640	98.9
20_3	54084	**54084**	**45.2**	-	-	-	-	-	-
20_4	47292	**47292**	**39.8**	60108	52.7	54120	47.4	54120	47.4
Avg.	31701.0	**32280.0**	**69.1**	40876.0	83.0	36020.0	80.6	46304.0	81.6
Avg.	14708.0	14901.0	**49.7**	15692.7	50.7	14346.5	50.3	17309.5	51.3

Table 2 provides the results obtained from the two mathematical formulations with and without the valid inequalities presented in Section 2. These models are not tested on instances with a planning horizon longer than 20 days as they are too large for CPLEX to handle. The best solution value obtained among the four models is reported under column ‘Best Obj’. Although the MIP combinations terminate with a smaller dual gap for some instances, the IP provides the best gaps on average and the best solution values in all but one instance where the IP with the valid inequalities produces the best solution. We also observe that the gaps are larger for instances from the first group in each set.

Table 3 Comparing the ILS solution values with the best IP/MIP results

		Best IP/MIP		ILS		
Inst.	Best Obj	Obj	LB	Min Obj	Max Obj	Avg. Obj
5_1	360	**360**	97.9	**360**	**360**	**360.0**
5_2	120	**120**	60.0	**120**	**120**	**120.0**
5_3	10044	**10044**	9923.0	**10044**	**10044**	**10044.0**
5_4	8952	**8952**	8532.0	**8952**	**8952**	**8952.0**
Avg.	4869.0	**4869.0**	4653.2	**4869.0**	**4869.0**	**4869.0**
10_1	540	**540**	186.0	**540**	**540**	**540.0**
10_2	360	**360**	112.9	**360**	**360**	**360.0**
10_3	16344	**16344**	13524.0	**16344**	**16344**	**16344.0**
10_4	12972	**12972**	10152.0	**12972**	**12972**	**12972.0**
Avg.	7554.0	**7554.0**	5993.7	**7554.0**	**7554.0**	**7554.0**
20_1	1980	13044	681.9	**1980**	2040	2028.0
20_2	1800	12384	488.0	**1800**	1920	1890.0
20_3	35844	54084	29664.0	**35844**	**35844**	**35844.0**
20_4	37212	47292	28452.0	**37212**	**37212**	**37212.0**
Avg.	19209.0	31701.0	14821.5	**19209.0**	19254.0	19243.5
Avg.	10544.0	14708.0	8489.5	**10544.0**	10559.0	10555.5

In Table 3, we compare the solution values of the ILS with the best upper and lower bound values obtained from the mathematical formulations. The column ‘Best Obj’ reports the best solution value obtained from the four mathematical models and the ten runs of the ILS. The last three columns of this table provide the minimum, maximum and the average solution values from the ten runs of the ILS for each instance. The ILS is able to find a solution with the same objective value as the best model solution in all instances with $|D| = 5$, 10, and for the instances with $|D| = 20$ it provides a better solution than the best obtained by the formulations. Although the solution values may seem far from the lower bound values, by introducing the ILS solutions to CPLEX via a warm start function (addMIPStart) we are able to confirm that the heuristic solutions are in fact optimal for the instances with $|D| = 5$. Therefore, it is likely that the solutions provided by the ILS for larger instances are also optimal or near-optimal.

Table 4 Detailed results for all instances solved by the ILS approach

	ILS					
	Obj				Average number of	
Inst.	Min	Max	Avg,	Std.	Switches	Interims
5_1	360	360	360	0.00	6	0
5_2	120	120	120	0.00	2	0
5_3	10044	10044	10044	0.00	53	26
5_4	8952	8952	8952	0.00	48	23
10_1	540	540	540	0.00	9	0
10_2	360	360	360	0.00	6	0
10_3	16344	16344	16344	0.00	92	41
10_4	12972	12972	12972	0.00	93	28
20_1	1980	2040	2028	25.30	33.8	0
20_2	1800	1920	1890	42.43	31.5	0
20_3	35844	35844	35844	0.00	197	91
20_4	37212	37212	37212	0.00	233	88
60_1	7260	7560	7416	117.30	123.6	0
60_2	4500	4800	4644	110.27	77.4	0
60_3	109800	109800	109800	0.00	664	265
60_4	108192	108192	108192	0.00	646	263
260_1	37008	39828	38136	945.64	626.8	2
260_2	28680	30480	29730	476.03	495.5	0
260_3	471396	471516	471450	52.54	2823.9	1144
260_4	482040	482220	482112	61.97	2909.2	1165

Table 4 presents the detailed results obtained from the ILS when solving all instances. In addition to the minimum, maximum and average, this table also provides the standard deviation of the solution values of the ten runs in the fifth column. The solution values of the ten runs are identical for 60% of the instances. The average standard deviation across all instances is 112.3. The last two columns of this table provide the average number of switches and interim workers hired. We observe that the solutions for the first group of instances have fewer switches and interim workers compared to the second group. Although it is costly, it is worth noting that switching machines is unavoidable in order to obtain a feasible solution in these instances.

