Personnel Scheduling: Models and Complexity

Peter Brucker · Rong Qu · Edmund Burke

Abstract Due to its complexity, its challenging features, and its practical relevance, personnel scheduling has been heavily investigated in the last few decades. However, there is a relatively low level of study on models and complexity in these important problems. In this paper, we present mathematical models which cover specific aspects in the personnel scheduling literature. Furthermore, we address complexity issues by identifying polynomial solvable and NP-hard special cases.

1 Introduction

De Causmaecker et al. [17] classified companies according to different personnel scheduling problems. The corresponding scheduling problems lead to different models. The classification is:

- permanence centred planning
- fluctuation centred planning
- mobility centred planning
- project centred planning.

The number of personnel needed, for example, for police services and hospitals, is usually defined in advance [4][13]. Corresponding personnel scheduling problems are called *permanence centred*. In contrast, for warehouses or distribution centers, as well as for call centers, fast food restaurants and postal services [2][4][5][6][22], the personnel planning is based on *fluctuating* demand. *Mobility centred* planning occurs when duties involve transportation. Transportation companies like airlines and railway companies face such planning problems [9][29][32][35]. Other examples are duties of

Peter Brucker
Fachbereich Mathematik/Informatik, Universität Osnabrück, Albrechtstr. 28a, 49069 Osnabrück, Germany
E-mail: pbrucker@uni-osnabrueck.de

Rong Qu, Edmund Burke
Automated Scheduling, Optimization and Planning (ASAP) Group, School of Computer Science, University of Nottingham, NG8 1BB, UK
E-mail: rxq,ekb@cs.nott.ac.uk

health and safety boards or those in connection with home health care [23]. *Project centred* planning arises in companies which divide their work into projects to which they assign different groups of employees. Typical examples are software development and consultancy, as well as construction project [3][25][34].

As a widely recognised and challenging problem that appears regularly in the scientific literature, personnel scheduling has attracted significant research attention in both research and practice. There are general guidelines on modelling different types of constraints in staff scheduling problems [8]. A descriptive general model for nurse rostering problems has also been built in conjunction with neighborhood design in local search approaches [7]. In [18], a structured model has been proposed for integrated staff planning and rostering problems. A formal description of the short term rostering is also presented. However, although algorithm solutions and models have been built in solving specific problems, there are no mathematical models in the literature for general personnel scheduling problems.

There are, however, some specific personnel scheduling models in the literature. For example, personnel scheduling with flexible demand [31], and personnel scheduling with restricted task changes (working place changes) [14] have been considered. As an example of just one specific type of personnel scheduling problem, nurse rostering has been heavily investigated in recent years and solved by using a variety of techniques, from heuristic algorithms to exact methods [13][20][21]. The Dantzig set covering formulation [15] can be used to formulate rostering problems [21]. However, such a model is usually used to formulate a simplified problem, and is difficult to be used to cater some common characteristics in personnel scheduling problems. It is important to develop general mathematical models which cover various special cases, and underpin a fundamental theory for personnel scheduling.

In this paper, a mathematical model which covers permanence and fluctuation centred planning as well as the rostering part of mobility centred planning is presented. As an example of permanence centred planning, a nurse rostering problem is briefly discussed. Furthermore, two other special scheduling problems found in the literature are presented. A fourth problem describes a situation in which one has to decide which employee has to work on which days of a period of several possible days. Such decisions have an influence on the ability to cover the demand for employees in each period of each day. Later the mathematical model is extended to cover project centred planning as well.

For the development of suitable methods to solve personnel scheduling problems, it is useful to get some insight into the complexity of special cases. Very little work has been carried out on relevant problems. In [28], the Changing Shift Assignment Problem which arises from airport ground crew scheduling has been concerned from the complexity point of view. A restricted version of the problem has shown to be NP-hard. In this paper we identify cases in general personnel scheduling problems which are either NP-hard or polynomially solvable. All polynomially solvable cases can be solved efficiently by network flow techniques which can be used as sub-routines within heuristic procedures for more complex problems. There is a rich literature on heuristics which solve specific problems. Our aim in this work is to develop a general theory for personnel scheduling. Such mathematical models allow developing general heuristics which can be adapted to special cases. This is often more efficient than to develop specific heuristics in specific cases. Unlike traditional scheduling, in which theoretical models and complexity issues have been studied for long, personnel scheduling lacks such theory.

