RAIRO-Oper. Res. 49 (2015) 393–412
DOI: 10.1051/ro/2014052

RAIRO Operations Research
www.rairo-ro.org

A CLASSIFICATION SCHEME FOR INTEGRATED STAFF ROSTERING AND SCHEDULING PROBLEMS

Mareike Paul[1] and Sigrid Knust[1]

Abstract. In the last decades job scheduling, staff rostering and staff assignment have received considerable attention, as have combinations of these problems. However, given the wide range of variants of all three basic problems, the number of combinations is immense. In this paper we introduce a new classification scheme for integrated staff rostering and job scheduling problems, extending existing schemes for project and machine scheduling. We provide some elementary reductions and show how problems studied in the literature fit into this new classification scheme. Furthermore, some complexity results are presented.

Keywords. Scheduling, rostering, assignment, staff, classification scheme, complexity.

Mathematics Subject Classification. 90B35, 68Q25.

1. Introduction

Companies often have to schedule jobs, roster their staff and assign employees to jobs in such a way that all jobs can be executed and the employees receive rosters they like. Additionally, several constraints have to be respected taking into account demands of the company, legal restrictions, wishes of the employees, *etc.* Since usually the whole problem is very complex, often decomposition approaches are applied. In such approaches the problem is tackled in different stages. One possibility is to schedule the jobs first and to handle staff scheduling and assignment

Received February 9, 2013. Accepted September 26, 2014.

[1] Institute of Computer Science, University of Osnabrück, 49069 Osnabrück, Germany. mareike.paul@uni-osnabrueck.de;sigrid.knust@uni-osnabrueck.de

Article published by EDP Sciences

according to the fixed job schedule afterwards. Another possibility is to determine a timetable for the staff first and to schedule the jobs according to the fixed staff timetable afterwards. A third way is to consider staff and job scheduling in an integrated way, *i.e.* to deal with all three problems simultaneously.

In *job scheduling problems* (*cf.* for example Błażewicz *et al.* [11]), the task is to determine start times for given jobs such that certain constraints are satisfied and an objective function is optimized. The most common constraints deal with the availability of resources (*e.g.* machines). For surveys on resource constrained project scheduling problems (RCPSP) see Hartmann and Briskorn [31], Brucker *et al.* [13] or Brucker and Knust [14].

Rostering problems try to assign employees to given shifts in accordance to different constraints. Constraints may be minimum demands for employees required for a particular shift, unavailabilities of employees, or contractual and legal restrictions such as a minimum rest time between two consecutive shifts. Burke *et al.* [20], Ernst *et al.* [24] and Van den Bergh *et al.* [43] give overviews of rostering problems.

Assignment problems combine the results of both scheduling and staffing. Given is a set of employees with availabilities due to fixed staff rosters and different qualifications. Furthermore, there are jobs with given start and completion times having minimum demands for employees with certain qualifications. The task is to find assignments of employees to jobs according to their qualifications such that all demands are fulfilled and each employee is assigned to at most one job at a time.

Since usually a lot of different constraints have to be respected, the number of combined problems is rather high. Hence, a classification scheme may be useful to describe problems in a standard way and to classify existing literature. Furthermore, such a scheme makes it easier to draw the borderline between polynomially solvable and $\mathcal{NP}$-hard problems.

Classification schemes have been proposed for different kinds of optimization problems. For example, Desrochers *et al.* [22] provide a scheme for vehicle routing and scheduling problems, Boysen *et al.* [12] give a classification of assembly line balancing. For machine scheduling a scheme was published by Graham *et al.* [28] and for example extended by Błażewicz *et al.* [10]. Classification schemes for project scheduling were proposed by Brucker *et al.* [13] and by Herroelen *et al.* [34]. For rostering problems, Causmaecker *et al.* [21] suggested a classification scheme, but as its entries describe categories of constraints and not the constraints themselves, this scheme is not regarded in this paper. All of these classification schemes use a three-field notation. In this paper we introduce a notation with five fields for classifying integrated problems.

The paper is structured as follows. In Section 2, we introduce the used notations as well as the constraints for a basic integrated staff rostering and scheduling problem. Section 3 presents the classification scheme including elementary reductions. In Section 4, we give a literature review, in Section 5 complexity results are presented. The paper concludes with some remarks.

2. Problem formulation

In this section we introduce a basic integrated staff rostering and scheduling problem as well as the used notations.

Let $\mathcal{E} = \{1, \ldots, E\}$ be a given set of employees and $\mathcal{Q} = \{1, \ldots, Q\}$ the set of their qualifications. Each employee e possesses Q_e qualifications, summarized in a subset $\mathcal{Q}_e \subseteq \mathcal{Q}$, and all employees with qualification q are combined in a set $\mathcal{E}_q$. Let $\mathcal{J} = \{1, \ldots, J\}$ be a given set of jobs. Each job $j \in \mathcal{J}$ must be executed for p_j time units within a given time horizon $\mathcal{T} = \{1, \ldots, T\}$. While being executed, job j requires at least b_{jq} employees possessing qualification $q \in \mathcal{Q}$. By m_j we denote the number of different qualifications job j requires. We generally assume that preemption of jobs is not allowed, but we allow *transition*. Transition means that an employee can change jobs at any time and can be replaced at a job without loss of time.

The time horizon $\mathcal{T}$ is divided into D days $d \in \mathcal{D} = \{1, \ldots, D\}$ and into S shifts $\sigma \in \mathcal{S} = \{1, \ldots, S\}$. Each shift σ is characterized by its starting time s_σ, its ending time e_σ and its length $l_\sigma := e_\sigma - s_\sigma$. We assume that each shift is assigned to exactly one day. The shifts of day $d \in \mathcal{D}$ are combined in the set $\mathcal{S}_d$.

The objective of the integrated staff rostering and job scheduling problem is to find a schedule for the jobs as well as a feasible assignment of employees to shifts and jobs. An assignment of employees to shifts is called a *staff roster*, an assignment of starting times to jobs is called a *schedule*, and an *assignment* of employees to jobs and qualifications determines which employee executes which job with which of his qualifications.

Such assignments have to respect the following basic constraints:

(1) Each job must be processed for its required processing time.
(2) For each job its demands for employees with certain qualifications must be fulfilled.
(3) Each employee can be assigned to at most one shift per day.
(4) The qualifications of employees have to be respected, *i.e.* each employee can only be scheduled for one qualification he offers.
(5) Each employee can only be assigned to at most one job at a time and only in shifts he is assigned to.

Additional constraints occuring in the literature will be discussed in Section 3.

3. Classification scheme

In this section, we introduce a scheme for classifying integrated staff rostering and job scheduling problems based on the basic model from the previous section. Additionally, we discuss further constraints studied in the literature. In this scheme we use the existing scheduling classification schemes to describe the scheduling environment. The remaining parts of the scheme are the *rostering and* *assignment environment*, parameters considering *employees and qualifications*, parameters concerning the *shifts*, and the *objective function for the rostering and assignment problem*.

The proposed classification scheme consists of a tuple $(\alpha)+(\beta \mid \gamma \mid \delta \mid \varepsilon)$, where

- α describes the scheduling problem including a scheduling objective function;
- β gives a description of the rostering and assignment environment;
- γ describes parameters for employees and qualifications;
- δ describes parameters for the shifts; and
- ε gives the objective function of the rostering and assignment part.

Both fields γ, δ may consist of more than one entry. Each entry can be the empty symbol ($\circ$) which can be omitted. If both parts γ and δ only consist of empty symbols, $\beta \mid \circ \mid \ \circ \mid \varepsilon$ can be reduced to $\beta \ \| \ \varepsilon$. Similarly, $\beta \mid \circ \mid \delta$ can be reduced to $\beta \mid \delta$ and $\gamma \mid \circ \mid \varepsilon$ to $\gamma \mid \varepsilon$.