5 Conclusion

This paper addressed a new staff assignment problem where daily personnel requirements of different skills may vary over the planning horizon. This suggests varying duty assignments for the personnel which may eventually lead to losing skills if not used for too long. It is possible to train or retrain for different skills, however, these training decisions result in a very challenging problem for which even finding a feasible solution necessitates a significant amount of computational effort. Another challenging component penalizes switching duties as it requires an additional adjustment period and is undesirable by personnel. Together with the decisions concerning the number of interim workers to hire on a daily basis, reaching high quality solutions in a reasonable amount of time is only possible via tailored methods.

We introduced integer and mixed integer programming formulations with several valid inequalities. In our experiments, these formulations reached high quality solutions within a reasonable amount of time for problems with short planning horizons. However, they were unable to handle problems with long planning horizons with the commercial solver that was utilized. The solver also had difficulties in closing the dual gap, especially, for the first group of instances where the employees have fewer skills and less training at the beginning of the planning horizon. This motivated the development of an iterated local search algorithm to solve larger problem instances, which produced high quality solutions when solving the instances generated.

An interesting extension to the problem could be to include shift rostering as an additional decision as this would enable grouping operators with complementary skills in addition to cross-training them. Alternatively, the problem could be extended to enable training of the interim workers so that they can be allocated to the machines which require more complicated skills, more specifically, the non-interim machines. From a methodological perspective, it is worthwhile investigating the structural properties of the proposed mathematical models for developing possibly competitive matheuristic approaches.

Acknowledgments

This research was supported by Dynamical Combinatorial Optimization (KU Leuven C2 project C24/17/012) and Data-driven logistics (FWO-S007318N). Editorial consultation provided by Luke Connolly (KU Leuven).

References

1. Azzouz, A., Ennigrou, M., Ben Said, L.: Scheduling problems under learning effects: classification and cartography. International Journal of Production Research **56**(4), 1642–1661 (2018)
2. Biskup, D.: A state-of-the-art review on scheduling with learning effects. European Journal of Operational Research **188**(2), 315–329 (2008)
3. Brusco, M.J., Johns, T.R.: Staffing a multiskilled workforce with varying levels of productivity: An analysis of cross-training policies. Decision Sciences **29**(2), 499–515 (1998)
4. De Bruecker, P., Beliën, J., Van den Bergh, J., Demeulemeester, E.: A three-stage mixed integer programming approach for optimizing the skill mix and training schedules for aircraft maintenance. European Journal of Operational Research **267**(2), 439–452 (2018)
5. De Bruecker, P., Van den Bergh, J., Beliën, J., Demeulemeester, E.: Workforce planning incorporating skills: State of the art. European Journal of Operational Research **243**(1), 1–16 (2015)
6. Ernst, A.T., Jiang, H., Krishnamoorthy, M., Sier, D.: Staff scheduling and rostering: A review of applications, methods and models. European Journal of Operational Research **153**(1), 3–27 (2004)
7. Guerry, M.A., De Feyter, T., Berghe, G.V., et al.: The roster quality staffing problem–a methodology for improving the roster quality by modifying the personnel structure. European Journal of Operational Research **230**(3), 551–562 (2013)
8. López, C.E., Nembhard, D.: Cooperative workforce planning heuristic with worker learning and forgetting and demand constraints. In: Proceedings of the 2017 Industrial and Systems Engineering Conference, pp. 380–385 (2017)
9. Nembhard, D.A., Bentefouet, F.: Parallel system scheduling with general worker learning and forgetting. International Journal of Production Economics **139**(2), 533–542 (2012)
10. Van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., De Boeck, L.: Personnel scheduling: A literature review. European Journal of Operational Research **226**(3), 367–385 (2013)
11. Wirojanagud, P., Gel, E.S., Fowler, J.W., Cardy, R.: Modelling inherent worker differences for workforce planning. International Journal of Production Research **45**(3), 525–553 (2007)