This paper is organized as follows. The mathematical models and its special cases are first presented in Section 2, followed by the complexity results in Section 3. In Section 3.1 network flow formulations are presented while Section 3.2 contains the NP-completeness results. These models can be directly applied in permanence and fluctuation centred planning. The last section contains conclusions.

2 Models

2.1 A Mathematical Model for Personnel Scheduling

A general personnel scheduling problem can be formulated as follows.

There is a planning horizon $[0, T]$ divided into periods $[t, t+1[$ for $t = 0, 1, \cdots, T-1$. Within the planning horizon m tasks $j = 1, \ldots, m$ must be performed. $D_j(t)$ is the number of employees needed to perform task j in time period $[t, t+1[$ $(t = 0, 1, \cdots, T-1)$. It is called the demand profile for task j.

There is set E of n employees. Associated with each employee $e \in E$ is a subset Q_e of tasks for which e is qualified, i.e. e can be assigned to tasks in Q_e only. A working pattern for an employee e is defined by

- a zero-one vector $(w_e(t))_{t=0}^{T-1}$ where $w_e(t) = 1$ if and only if e is available in period $[t, t+1[$, and
- an assignment of a task belonging to Q_e for each time period $[t, t+1[$ with $w_e(t) = 1$.

A working pattern of an employee thus consists of tasks which the employee is qualified to perform, at time periods the employee is available. The working patterns are represented by binary vectors $\pi = (\pi(j, t))$ where $\pi(j, t) = 1$ if and only if $[t, t+1[$ is a working period in which task j has to be performed.

The working pattern concept is illustrated by the following example.

Example 1 Assume that the planning horizon are the seven days of a week and that $t = 0$ corresponds to Monday. Furthermore, assume that an employee e who is qualified for two tasks 1 and 2 works from Tuesday to Saturday (i.e. $w_e(t) = 0$ for $t = 0, 6$ and $w_e(t) = 1$ for $t = 1, \cdots, 5$) and is assigned to task 2 on Saturday and to task 1 on the other working days. Then the corresponding working pattern π is specified by $\pi(1, \nu) = 1$ for $\nu = 1, \cdots, 4$, $\pi(2, 5) = 1$, and $\pi(j, t) = 0$ for all other values (j, t).

Note that not all possible working patterns may be feasible for an employee. The feasible working patterns are usually specified by some hard constraints which depend on the specific problem. The set of all feasible working patterns for employee e is denoted by P_e.

One has to assign to employees $e \in E$ feasible working patterns $\pi_e \in P_e$ such that

- *the demand $D_j(t)$ of all tasks j is covered in each period $[t, t+1[$, and*
- *the costs of the assignment to employees are minimized.*

Additional so called "soft constraints" may be imposed onto the working pattern $\pi_e \in P_e$. In this case a penalty $u(\pi_e)$ may be defined which measures how much pattern π_e violates the soft constraints. Possible assignment costs are

- the sum of penalties $u(\pi_e)$ of all assigned employees e, or

- the number of employees needed, or
- the costs for the employees which are assigned.

The mathematical model may be restricted by not allowing task changes, i.e. each employee e must be assigned to the same task in Q_e during all his working periods, with no change of working places. In another version task changes may be restricted in the sense that during certain time periods task changes are not allowed.