The α-field is used as in known scheduling classification schemes. If the job schedule is already fixed and has not to be calculated, the part $(\alpha)+$ is omitted. In the following we describe the five fields $\alpha, \beta, \gamma, \delta, \varepsilon$ in more detail.

3.1. The α-field: Scheduling part

The α-field gives information about the scheduling part in the integrated problem. For this, we use existing classification schemes, *e.g.* the classification schemes by Graham *et al.* [28] for machine scheduling or Brucker *et al.* [13] for project scheduling. Both classification schemes consist of three fields $\alpha' \mid \beta' \mid \gamma'$ with a description of the environment (machines or resources) in α', a job parameter description in β' and the objective function in γ'. Examples (which also occur in Tab. 1 in Sect. 4) are:

- $PS \mid prec \mid C_{\max}$ denotes the classical RCPSP where (renewable) resources have to be considered, precedence constraints between the jobs may be given, and the objective is to minimize the makespan.
- $PS \mid r_j, d_j \mid -$ denotes the feasibility problem of the RCPSP taking into account given time windows $[r_j, d_j]$ for each job j. The objective is to find a feasible RCPSP schedule respecting the time windows and resource constraints.
- $PS, \infty \mid pmtn, r_j, d_j \mid -$ denotes the same problem but without any resource constraints (PS, ∞ indicates that there are sufficient resources). Additionally, preemption of the jobs is allowed.
- $J \,\|\, C_{\max}$ denotes the classical job-shop problem minimizing the makespan.
- $\cdot \mid r_j, tmp \mid -$ denotes a problem with dedicated machines (*i.e.* each job has to be processed on a preassigned machine) and release dates r_j. Additionally, tmp means that general timing constraints in form of time-lags d_{ij} for the differences between the starting times of jobs j and i have to be respected. The objective is to find a feasible solution satisfying all release dates and time-lags.

In addition to the usual scheduling entries, we use two further ones in connection with employees: the entry $p_j(e)$ denotes that the durations p_j of the jobs are not fixed, but depend on the executing employees. The entry s^e_{jk} denotes the presence of setup times between jobs j, k which are consecutively executed by the same employee e.

3.2. The β-field: Rostering and assignment environment

The description of the environment $\beta \in \{A, R, RA\}$ gives information about the kind of problem considered apart from scheduling:

- $\beta = A$ denotes job assignment problems with given staff rosters and given start times for the jobs, *i.e.* only employees have to be assigned to jobs.
- $\beta = R$ denotes staff rostering problems without any jobs, *i.e.* only the assignment of employees to shifts has to be done.
- $\beta = RA$ denotes integrated staff rostering and job assignment problems.

3.3. The γ-field: Employees and qualifications

The first parameter $\gamma_1 \in \{Q = 1, Q_e = 1, \circ\}$ gives information about the number of qualifications and their distribution to the employees:

$$\gamma_1 = \begin{cases} Q = 1 & \text{the total number of qualifications is equal to 1,} \\ Q_e = 1 & \text{each employee has only one qualification,} \\ \circ & \text{arbitrary distribution of qualifications to employees.} \end{cases}$$

The parameters γ_2 and γ_3 give further information about the *demands* of the jobs. Each job may require several qualifications or just a single one. For the correponding numbers m_j we distinguish between the following alternatives:

$$\gamma_2 = \begin{cases} m_j = 1 & \text{each job requires only one qualification,} \\ \circ & \text{jobs may require arbitrary numbers of qualifications.} \end{cases}$$

The parameter γ_3 considers the demands of jobs for employees with qualification q.

$$\gamma_3 = \begin{cases} b_{jq} \leq 1 & \text{all demands of jobs are at most 1,} \\ b_{jq}(t) & \text{the demands of the jobs are time dependent,} \\ \circ & \text{the demands are arbitrary integers, but not time dependent.} \end{cases}$$

In the basic model presented in Section 2, we assume that every employee can be replaced by any other with the same qualification when performing a job (transition allowed). If this assumption is omitted, two variants are common. In the first variant, an employee assigned to a job may only be replaced by another if his shift ends. The second (more strict) assumption forbids changes of the executing employees at all (*i.e.* an employee may only be assigned to a job if he is available for the whole processing time). The parameter $\gamma_4 \in \{\text{notrans}, \text{trans}(\sigma), \circ\}$

distinguishes these situations:

$$\gamma_4 = \begin{cases} \text{notrans} & \text{no changes of employees during the execution of jobs are allowed,} \\ \text{trans}(\sigma) & \text{transition is only allowed at the end of shifts,} \\ \circ & \text{transition is allowed.} \end{cases}$$

It may occur that employees are *unavailable* in some shifts due to vacations, consultations, or skill enhancement. In this case sets U_e are defined, containing all shifts σ in which employee e is unavailable (*i.e.* cannot be assigned to any shift in U_e). For denoting the presence of unavailabilities the parameter $\gamma_5 \in \{U_e, \circ\}$ is used where

$$\gamma_5 = \begin{cases} U_e & \text{employees may be unavailable for certain shifts,} \\ \circ & \text{unavailabilities of employees are not considered.} \end{cases}$$

Due to their contracts employees may have *maximum* and *minimum working times* H_e^+, H_e^- over the whole time horizon. We introduce parameters $\gamma_6 \in \{H_e^+,\ \circ\}$ and $\gamma_7 \in \{H_e^-,\ \circ\}$ with:

$$\gamma_6 = \begin{cases} H_e^+ & \text{maximum working times for the employees have to be respected,} \\ \circ & \text{maximum working times are not considered.} \end{cases}$$

$$\gamma_7 = \begin{cases} H_e^- & \text{minimum working times for the employees have to be respected,} \\ \circ & \text{minimum working times are not considered.} \end{cases}$$

Figure 1 shows elementary (polynomial-time) reductions between the entries of the parameters γ_1 to γ_7. In this graph there is an arc between two possible entries γ_i', γ_i'' of one parameter γ_i if γ_i' reduces to γ_i'' (*i.e.* γ_i' is a special case of γ_i''). Note that the entries for γ_1 and γ_2 were joined in the first graph since the combination $Q = 1, m_j = 1$ is superfluous (if only one qualification exists, all jobs need this qualification).

From the reduction graphs and known complexity results we can derive further complexity results. If on the one hand, a problem is polynomially solvable, also all special cases are polynomially solvable. If on the other hand, a problem is $\mathcal{NP}$-hard, then also all generalizations according to the reduction graphs are $\mathcal{NP}$-hard. Hence, the reduction graphs help in finding generalizations and special cases.

3.4. The δ-field: Information on the shifts

In some rostering problems a set of feasible *shift patterns* is given. A shift pattern is a sequence of shifts determining the working shifts and the days off for one employee for the whole time horizon. These patterns usually have to respect some constraints, *e.g.* that they contain at most one shift per day. Each employee must be assigned to one of these shift patterns with respect to the demands of shifts and employee dependent factors, *e.g.* his qualifications, minimum or maximum

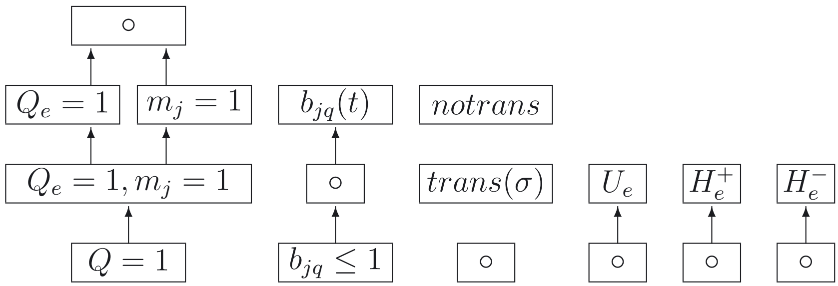

FIGURE 1. Reduction graphs for parameters γ_1 to γ_7.

working times or unavailabilities. Whether such shift patterns are considered or not, is stated by $\delta_1 \in \{shp, \ \circ\}$ with

$$\delta_1 = \begin{cases} shp & \text{shift patterns have to be assigned to the employees,} \\ \circ & \text{no shift patterns are considered.} \end{cases}$$