On the other hand a model with $D_j(t) \in \{0,1\}$ for all t can be further generalized by the introduction of a flexible demand. In this case each task j has a duration p_j and must be processed within a time window $\big[L_j, R_j\big[\subseteq [0,T]$ with $R_j - L_j \geq p_j$, i.e. $D_j(t) \in \{0,1\}$ for $t = L_j, L_j+1, \ldots, R_j - 1$ where $\sum_{t=L_j}^{R_j-1} D_j(t) = p_j$.

The problem can be modelled by a binary linear program. Let $x_{e\pi}$ be a binary variable which is set equal to one if and only if the working pattern π is assigned to employee e. Additionally let $c_{e\pi}$ be the costs if the working pattern π is assigned to employee e. Then the linear program has the form

$$\min \sum_{e\in E} \sum_{\pi \in P_e} c_{e\pi} x_{e\pi} \tag{1}$$

subject to

$$\sum_{\pi \in P_e} x_{e\pi} \leq 1, \quad e \in E \tag{2}$$

$$\sum_{e\in E} \sum_{\pi \in P_e} \pi(j,t) x_{e\pi} \geq D_j(t), \quad \text{for all } (j,t) \tag{3}$$

$$x_{e\pi} \in \{0,1\} \text{ for all } e \in E \text{ and } \pi \in P_e.$$

By constraint (2) at most one working pattern π is assigned to employee e. In some cases each employee must be assigned a working pattern, and thus the inequality, i.e. ≤ 1 in (2) should be replaced by the equality, i.e. $= 1$. The expression at the left hand side of (3) is equal to the number of employees assigned to task j in period $[t, t+1[$ and (3) forces this number to be at least equal to the demand of j in period $[t, t+1[$.

To take care of flexible demand one has to introduce additional binary variables y_{jt} denoting the minimal number of employees needed by task j in period $[t, t+1[$. In (3) one has to replace $D_j(t)$ by y_{jt} and the constraints

$$\begin{array}{c} \sum_{t=L_j}^{R_j-1} y_{jt} = p_j \quad \text{for all tasks } j \\ y_{jt} \in \{0,1\} \end{array} \tag{4}$$

must be added.

Sometimes it is useful to generalize the binary linear program (1) to (3) by adding the constraints

$$\sum_{e\in E} \sum_{\pi \in P_e} \pi(j,t) x_{e\pi} \leq UD_j(t) \quad \text{for all } (j,t) \tag{5}$$

where $UD_j(t) \geq D_j(t)$ is an upper bound on the number of employees needed for task j in period $[t, t+1[$. If $D_j(t) = UD_j(t)$ then exactly $D_j(t)$ employees are needed.

The general mathematical model we defined here is similar to that by Dowsland in [19] for nurse rostering problems. A tabu search is devised to explore all possible pre-defined feasible shift patterns x_{ij} for nurses, $x_{ij} = 1$ if nurse i works on pattern j.

p_{ij} is the penalty associated with nurse i on working pattern j. The objective of the tabu search is to find the best quality roster (of the lowest total penalty), satisfying the covering requirement.

2.2 Project Centred Planning

In traditional scheduling theory, resource constrained project scheduling has been well investigated [10]. In personnel scheduling, no general model exists in project centred planning. In project centred planning, the demand $D_j(t)$ for each task (which is usually called activity) is not fixed for the time periods $[t, t+1[$ but depends on a schedule for the tasks j. A schedule for task j is defined by a starting time S_j and a processing time p_j. Furthermore, task j is processed without interruption until the finishing time $C_j = S_j + p_j$.At least $D_j(k)$ employees are needed to perform task j in period $\left[S_j + k - 1, S_j + k\right[$ for $k = 1, \cdots, p_j$.

Additionally, there are precedence constraints $(i, j) \in A$ meaning that task j cannot start before task i finishes, i.e.

$$S_i + p_i \leq S_j. \tag{6}$$

Usually, the precedence constraints are defined by an acyclic directed graph (V, A) where $V = \{1, \cdots, m\}$ is the set of all tasks and $A \subseteq V \times V$.