Two shifts $\sigma, \sigma' \in S$ are called *overlapping* if $(s_\sigma, e_\sigma] \cap (s_{\sigma'}, e_{\sigma'}] \neq \emptyset$. The parameter $\delta_2 \in \{\textit{no-ol}, \circ\}$ states whether shifts may *overlap* or not:

$$\delta_2 = \begin{cases} \textit{no-ol} & \text{shifts may not overlap,} \\ \circ & \text{shifts may overlap.} \end{cases}$$

In some companies, employees are not allowed to work in two specific shifts on consecutive days. For example, an employee working in a late shift of one day is not allowed to work in a morning shift on the following day. More generally, sets may be given containing pairs of shifts which may not be assigned to the same employee. Such restrictions are called *forbidden shift changes*. We use the parameter $\delta_3 \in \{fch, \ \circ\}$ with

$$\delta_3 = \begin{cases} fch & \text{forbidden shift changes have to be taken into account,} \\ \circ & \text{forbidden shift changes are not considered.} \end{cases}$$

Parameter δ_4 gives information about the *maximum number of shifts* per day.

$$\delta_4 = \begin{cases} s_{\mathrm{d}} \leq f & \text{the number of shifts per day is at most } f, \\ \circ & \text{no limits on the numbers of shifts per day are considered.} \end{cases}$$

Another information about shifts is their *length*. For this parameter, we distinguish between the case that all shifts have the same length and the case that all shifts have arbitrary lengths. This information is provided by the parameter $\delta_5 \in \{l_\sigma = l, \circ\}$ with

$$\delta_5 = \begin{cases} l_\sigma = l & \text{all shifts } \sigma \text{ have the same length } l, \\ \circ & \text{the shifts have arbitrary lengths.} \end{cases}$$

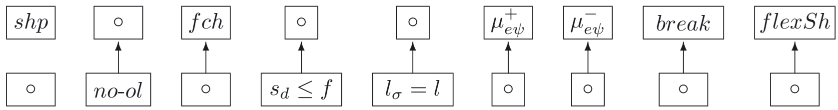

FIGURE 2. Reduction graphs for parameters δ_1 to δ_9

In most companies, shifts are separated into shifts of different types ψ (*e.g.* morning, late and night shifts), so-called *shift types*. Due to fairness aspects or contract regulations, the number of shifts of one type assigned to an employee in the planning horizon may be restricted (by upper or lower bounds). Parameters $\delta_6 \in \{\mu^+_{e\psi},\ \circ\}$ and $\delta_7 \in \{\mu^-_{e\psi},\ \circ\}$ are introduced with

$$\delta_6 = \begin{cases} \mu^+_{e\psi} & \text{UBs for number of shifts per type and employee are given,} \\ \circ & \text{no UBs for number of shifts per type and employee are given,} \end{cases}$$

$$\delta_7 = \begin{cases} \mu^-_{e\psi} & \text{LBs for number of shifts per type and employee are given,} \\ \circ & \text{no LBs for number of shifts per type and employee are given.} \end{cases}$$

Associated with each shift, $\sigma \in \mathcal{S}$ may be a set of *breaks* in which the employees working in shift σ are not available for executing jobs. We introduce a parameter $\delta_8 \in \{\text{break},\ \circ\}$ to indicate the consideration of breaks.

$$\delta_8 = \begin{cases} \text{break} & \text{breaks have to be considered,} \\ \circ & \text{breaks are not considered.} \end{cases}$$

In some companies, the shifts of employees depend on the assigned jobs and their starting and ending times are not fixed beforehand. In this case the shifts are called *flexible*, indicating that their starting and ending times have to be determined. We have the parameter $\delta_9 \in \{\text{flexSh},\ \circ\}$ with

$$\delta_9 = \begin{cases} \text{flexSh} & \text{shifts are flexible,} \\ \circ & \text{starting and ending times of all shifts are fixed.} \end{cases}$$

Elementary reductions between the entries of parameters δ_1 to δ_9 are shown in Figure 2.

3.5. THE ε-FIELD: THE OBJECTIVE FUNCTION

The parameter ε denotes the objective function concerning the rostering and assignment part and sometimes also its relation to the objective function of the scheduling part.

Examples for studied objective functions are:

- $-$: only the feasibility problem has to be solved. If there is a scheduling part in α and an objective function f is defined for this part, then the function f is the objective of the whole problem.

- $AC(e, \sigma)$, $AC(e, \sigma, \sigma')$, $AC(e, j, q)$, $\ldots$: the objective is to minimize assignment costs (AC) for employees and shifts, for employees and shift changes of consecutive days, for employees, jobs and qualifications, *etc.* This denotes the minimization of the sum of all costs for assigning an employee e to a shift σ *etc.* In this context e means employees, j means jobs, σ, σ' mean shifts, q means qualifications, t means time periods, and π means shift patterns.
- $C_{\text{overstaff}}, C_{\text{understaff}}$: the objective is to minimize costs for overstaffing or understaffing of the jobs (or shifts). In this situation desired numbers of staff for jobs (or shifts) are given and the deviations from these values have to be minimized.
- $C_{\text{overtime}}, C_{\text{undertime}}$: the objective is to minimize costs for overtime or undertime of the employees. In this situation, a desired working time for each employee is given and the deviations from these values have to be minimized.
- $C_{\text{use(e)}}$: the objective is to minimize the staff needed.
- C_{ext}: the objective is to minimize costs for hiring external staff if the given employees can not execute all jobs.
- C_{idle}: the objective is to minimize costs for idle time of employees, *i.e.* the time an employee is working due to the rosters, but having no job to perform.
- C_{travel}: the objective is to minimize travel costs which arise if two jobs executed by the same employee take place at different locations. Often they are related to travel times, which may be indicated as setup times s^e_{jk} in the scheduling part.
- $f(AC(e, \pi), \cdot)$: the objective is to minimize the value of a function f including assignment costs $AC(e, \pi)$ and the scheduling objective function from α, indicated by the $\cdot$.
- $Lex(\cdot, AC(e, \pi))$: the lexicographic objective is to first minimize the objective of the scheduling part, indicated by the $\cdot$, and afterwards to minimize the assignment costs $AC(e, \pi)$.

4. Literature review on integrated problems

In this section, we classify existing literature for integrated problems in the proposed scheme. In Table 1, the authors, the problem and the used methods are listed using the abbreviations from Table 2. In the first part of Table 1, integrated scheduling and assignment problems are presented, while the second part lists papers considering rostering and assignment problems. The third part contains integrated scheduling, rostering and assignment problems. Within each part, the papers are ordered according to the authors.

The papers listed in the first section all deal with scheduling and assignment problems. Roberts and Escudero [41] tackle a rather simple form of the problem. A set of jobs has to be assigned to a set of employees, each with one qualification, minimizing the idle time of employees. In this problem, the jobs have time-dependent demands and are broken down to one-hour jobs. Drexl [23] slightly changes this model. He makes two different assumptions: first he assumes that

Table 1. Summary of literature dealing with integrated problems.

Authors	Problem	Method
Bellenguez and Neron [7]	$(PS,\infty \mid prec \mid C_{\max})$ $+(A \mid \text{notrans} \mid s_d = 1, l_\sigma = 1 \mid -)$	LB
Bellenguez and Neron [8]	$(PS,\infty \mid prec \mid C_{\max})$ $+(A \mid \text{notrans} \mid s_d = 1, l_\sigma = 1 \mid -)$	B&B
Brucker and Qu [17]	$(PS,\infty \mid pmtn, r_j, d_j \mid -)$ $+(A \mid m_j = 1, b_{jq} \leq 1 \mid f(C_{\text{overtime}}, AC(e,t)))$	NF
Drexl [23]	$(PS,\infty \mid prec, p_j(e) \mid -)$ $+(A \mid Q = 1, b_{jq} \leq 1, \text{notrans}, H_e^+ \mid s_d \leq 1, l_\sigma = 1 \mid AC(e,j))$	H,DP,B&B
Ferreira and Bazzan [25]	$(PS,\infty \mid prec \mid C_{\max})$ $+(A \mid \text{notrans} \mid s_d \leq 1, l_\sigma = 1 \mid -)$	H
Heimerl and Kolisch [32]	$(PS,\infty \mid r_j, d_j \mid -)$ $+(A \mid m_j = 1 \mid f(C_{\text{ext}}, AC(t,e,j,q))$	MIP
Roberts and Escudero [41]	$(PS,\infty \mid p_j = 1 \mid -)$ $+(A \mid Q_e = 1, b_{jq}(t) \mid C_{\text{idle}})$	IP
Awad and Chinneck [6]	$RA \mid \text{trans}(\sigma), U_e \mid s_d \leq 6 \mid f(C_{\text{ext}}, AC(e,\sigma), AC(e,j), AC(e,\sigma,\sigma'), C_{\text{overstaff}})$	H + GA
Brunner *et al.* [18]	$RA \mid \text{trans}(\sigma) \mid \text{flexSh} \mid f(C_{\text{overtime}}, AC(e,t), C_{\text{ext}})$	MIP
Brunner *et al.* [19]	$RA \mid \text{trans}(\sigma), U_e \mid \text{break}, \text{flexSh} \mid f(C_{\text{ext}}, AC(e,j,q,t), C_{\text{overtime}})$	B&P
Kilby [35]	$RA \mid m_j = 1, b_{jq} \leq 1, H_e^+ \mid no\text{-}ol, fch, s_d \leq 3 \mid f(C_{\text{undertime}}, C_{\text{use}(e)}, AC(e,\sigma,\sigma'))$	H
Loucks and Jacobs [40]	$RA \mid \text{flexSh} \mid f(C_{\text{untertime}}, C_{\text{overtime}}, C_{\text{overstaff}})$	H
Alfares *et al.*[1]	$(PS,\infty \mid prec \mid C_{\max})$ $+(RA \mid Q = 1 \mid shp, no\text{-}ol, s_d \leq 1, l_\sigma = 1 \mid f(AC(e,\pi),\ \cdot\ ))$	DP
Alfares *et al.* [2]	$(PS,\infty \mid prec, p_j(e) \mid C_{max})$ $+(RA \mid Q_e = 1 \mid shp, s_d \leq 1, l_\sigma = 1 \mid f(AC(e,\pi,q),\ \cdot\ ))$	H
Artigues *et al.* [4]	$(\cdot \mid r_j, d_j, tmp \mid -)$ $+(RA \mid no\text{-}ol \mid f(AC(j,t), AC(e,q,t)))$	CP + LP
Artigues *et al.* [5]	$(J \mid\mid C_{max})$ $+(RA \mid \text{trans}(\sigma), U_e \mid no\text{-}ol, fch, l_\sigma = l \mid Lex(\cdot, AC(e,q,\sigma)))$	CP + LP
Bertels and Fahle [9]	$\left(PS,\infty \mid r_j, d_j, s^e_{jk} \mid f(AC(j,t)\right)$ $+ (RA \mid m_j = 1, b_{jq} \leq 1, \text{notrans}, U_e, H_e^+, H_e^- \mid \text{flexSh} \mid f(C_{\text{overtime}}, AC(e,t), AC(e,j), C_{\text{travel}}, \cdot))$	H
Guyon *et al.* [29]	$(PS,\infty \mid pmtn, r_j, d_j \mid -)$ $+(RA \mid m_j = 1, b_{jq} \leq 1 \mid shp \mid AC(e,\pi))$	D,D + CG
Guyon *et al.* [30]	$(J \mid\mid -)$ $+(RA \mid m_j = 1, b_{jq} \leq 1, \text{trans}(\sigma), U_e \mid no\text{-}ol, fch, s_d \leq 3, l_\sigma = l \mid AC(e,q,\sigma))$	D + CG, CG + B&B
Herbers [33]	$\left(PS,\infty \mid r_j, d_j, s^e_{jk} \mid -\right)$ $+(RA \mid \text{trans}(\sigma) \mid \mu^+_{e\psi}, \mu^-_{e\psi}, \text{break} \mid AC(e,\sigma))$	D + B&P

TABLE 2. Abbreviations used for the methods in Table 1.

B&B	=	Branch-and-bound	GA	=	Genetic algorithm
B&P	=	Branch-and-price	H	=	Heuristic approach
CP	=	Constraint programming	LB	=	Lower bounds
CG	=	Cut generation	LP	=	Linear programming
D	=	Decomposition	(M)IP	=	(Mixed-) Integer programming
DP	=	Dynamic programming	NF	=	Network flow problem

each jobs requires only one employee, and second that the duration of a job depends on the employee assigned to the job. Bellenguez and Neron [7] tackle the multi-skill project scheduling problem with hierarchical levels of skills. For each employee not only a set of qualifications is given, but also the level of his experience. They introduce two destructive lower bounds, one of them based on energetic reasoning. A similar problem is considered by Ferreira and Bazzah [25]. They call this problem the distributed RCPSP and apply a swarm intelligence approach to the problem. Bellenguez and Neron [8] investigate another kind of the multi-skill project scheduling problem. For this problem they present a branch-and-bound algorithm and use two destructive lower bounds, one of them again based on energetic reasoning. Heimerl and Kolisch [32] schedule and assign jobs with time windows. Additionally, they allow the use of external staff to meet all demands. Brucker and Qu [17] deal with a preemptive scheduling problem. They extend a network flow model for integrated scheduling and assignment problems with one qualification to the case of several qualifications.

In the second part of the table, integrated rostering and assigment problems are listed. All problems investigated are very different, reaching from problems with fixed shifts and one qualification to problems with flexible shifts and several qualifications which are arbitrarily distributed. Loucks and Jacobs [40] tackle a problem with hardly any constraints, but with flexible shifts. For this problem they present a heuristic consisting of several steps. Awad and Chinneck [6] investigate the problem of assigning proctors to fixed exams. Within their problem employees can carpool, which has to be respected in the rostering part. For solving this problem, a genetic algorithm in combination with another heuristic is proposed. Kilby [35] presents an algorithm based on augmented regrets, which he applies on a rostering and assignment problem with a large number of constraints. Brunner *et al.* [18, 19] deal with the rostering and assignment of physicians. In both problems, the shifts are flexible; in [19] additionally breaks have to be considered. For solving these problems, a MIP-formulation [18] and a branch-and-price algorithm [19] were proposed.

The last section of the table lists integrated scheduling, rostering and assignment problems. Alfares *et al.* [1] published a paper dealing with an integrated problem with shift patterns. Each day consists of one shift with a length of one time period, and the employees are equally skilled. A similar problem is tackled by Alfares *et al.* [2]. The main differences are the use of labour classes, where each employee

TABLE 3. Known complexity results for pure assignment or rostering problems.