The latest finishing time $C_{\max} = \max_{j=1}^{m} C_j = \max_{j=1}^{m} \left(S_j + p_j\right)$ of all tasks is called makespan of the project.

One has to assign at most one working pattern $\pi \in P_e$ to each employee e and to schedule the tasks, i.e. fix the starting times S_j, such that

- *all precedence constraints (6) are satisfied,*
- *at least $D_j(k)$ employees are available to perform each task j in periods $[S_j + k - 1, S_j + k[$ for $k = 1, \cdots, p_j$*
- *the makespan is minimized.*

We assume that all tasks are available at time 0 and that all data are integer. Then the problem can be formulated as an integer linear program of the following form:

$$\text{minimize } C_{\max}$$

subject to

$$S_j + p_j \leq C_{\max} \quad j = 1, \cdots, m \tag{7}$$

$$S_i + p_i \leq S_j \quad (i, j) \in A \tag{8}$$

$$\sum_{\pi \in P_e} x_{e\pi} \leq 1 \quad e \in E \tag{9}$$

$$\sum_{e \in E} \sum_{\pi \in P_e} \pi \left(j, S_j + k - 1\right) x_{e\pi} \geq D_j(k) \quad \text{for all } j = 1, \cdots, m \quad \text{and } k = 1, \cdots, p_j \tag{10}$$

$$x_{e\pi} \in \{\,0{,}1\} \quad \text{for all } e \in E \text{ and } \pi \in P_e$$
$$S_j \geq 0 \quad j = 1, \cdots, \text{m}$$

2.3 Some Special Problems

In the following subsections 2.3.1 - 2.3.4, four problems which are covered by the mathematical models discussed in Section 2.1 are presented. Problem 1 is a nurse rostering problem where in a working pattern, at most one task (shift) is assigned to each day of five weeks within the time horizon. In Problem 2, the employees have to work in blocks of consecutive days, separated by rests of consecutive days, without changing tasks in the blocks. In Problem 3, the demand is flexible, employees are available within time windows, and arbitrary task changes are possible. Problem 4 is decomposed into two levels: in the first level the working days and rest days are to be fixed for each employee, and in the second level for each day an intraday scheduling problem is solved which calculates for each employee working on the day a working pattern. The first three problems are from the literature and the last problem arises in connection with many personnel scheduling problems. Due to the general definition of working pattern these different problems can be covered by the same model.

Some remarks on mobility centred planning can be found in Section 2.3.5.

2.3.1 Problem 1([12]): A nurse rostering problem

In a typical nurse rostering problem, one has to assign nurses to shifts within a planning period. There may be different types of nurses and the number of different shift types is quite small. Table 1 taken from ([12]) shows the demand for one type of nurse during a week.

Table 1 Shift types and demand during a week.

			Demand						
Shift type	Start time	End time	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Early	07:00	16:00	3	3	3	3	3	2	2
Day	08:00	17:00	3	3	3	3	3	2	2
Late	14:00	23:00	3	3	3	3	3	2	2
Night	23:00	07:00	1	1	1	1	1	1	1

The assignment has to satisfy hard constraints and soft constraints. Hard constraints must be fulfilled. This is not the case for soft constraints. If they are not fulfilled penalties are charged. Constraints are usually problem dependant. Therefore, we list here some important constraints.

Possible hard constraints are:

1. The demand needs to be fulfilled (i.e. all requested shifts must be covered).
2. For each day, one nurse may start only one shift.
3. The maximum number of night shifts is 3 per period of 5 consecutive weeks.
4. A nurse must receive at least 2 weekends off duty per 5 week period. A weekend off duty lasts from Saturday 00:00 to Monday 04:00.
5. Following a series of at least 2 consecutive night shifts, a 42 hours rest is required.
6. The number of consecutive night shifts is at most 3.
7. The number of consecutive shifts is at most 6.