Problem	Complexity	Ref.	Reduction/model
$A \mid m_j = 1 \mid no\text{-}ol, s_d \leq 1 \mid AC(e,j)$	$\mathcal{P}$	[16]	transshipment problem
$A \mid Q = 1 \mid s_d \leq S \mid -$	$\mathcal{P}$	[16, 42]	min cost flow problem
$A \mid Q = 1, b_{jq} \leq 1, \text{notrans} \mid no\text{-}ol, s_d \leq 1 \mid C_{use(e)}$	$\mathcal{P}$	[38]	min cost flow problem
$A \mid m_j = 1, b_{jq} \leq 1, \text{notrans} \mid no\text{-}ol, s_d \leq 1 \mid -$	$\mathcal{NP}$-hard	[3]	3-SAT
$A \mid m_j = 1, b_{jq} \leq 1, \text{notrans} \mid C_{\text{use}(e)}$	$\mathcal{NP}$-hard	[36]	circular arc coloring
$A \mid m_j = 1, b_{jq} \leq 1, \text{notrans} \mid no\text{-}ol, s_d \leq 1 \mid C_{\text{use}(e)}$	$\mathcal{NP}$-hard	[38]	3D-matching
$A \mid m_j = 1, b_{jq} \leq 1 \mid no\text{-}ol, s_d \leq 1 \mid C_{\text{use}(e)}$	$\mathcal{NP}$-hard	[38]	3D-matching
$R \mid Q = 1 \mid fch \mid -$	$\mathcal{P}$	[37]	min cost flow problem
$R \mid fch \mid -$	$\mathcal{NP}$-hard	[39]	3-SAT
$R \mid Q = 1, b_{\sigma q} \leq 1 \mid shp, s_d \leq 1, l_\sigma = 1 \mid C_{\text{use}(e)}$[2]	$\mathcal{NP}$-hard	[16, 26]	exact covering by 3-sets

possesses one of several qualifications, and that the durations of the jobs depend on the executing employee. Another paper dealing with an integrated problem with given shift patterns was published by Guyon *et al.* [29]. In their problem, the jobs additionally have time windows and preemption is allowed. Time windows for jobs are also considered by Herbers [33]. Additionally, upper and lower bounds for the number of shifts per type and employee are introduced. He additionally considers breaks and groups of employees (called crews), which should be scheduled as parallel as possible. Bertels and Fahle [9] deal with the home health care problem, *i.e.* the problem of assigning nurses to jobs at different clients taking into account travel times. Additionally, hard and soft time windows are given; the shifts are flexible. Artigues *et al.* [4] tackle a problem with dedicated machines and arbitrary time-lags between jobs. Artigues *et al.* [5] consider the job-shop problem as the underlying scheduling environment. Additional constraints concern forbidden shift changes and the restriction of non-overlapping shifts. All these assumptions are also present in Guyon *et al.* [30] with the additional constraint of unavailabilities of employees.

5. Complexity results

In this section, we present some complexity results. Known complexity results for pure assignment or rostering problems are summarized in Table 3. Complexity results for pure machine scheduling problems can be found at the website [15]. If a pure problem is already $\mathcal{NP}$-hard, an integrated problem containing it as a subproblem is also $\mathcal{NP}$-hard. On the other hand, polynomially solvable special cases may be useful in decomposition algorithms for integrated problems.

The results for assignment problems with $m_j = 1, b_{jq} \leq 1$ and the objective function $C_{\text{use}(e)}$ are derived from the class of fixed interval scheduling problems (see Kolen *et al.* [36]). In these problems, jobs with fixed start and completion

[2]Here, $b_{\sigma q}$ denotes the demand of shift σ for employees with qualification q (only used for pure rostering)

times have to be assigned to machines with different capabilities (*i.e.* each job can only be executed by a subset of the machines). While in some problems preemption is allowed (*i.e.* the processing of a job may be interrupted and continued by another machine), in other problems preemption is forbidden. The basic interval scheduling problem is to process all jobs using a minimum number of machines. If we interpret the employees as machines, minimizing the number of used employees is the same kind of problem. Each job can be processed by all machines having the required qualification. The possibility of transition is equivalent to allowing preemption. Usually, in interval scheduling problems it is assumed that all machines are available for the whole time horizon. In order to indicate this in our classification scheme, we use the entries *no-ol* and $s_d \leq 1$ in the δ-field. This means that employees are present during the whole time horizon, since shifts do not overlap and there is only one shift per day.

More generally, if demands $b_{jq} \geq 1$ for employees with qualification q are given, we can introduce b_{jq} jobs, each demanding for one machine corresponding to employees with qualification q. As all these jobs have to be processed within a fixed time window, they can not be executed by the same machine. This equivalence can be used to show the complexity of more general assignment problems.

In the following, we present new complexity results for two integrated problems. Both problems combine a rostering and an assignment problem, each of them being polynomially solvable. For the first problem, we give a polynomial algorithm, for the second problem we prove $\mathcal{NP}$-completeness.

5.1. The problem $RA \mid Q_e = 1, \text{trans}(\sigma) \mid no\text{-}ol, fch \mid -$

In this problem, the schedule from the scheduling part is already fixed and jobs $j \in \mathcal{J}$ with fixed starting times S_j and completion times C_j are given. Each employee offers only one qualification and transition is only allowed at the end of a shift. There are no overlapping shifts, but some forbidden shift changes have to be taken into account. The objective is to find a feasible asignment of employees to shifts and jobs. This problem can be solved by considering Q feasible flow problems, one for each qualification q. Within each of the flow problems, we use a slightly adapted graph from [38].

As each employee offers only one qualification, the qualifications can be considered independently from each other. The main idea is to construct Q graphs, one for each qualification, in the following way: for each shift a subgraph is constructed, containing all jobs that have to be scheduled in that shift. Each subgraph consists of a chain of vertices and therefore has a first and a last vertex. Two subgraphs are connected by a directed arc from the last vertex of one subgraph to the first vertex of the next subgraph if the two corresponding shifts belong to consecutive days and if it is allowed to work in both shifts (*i.e.* there is no forbidden shift change for them). In order to resemble a day off, for each day a free shift is added to the graph. Additionally, minimum and maximum capacities for the arcs are defined. Then a feasible flow is computed for this graph, indicating consecutive

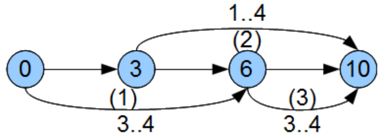

FIGURE 3. Example for a graph $G_{q,\sigma}$. There are three jobs which have to be executed by four employees. The vertices correspond to time periods where jobs starts or end. Here, these time periods are 0, 3, 6 and 10. Job 1 starts at time 0, finishes at time 6 and has a demand for 3 employees. Therefore, an arc from vertex 0 to vertex 6 with minimum capacity 3 and maximum capacity 4 is added (labeled by (1)). Job 2 starts at time 3, ends at time 10 and requires one employee. Job 3 starts at time 6, finishes at time 10 and must be executed by at least 3 employees. Finally, the arcs between the vertices without any labeling represent idle time of employees.

assignments to jobs and shifts for employees (without looking at specific employees). Afterwards, this flow is split up into several paths and these are assigned to specific employees.

For each qualification q and each shift σ, a set $\mathcal{J}_{q,\sigma}$ is computed, containing all jobs $j \in \mathcal{J}$ with $b_{jq} > 0$ and $(S_j, C_j] \cap (s_\sigma, e_\sigma] \neq \emptyset$. Based on these sets, graphs $G_{q,\sigma}$ are constructed as follows. We consider all time periods $t \in [s_\sigma, e_\sigma]$ when a job starts or ends as well as the start and the end time of the shift ($s_\sigma = t_0 < \ldots < t_i < \ldots < t_{k_\sigma} = e_\sigma$ with $t_i = \min\{t \in (t_{i-1}, e_\sigma] \mid \exists\, j \in J_{q\sigma} : t = S_j \vee t = C_j\}$, $i = 1, \ldots, k_\sigma - 1$). For each time period t_i, a vertex $v(t_i)$ is added. Between two vertices $v(t_i), v(t_{i+1})$, an arc is introduced, representing idle time of employees. For each job $j \in \mathcal{J}_{q\sigma}$, an arc $(v(\max\{s_\sigma, S_j\}), v\,(\min\{e_\sigma, C_j\}))$ with minimum capacity b_{jq} and maximum capacity $|\mathcal{E}_q|$ is added. An example for such a graph is shown in Figure 3. Afterwards, the graphs can be reduced as described in Kroon [38].