Possible soft constraints and their penalties are:

1. For the period from Friday 23:00 to Monday 0:00, a nurse should have either no shifts or 2 shifts (complete weekend). (penalty 1000)
2. Avoid sequences of shifts of length 1 for all nurses. (penalty 1000)
3. The rest after a series of day, early or late shifts is at least 2 days. (penalty 100)
4. For all nurses, the length of a series of early shifts should be within the range [2, 3]. It could be within another series of shifts. (penalty 10)
5. An early shift after a day shift should be avoided. (penalty 5)
6. An early shift after a late shift should be avoided. (penalty 5)
7. A day shift after a late shift should be avoided. (penalty 5)

The problem considered is to find shift assignments for nurses which satisfy the hard constraints and minimize the sum of penalties of violated soft constraints. An additional objective could be to minimize the number of nurses.

This problem can be seen as a special case of the mathematical model in which the time horizon consists of all days of a period of five consecutive weeks. There are only four task types which correspond to the shifts. The feasible working patterns are defined by the hard constraints.

In [12], high quality partial schedules (low penalty patterns) of five weeks have been used to construct cyclic assignments of nurses by repeatedly shifting the schedules for nurses. A complete roster is then constructed by assigning the remaining shifts to nurses based on the cyclic assignments for the above problem. In the literature on nurse rostering [21], cyclic scheduling has also been studied especially in early research.

In a recent survey on nurse rostering [13], approaches of meta-heuristics, heuristics, artificial intelligence and mathematical programming have been reviewed. Recently, the First International Nurse Rostering Competition 2010 [26] has been launched to encourage multi-disciplinary research in nurse rostering.

2.3.2 Problem 2([14]): A problem with restricted task changes

Another problem has been formulated as follows.

There is a planning horizon of T days $d = 1, ..., T$. n_d tasks are to be performed on day d. Task j needs $D_j(d)$ employees.

Each employee performs working blocks of consecutive days, briefly called *blocks*, and after each block has a rest of consecutive days called *rests*. Within each block the employee performs the same task. There are B block types b. b has a duration of d_b days. There are R rest types r. r has a duration of d_r days.

A *working pattern* consists of a sequence of blocks assigned to tasks and rests, alternating between the blocks and rests such that the total number of working and rest days is equal to T.

There is a list of infeasible sequences of the form $(b_1, j_1)r(b_2, j_2)$, meaning that it is not possible to have a block type b_1 assigned to task j_1 followed by a block type b_2 assigned to task j_2 with a rest of type r in between.

Each employee works for at most s days belonging to a set of special days (including Sundays and holidays) within the planning horizon (*special day condition*).

A working pattern is *feasible* if it contains no infeasible sequences and satisfies the special day condition.

One has to assign feasible working patterns to the employees such that

- *the needs of all tasks within the planning horizon are covered, and*

– *the number of employees involved is minimized.*

This problem is a problem with restricted task changes because within a block an employee cannot change the task he/she has to perform.

2.3.3 Problem 3([31]): A problem with flexible demand

The following problem is a problem with flexible demand.

The planning period considered consists of several days.

For each employee e, one has to decide which are the days off for e. Thus, after such decisions, associated with each day d within the planning period there is a subset A_d of employees which have to work on day d. The corresponding intraday scheduling problem is a problem with flexible demand in which task changes are possible. It can be described as follows.

Each employee $e \in A_d$ is available during some time window $[S_e^d, F_e^d[$ which can be empty (in case of day off for e). A shift for employee e is a time interval $[V_e^d, W_e^d[$ with $S_e^d \leq V_e^d \leq W_e^d \leq F_e^d$ and $W_e^d - V_e^d \geq m_e^d$, where m_e^d is a given minimal shift length. During each period within a shift an employee performs a task, or has a (long or short) break, or is idle. There are maximal or minimal time distances between V_e^d, W_e^d, the starting times, or finishing times of breaks. Breaks cannot be interrupted.