These graphs $G_{q,\sigma}$ are connected to form a graph G_q in the following manner: The graph has D levels, each representing a day $d \in D$. In level d, there are $|S_d| + 1$ subgraphs $G_{q,\sigma}$, $\sigma \in S_d \cup \{f_d\}$ where f_d indicates a free shift on day d. Two subgraphs $G_{q,\sigma}$ and $G_{q,\sigma'}$ of shifts σ, σ' belonging to two consecutive days are connected by an arc from the last vertex in $G_{q,\sigma}$ (representing time e_σ) to the first vertex in graph $G_{q,\sigma'}$ (representing time $s_{\sigma'}$) if there is no forbidden shift change between these two shifts. Afterwards, a source-vertex S, an auxiliary vertex A and a target T are added. The source is linked to the auxiliary vertex with maximum capacity $|\mathcal{E}_q|$, as we can only schedule $|\mathcal{E}_q|$ employees for qualification q. Furthermore, the auxiliary vertex is connected to all first vertices in the graphs representing a shift of day 1, and the final vertices in the graphs representing shifts of the last day are connected to the target. An example for such a graph G_q is shown in Figure 4.

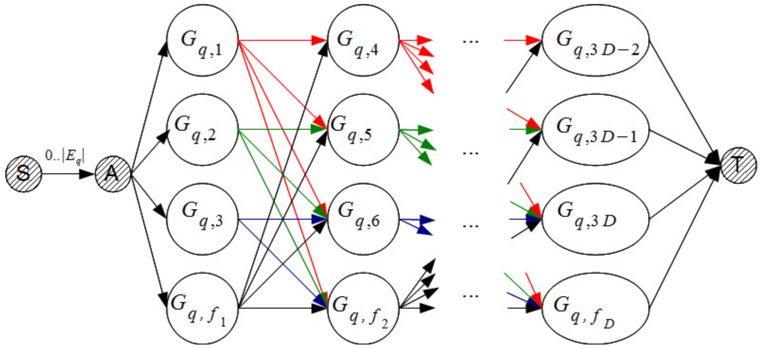

FIGURE 4. Example for a graph G_q where each day has three shifts: a morning, a late, and a night shift. The graph has D levels, in level d, there are $|S_d| + 1 = 4$ subgraphs $G_{q,\sigma}$, $\sigma \in S_d \cup \{f_d\}$ where f_d indicates a free shift. We assume that it is not allowed to work in two shifts σ, σ' of two consecutive days if $\sigma' < \sigma + 3$. For example, it is not allowed to work in the late shift of day 1 ($\sigma = 3$) and in the morning shift of day 2 ($\sigma' = 4$), *i.e.* no arc exists between the corresponding vertices. On the other hand, since it is allowed to have a free shift at day 1 and to work in any shift of day 2, the corresponding arcs exist.

For G_q a feasible flow is computed if it exists. This flow is split up into paths from the start vertex to the end vertex. These paths are then assigned to employees, saying which jobs and shifts an employee performs. More specifically, if P is a path, e is the employee assigned to P and $\mathcal{J}_P$ are all jobs represented by an arc in P, then e is assigned to all jobs in $\mathcal{J}_P$ and to the corresponding shifts. In the remaining shifts e does not have to work.

Due to the construction of the network, a feasible solution exists if and only if a feasible flow for each graph $G_q, q \in \mathcal{Q}$ exists. Each graph G_q has $\mathcal{O}(SJ)$ vertices and $\mathcal{O}(S(J+S))$ arcs. In order to compute a feasible flow, we may use the algorithm of Goldberg and Tarjan [27], which has a running time of $\mathcal{O}(nm \log \frac{n^2}{m})$ for a graph with n vertices and m arcs. Therefore, for one qualification, we need $\mathcal{O}(S^2 J(J + S) \cdot \log(\frac{SJ^2}{S+J}))$ time to compute a feasible flow. Additionally, $\mathcal{O}(|\mathcal{E}_q| \cdot S \cdot (J + S))$ time is needed to split the flow for qualification q into paths. Since $\sum_{q\in\mathcal{Q}} |\mathcal{E}_q| = E$, the complexity of the whole algorithm is $\mathcal{O}(QS^2 J(J+S) \cdot \log(\frac{SJ^2}{S+J}) + ES(J+S))$, which is polynomial.

By a slight modification, not only the feasibility problem but also the problem minimizing the objective function $C_{use(e)}$ can be solved. For this purpose, we only have to introduce costs of 1 on the arc from the source S to the auxiliary vertex A and to solve a minimum cost flow problem (all other arcs get cost zero). Then a feasible flow with minimum costs corresponds to an assignment with a minimum number of employees.

5.2. The problem $RA \mid no\text{-}ol \mid -$

In this problem again the objective is to find a feasible assignment of employees to shifts and jobs respecting a fixed job schedule. In contrast to the previous problem, each employee may have more than one qualification. On the other hand, transition is allowed, but there are no forbidden shift changes. This problem is a combination of the problems $R \mid\mid -$ and $A \mid no\text{-}ol \mid -$, which are both polynomially solvable.

In the rostering problem $R \mid\mid -$ we have to assign employees to shifts such that demands $b_{\sigma q}$ for qualifications q in shifts σ are satisfied. Since there are no forbidden shift changes, the days are independent from each other and can be treated separately. For each day, we get a transshipment problem (*cf.* Brucker *et al.* [16]) on a bipartite graph with the employees on one side and shift-qualification pairs (σ, q) on the other side. An employee e and a shift-qualification pair (σ, q) are connected by an arc if $e \in \mathcal{E}_q$. Each employee has a supply of 1, each shift-qualification pair (σ, q) has a demand of $b_{\sigma q}$. A feasible solution of this transshipment problem defines a feasible assignment of employees to shifts and qualifications.

On the other hand, in the assignment problem $A \mid no\text{-}ol \mid -$, employees are already assigned to shifts and now have to be assigned to jobs such that demands b_{jq} for qualifications q and jobs j are satisfied. Since the shifts do not overlap, each shift σ can be considered separately. The start and completion times of jobs processed in σ are sorted such that $s_\sigma = t_1 < \ldots < t_{\tau_\sigma} = e_\sigma$ is the sequence of these time periods. For each interval $(t_i, t_{i+1}]$, $i = 1, \ldots, \tau_\sigma - 1$ again a transshipment problem on a bipartite graph is solved with the working employees having a supply of 1 on one side and job-qualification pairs (j, q) with a demand of b_{jq} on the other side. An employee e and a job-qualification pair (j, q) are connected by an arc if $e \in \mathcal{E}_q$. Obviously, a feasible solution of this transshipment problem defines a feasible assignment of employees to jobs and qualifications.

For the combination of these problems, we will prove $\mathcal{NP}$-completeness. This problem is an example for the situation that combining a polynomially solvable assignment problem with a polynomially solvable rostering problem can result in an $\mathcal{NP}$-complete integrated problem.

Since there are no forbidden shift changes, we can consider each day independently. The jobs define demand profiles, *i.e.* values $\hat{b}_{qt} = \sum_{j \in \mathcal{J}} \{b_{jq} \mid t \in (S_j, C_j]\}$ for each time period $t \in \mathcal{T}$ and each qualification $q \in Q$. In the following we will prove $\mathcal{NP}$-completeness for the problem of assigning employees to shifts such that the demand profiles are respected. Since shifts do not overlap, we can consider each shift separately. Assume we are given an interval I in shift σ, in which the demands for employees for the different qualifications do not change. A demand vector $b^I = (b_q^I)_{q \in \mathcal{Q}}$ for interval I is determined by the demands b_q^I for employees with qualification q in interval I. Since jobs and hence demand profiles are given, we can divide the time covered by a shift σ into intervals with constant demands. Let $s_\sigma = t_1 < \ldots < t_{\nu_\sigma} = e_\sigma$ be the points of time when the demands of at least one qualification change. For each interval I of the form $(t_i, t_{i+1}]$, the demand vector b^I can be computed. Then we have to assign the employees to the shifts such that all demand vectors for all intervals can be fulfilled.