There are n tasks $j = 1, \cdots, n$. Each employee is qualified for all tasks. Each task j has a duration p_j^d and must be processed within a time window $[R_j^d, D_j^d[$ with $D_j^d - R_j^d \geq p_j^d$. Different employees may perform a task which may be divided over different employee. Also interruptions and later processing of a task are possible. However, in each time period at most one employee processes task j and the total processing time must be equal to p_j^d.

For each day d, one has to assign feasible shifts to the employees $e \in A_d$ and for each shift to assign tasks to its active periods such that

– *the duration of each task is covered within its time window, and*
– *the total labor costs are minimal.*

The labor costs are defined as follows: meal breaks are unpaid; short rest breaks are compensated; an overtime rate is paid for the time of a shift exceeding a given limit M; if an employee is not given at least two days off for a week then there is an additional pay. In [31] the problem is solved by decomposing it in two parts. At an upper level the working days and shifts for the working days are fixed for each employee. Then for each day it is checked whether all tasks can be covered by the shifts of the employees working on this day. The following Theorem 4 shows that the latter problem can be solved by network flow techniques such as, for example, the cycle cancelling algorithm, or the network simplex algorithm [1]. To solve the overall problem a tabu search is performed for the upper level.

The next problem is related to Problem 3. However, there are the following differences:

– instead of flexible demand the demand is given by demand profiles;
– daily shifts are defined by sets of working periods;
– rather than being qualified for any task, employees must perform tasks belonging to a subset of tasks.

2.3.4 Problem 4: A multi-day personnel scheduling problem

A multi-day personnel scheduling problem can be formulated informally as follows.

There is a planning horizon consisting of a number of consecutive days. Associated with each day is a set of periods in which certain tasks have to be performed. For each period of a day and task which has to be performed in this period, a given number of employees is needed (demand profile).

On the other side there are employees. The planning horizon must be divided into working days and rest days for each employee.

A shift has to be assigned to each working day of an employee. *Shifts* consist of a set of *working periods* possibly interrupted by breaks and idle times which are part of the shift.

For each employee there is a set of tasks he can be assigned to.

A *working pattern* for an employee is defined by

- the set of working days,
- for each working day a shift, and
- for each working period of a shift a part of the task which can be performed by the employee. It is possible that in a shift of the employee, different tasks are performed.

A working pattern is feasible for an employee if it satisfies a number of constraints (which may depend on the employee).

One has to decide which employee is needed and to assign to each employee which is needed a feasible working pattern. This has to be done in such a way that

- *all tasks can be performed (i.e. the demand of tasks for employees is satisfied), and*
- *corresponding costs are minimized.*

As in the other models possible costs are labour costs, penalty costs for violating some (soft) constraints, or assignment costs.

The problem can be decomposed into two levels which we denote by *days scheduling* and *intraday scheduling* level. At the days level one has to assign the working days to employees; while at the intraday level for each employee working on the day, one has to assign a shift, and to each working period of this shift a task for which the employee is qualified. If the planning horizon consists of a number of consecutive days, with time periods for each day, working patterns can be defined with respect to these time periods on all days. Thus, Problem 4 may be considered as a special case of the problem introduced in Section 2.1.

As mentioned above, Problem 4 arises in connection with many personnel scheduling problems in the literature. The staff scheduling problem in the postal service [6] can be classified under Problem 4, as well as fluctuation centered planning problem.

In Section 3.1 (see Theorem 1), we will show that if in the general model the working days and shifts of all employees are fixed then a feasible assignment of tasks (if it exists) can be calculated in polynomial time if task changes are allowed. Such feasible assignment can be used in the evaluation step of a local search.

2.3.5 Mobility Centred Planning

The main application of mobility centred planning can be found in the area of public transport. Typical examples are crew scheduling models for railway companies and airlines. Vehicle (or plane) scheduling refers to optimization problems in which vehicles (or planes) must be assigned to time-tabled trips starting and ending at some depot. Trips are divided into segments such that at the beginning and the end of a segment a change of personal is possible. The next step is crew scheduling which is conventionally divided into further steps: crew pairing and crew rostering. A crew pairing is a sequence of connected segments that starts and ends at the same base. The objective is to cover each train or flight segment by pairings with minimum costs. The objective of crew rostering is to assign crew members to sequences of crew pairings.