This rostering and assignment problem is $\mathcal{NP}$-complete as 3-SAT can be reduced to it. In an instance of 3-SAT, there are m clauses and n variables, each clause containing three literals of the set $\{x_1, \overline{x}_1, \ldots, x_n, \overline{x}_n\}$. In a feasible solution each variable x_j is set either to the value $true$ (then $\overline{x}_j = false$) or $false$ (then $\overline{x}_j = true$) in such a way that each clause contains at least one $true$-value.

An instance I of 3-SAT can be transformed into an instance I' of our problem as follows. The number of shifts is set to two. The set $\mathcal{E} = \{e(x_1), e(\overline{x}_1), \ldots, e(x_n), e(\overline{x}_n)\}$ contains an employee for each literal. The set $\mathcal{Q}$ contains $m+n$ entries. Each clause C_i, $i = 1, \ldots, m$, of 3-SAT is transformed into a qualification $q(C_i)$, and each variable x_j, $j = 1, \ldots, n$, into a qualification $q(x_j)$. Two types of demand vectors are introduced. The first type contains m demand vectors $v_1, \ldots, v_m$ for the first shift. Vector v_i demands for one employee with qualification $q(C_i)$. The other type contains one demand vector v_{m+1} for both shifts, demanding for one employee of each qualification $q(x_j), j = 1, \ldots, n$. The set $\mathcal{E}_q$ contains employee e if either q is the qualification of a variable $q(x_j)$ and e corresponds to one of the literals x_j, $\overline{x}_j$, or q is the qualification of a clause C_i and the literal corresponding to e appears in C_i.

Let e_i be the ith unit vector. We can summarize:

- $\mathcal{S} := \{\sigma_1, \sigma_2\}$;
- $\mathcal{E} := \{e(x_1), e(\overline{x}_1), \ldots, e(x_n), e(\overline{x}_n)\}$;
- $\mathcal{Q} := \{q(C_1), \ldots, q(C_m), q(x_1), \ldots, q(x_n)\}$;
- $e \in \mathcal{E}_q$ if $\begin{cases} q = q(x_j) \text{ and } e \in \{x_j, \overline{x}_j\}; \text{ or} \\ q = q(C_i) \text{ and } e \text{ appears in } C_i; \end{cases}$
- demand vectors for shift σ_1: $e_1, ..., e_m$, $v_{m+1} = \sum_{i=m+1}^{m+n} e_i$;
- demand vector for shift σ_2: $v_{m+1} = \sum_{i=m+1}^{m+n} e_i$.

This transformation can obviously be done in polynomial time. We now show that I has a feasible solution if and only if I' has a feasible solution.

Let a feasible solution for I be given. This solution can be transformed into a solution of the transformed instance I'. Let L^+ be the set of true literals. W.l.o.g. we can assume that L^+ contains n literals, where each literal comes from another variable. We can assign the corresponding employees to shift σ_1 and the remaining n employees to shift σ_2. The resulting solution for I' is feasible: as the literals in L^+ cover all variables (or their negation), the demand vectors $\sum_{i=m+1}^{m+n} e_i$ are fulfilled for both shifts. As the solution for 3-SAT is feasible, the remaining demand vectors for shift σ_1 are also fulfilled.

Assume conversely that a feasible solution for the instance I' is given. This solution can be transformed into a feasible solution for I. Let E^{σ_1} be the employees working in shifts σ_1 and E^{σ_2} the employees working in shift σ_2. The literals corresponding to the employees in set E^{σ_1} are set to $true$, the other ones are set to *false*. As the demand profiles $\sum_{i=m+1}^{m+n} e_i$ are fulfilled by the sets E^{σ_1}, E^{σ_2}, the employees corresponding to the literals x_j, $\overline{x}_j$ are assigned to different shifts for $j = 1, \ldots, n$. Since the set E^{σ_1} also fulfills the remaining demand vectors, for each qualification there is at least one employee who can be assigned to that qualification. Therefore, in the solution for 3-SAT there is at least one literal with the value *true* in each clause.

Example 5.1. The instance I given by $(x_1 \vee \overline{x}_3 \vee x_5) \wedge (x_2 \vee x_4 \vee \overline{x}_5) \wedge (\overline{x}_1 \vee x_2 \vee x_3) \wedge (x_1 \vee \overline{x}_2 \vee \overline{x}_4)$ of 3-SAT with $m = 4$ clauses and $n = 5$ variables is transformed into the following instance I':

- $\mathcal{S} := \{\sigma_1, \sigma_2\}$,
- $\mathcal{E} := \{e(x_1), e(\overline{x}_1), \ldots, e(x_5), e(\overline{x}_5)\}$,
- $\mathcal{Q} := \{q(C_1), \ldots, q(C_4), q(x_1), \ldots, q(x_5)\}$,
- $\mathcal{E}_{q(x_1)} = \{e(x_1), e(\overline{x}_1)\}$, $\mathcal{E}_{q(x_2)} = \{e(x_2), e(\overline{x}_2)\}$, $\mathcal{E}_{q(x_3)} = \{e(x_3), e(\overline{x}_3)\}$, $\mathcal{E}_{q(x_4)} = \{e(x_4), e(\overline{x}_4)\}$, $\mathcal{E}_{q(x_5)} = \{e(x_5), e(\overline{x}_5)\}$, $\mathcal{E}_{q(C_1)} = \{e(x_1), e(\overline{x}_3), e(x_5)\}$, $\mathcal{E}_{q(C_2)} = \{e(x_2), e(x_4), e(\overline{x}_5)\}$, $\mathcal{E}_{q(C_3)} = \{e(\overline{x}_1), e(x_2), e(x_3)\}$, $\mathcal{E}_{q(C_4)} = \{e(x_1), e(\overline{x}_2), e(\overline{x}_4)\}$
- demand vectors for shift σ_1: $(1,0,0,0,0,0,0,0,0)$, $(0,1,0,0,0,0,0,0,0)$, $(0,0,1,0,0,0,0,0,0)$, $(0,0,0,1,0,0,0,0,0)$, $(0,0,0,0,1,1,1,1,1)$
- demand vector for shift σ_2: $(0,0,0,0,1,1,1,1,1)$.

A feasible solution for I is $x = (true, true, false, false, false)$, with $L^+ = \{x_1, x_2, \overline{x}_3, \overline{x}_4, \overline{x}_5\}$. The corresponding solution for I' assigns employees $e(x_1)$, $e(x_2), e(\overline{x}_3)$, $e(\overline{x}_4)$, $e(\overline{x}_5)$ to shift σ_1 and $e(\overline{x}_1)$, $e(\overline{x}_2)$, $e(x_3)$, $e(x_4)$, $e(x_5)$ to shift σ_2. The demand vectors for shift σ_1 are fulfilled: employee $e(x_1)$ offers qualifications $q(C_1)$ and $q(C_4)$, employee $e(x_2)$ offers $q(C_2)$ and $q(C_3)$. Furthermore, $e(x_j) \in \mathcal{E}_{q(x_j)}$ for $q = 1, 2$ and $e(\overline{x}_j) \in \mathcal{E}_{q(x_j)}$ for $q = 3, 4, 5$. The demand vector for shift σ_2 is also fulfilled: $e(\overline{x}_j) \in \mathcal{E}_{q(x_j)}$ for $q = 1, 2$ and $e(x_j) \in \mathcal{E}_{q(x_j)}$ for $q = 3, 4, 5$.

6. Concluding remarks

In this paper, we introduced a classification scheme for integrated staff rostering and scheduling problems, consisting of five fields $(\alpha) + (\beta \mid \gamma \mid \delta \mid \varepsilon)$. We gave a survey on problems studied in the literature and showed how they can be classified according to this scheme. Additionally, we summarized known complexity results and provided two new results.

Future research concerning complexity could be helpful to determine the border between polynomially solvable and $\mathcal{NP}$-hard problems. Additionally, polynomially solvable special cases may be useful in decomposition algorithms for integrated problems. Based on these results more efficient algorithms for integrated problems in practice could be developed.

References

[1] H. Alfares, J. Bailey and Wen Y. Lin, Optimization and heuristic models to integrate project task and manpower scheduling. *Comput. Ind. Eng.* **29** (1995) 473–476.

[2] H.K. Alfares, J.E. Bailey and W.Y. Lin, Integrated project operations and personnel scheduling with multiple labour classes. *Prod. Plan. Control* **10** (1999) 570–578.

[3] E.M. Arkin and E.B. Silverberg, Scheduling jobs with fixed start and end times. *Discrete Appl. Math.* **18** (1987) 1–8.

[4] C. Artigues, M. Gendreau and L.-M. Rousseau, A flexible model and a hybrid exact method for integrated employee timetabling and production scheduling, in *Proc. of the 6th International Conference on Practice and Theory of Automated Timetabling (PATAT)*, Vol. 3867. Springer (2007) 67–84.

[5] C. Artigues, M. Gendreau, L.-M. Rousseau and A. Vergnaud, Solving an integrated employee timetabling and job-shop scheduling problem via hybrid branch-and-bound. *Comput. Oper. Res.* **36** (2009) 2330–2340.

[6] R.M. Awad and J.W. Chinneck, Proctor assignment at Carleton University. *Interfaces* **28** (1998) 58–71.

[7] O. Bellenguez and E. Néron, Lower bounds for the multi-skill project scheduling problem with hierarchical levels of skills, in *Proc. of the 5th International Conference on Practice and Theory of Automated Timetabling (PATAT)*, Vol. 3616. Springer (2005) 229–243.

[8] O. Bellenguez-Morineau and E. Néron, A branch-and-bound method for solving multi-skill project scheduling problem. *RAIRO–Oper. Res.* **41** (2007) 155–170.

[9] S. Bertels and T. Fahle, A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem. *Comput. Oper. Res.* **33** (2006) 2866–2890.

[10] J. Błażewicz, J.K. Lenstra and A.H.G. Rinnooy Kan, Scheduling subject to resource constraints: classification and complexity. *Discrete Appl. Math.* **5** (1983) 11–24.

[11] J. Błażewicz, K.H. Ecker, E. Pesch, G. Schmidt and J. Weglarz, *Handbook on Scheduling: From Theory to Applications*. Springer (2007).

[12] N. Boysen, M. Fliedner and A. Scholl, A classification of assembly line balancing problems. *Eur. J. Oper. Res.* **183** (2007) 674–693.

[13] P. Brucker, A. Drexl, R. Möhring, K. Neumann and E. Pesch, Resource-constrained project scheduling: Notation, classification, models, and methods. *Eur. J. Oper. Res.* **112** (1999) 3–41.

[14] P. Brucker and S. Knust, *Complex Scheduling*. Springer, 2nd edition (2012).

[15] P. Brucker and S. Knust, Complexity results for scheduling problems. http://www.inf.uos.de/knust/class/.

[16] P. Brucker, R. Qu and E.K. Burke, Personnel scheduling: Models and complexity. *Eur. J. Oper. Res.* **210** (2011) 467–473.

[17] P. Brucker and R. Qu, Network flow models for intraday personnel scheduling problems. *Ann. Oper. Res.* **218** (2014) 107–114.

[18] J. Brunner, J. Bard and R. Kolisch, Flexible shift scheduling of physicians. *Health Care Manage. Sci.* **12** (2009) 285–305.

[19] J. Brunner, J. Bard and R. Kolisch, Midterm scheduling of physicians with flexible shifts using branch-and-price. *IIE Trans.* **43** (2010) 84–109.

[20] E.K. Burke, P. De Causmaecker, G. Vanden Berghe and H. Van Landeghem, The state of the art of nurse rostering. *J. Schedul.* **7** (2004) 441–499.

[21] P. De Causmaecker and G. Vanden Berghe, A categorisation of nurse rostering problems. *J. Schedul.* **14** (2011) 3–16.

[22] M. Desrochers, J.K. Lenstra and M.W.P. Savelsbergh, A classification scheme for vehicle routing and scheduling problems. *Eur. J. Oper. Res.* **46** (1990) 322–332.

[23] A. Drexl, Scheduling of project networks by job assignment. *Manage. Sci.* **37** (1991) 1590–1602.

[24] A.T. Ernst, H. Jiang, M. Krishnamoorthy, B. Owens and D. Sier, An annotated bibliography of personnel scheduling and rostering. *Ann. Oper. Res.* **127** (2004) 21–144.

[25] P.R. Ferreira Jr. and A.L.C. Bazzan, Distributed task scheduling using a swarm intelligence approach. In *Proceedings of the 7th Brazilian Meeting on Artificial Intelligence.* SBC (2009) 979–988.

[26] M.R. Garey and D.S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W.H. Freeman & Co. (1990).

[27] A.V. Goldberg and R.E. Tarjan, A new approach to the maximum flow problem, in *Proc. of the eighteenth annual ACM symposium on Theory of computing*, STOC '86. ACM (1986) 136–146.

[28] R.L. Graham, E.L. Lawler, J.K. Lenstra and A.H.G. Rinnooy Kan, Optimization and approximation in deterministic sequencing and scheduling: A survey. *Ann. Discrete Math.* **5** (1979) 287–326.

[29] O. Guyon, P. Lemaire, E. Pinson and D. Rivreau, Cut generation for an integrated employee timetabling and production scheduling problem. *Eur. J. Oper. Res.* **201** (2010) 557–567.

[30] O. Guyon, P. Lemaire, E. Pinson and D. Rivreau, Solving an integrated job-shop problem with human resource constraints. *Ann. Oper. Res.* **213** (2014) 147–171.

[31] S. Hartmann and D. Briskorn, A survey of variants and extensions of the resource-constrained project scheduling problem. *Eur. J. Oper. Res.* **207** (2010) 1–14.

[32] C. Heimerl and R. Kolisch, Scheduling and staffing multiple projects with a multi-skilled workforce. *OR Spektrum* **32** (2010) 343–368.

[33] J. Herbers, *Models and Algorithms for Ground Staff Scheduling on Airports.* Ph.D. thesis, Rheinisch-Westfälische Technische Hochschule Aachen (2005).

[34] W. Herroelen, E. Demeulemeester and B. de Reyck, A classification scheme for project scheduling problems, in *Project Scheduling – Recent Models, Algorithms and Applications*, edited by J. Weglarz. Kluwer Academic Publishers (1998) 1–26.

[35] P. Kilby, The augmented regret heuristic for staff scheduling, in *Proceedings of the 16th Australian Society of Operations Research (ASOR 2001)*, McLaren Vale, South Australia (2001).

[36] A.W.J. Kolen, J.K. Lenstra, C.H. Papadimitriou and F.C.R. Spieksma, Interval scheduling: A survey. *Nav. Res. Logist.* **54** (2007) 530–543.

[37] G.J. Koop, Multiple shift workforce lower bounds. *Manage. Sci.* **34** (1988) 1221–1230.

[38] L.G. Kroon, M. Salomon and L.N. van Wassenhove, Exact and approximation algorithms for the tactical fixed interval scheduling problem. *Oper. Res.* **45** (1997) 624–638.

[39] H.C. Lau, On the complexity of manpower shift scheduling. *Comput. Oper. Res.* **23** (1996) 93–102.

[40] J.S. Loucks and F.R. Jacobs, Tour scheduling and task assignment of a heterogeneous work force: A heuristic approach. *Dec. Sci.* **22** (1991) 719–738.

[41] S.M. Roberts and L.F. Escudero, Scheduling of plant maintenance personnel. *J. Optim. Theor. Appl.* **39** (1983) 323–343.

[42] M. Segal, The operator-scheduling problem: A network-flow approach. *Oper. Res.* **22** (1974) 808–823.

[43] J. Van den Bergh, J. Belien, P. De Bruecker, E. Demeulemeester and L. De Boeck, Personnel scheduling: A literature review. *Eur. J. Oper. Res.* **226** (2013) 367–385.