In terms of the mathematical model discussed in this paper the sequences of crew pairings can be considered as working pattern. They have to be assigned to crew members in such a way that each train or flight segment (which may be considered as task) will be covered.

3 Complexity

3.1 Polynomially Solvable Cases

We describe special cases of the mathematical model which are polynomially solvable. These problems can be formulated in connection with Problems 3 and 4 in Section 2.3. However, they are also of interest on their own.

First, we present a lemma which is used in connection with the theorems we formulate next.

Lemma 1 *The special case of the mathematical model in Section 2.1in which the demand of each task j is constant, i.e. $D_j(t) = D_j$ for $t = 0, \cdots, T-1$, and each employee is available in all periods, i.e. $w_e(t) = 1$ for all $t = 0, \cdots, T-1$ can be formulated as a minimum cost network flow problem [1].*

Proof Due to the special data task changes are not needed for employees. Therefore the problem can be formulated as a transhipment problem [30]. The underlying network has

- a node e with supply 1 for each employee e, and
- a node j with demand D_j for each task j.

Furthermore, there is a directed arc (e, j) with costs c_{ej} if and only if $j \in Q_e$. □

The next theorem is an immediate consequence of the previous lemma.

Theorem 1 *The following decision version of the mathematical model in Section 2.1 in which task changes are allowed can be solved in polynomial time: Can a set of employees cover the given demand for the tasks?*

Proof Let $t_1 = 0 < t_2 < \cdots < t_r$ be all time instances where the demand profiles of tasks or the availability of employees are changing. For each time interval $I_k = [t_k, t_{k+1}[$ $(k = 0, \cdots, r-1)$ let A_k be the set of employees available in the interval I_k. To solve the overall decision problem one can solve for each interval I_k the decision problem: Can the employees in A_k cover the demand of all tasks in I_k?

Each of these decision problems has the special structure described in Lemma 1.

□

The following theorem is due to Segal ([33]).

Theorem 2 *([33]) The special case of the mathematical model in Section 2.1 in which there is only one task and for each employee e the working periods (shifts) are given by one interval (i.e. $w_e(t) = 1$ for $s_e \le t \le f_e$, and $w_e(t) = 0$ otherwise) can be formulated as a minimal cost flow problem.*

Proof Let $\mathcal{T}$ denote the set of all points t in time where either the demand $D(t)$ for employees changes or some shift begins or ends. Denote by $t_1 = 0 < t_2 < \cdots < t_s$ the ordered sequence of all elements in $\mathcal{T}$.

The network (V, A) is constructed as follows. There is a node $i \in V$ for each time point t_i and there are two types of directed arcs:

- requirement arcs $(i, i+1)$ for $i = 1, \cdots, s-1$ with the demand $D(t_i)$ for employees in the interval $[t_i, t_{i+1}[$ as lower capacity. Associated with these arcs are an upper capacity equal to plus infinity and costs equal 0.
- arcs $(i, j)_e$ for each shift of each employee e from node i representing the end of the shift to node j representing the starting time of the shift. These arcs have a lower capacity 0 and the costs are defined by c_e. The upper capacity is equal to 1.

To solve the problem one has to calculate a minimum cost circulation which satisfies the lower and upper capacities on the arcs. □

The next result is due to Dantzig & Fulkerson [16].

Theorem 3 *([16]) The following problem can be formulated as a minimal cost network flow problem: All employees are available during the whole time horizon $[0, T]$ and each task j has to be processed by exactly one employee during the time interval $\left[s_j, f_j\right[\subseteq [0, T]$ without interruption. All employees are qualified for all tasks. One has to assign a minimal number of employees to tasks such that all tasks are performed (if this is possible).*

Proof To formulate the problem as a minimum cost network flow problem (see Ahuja et al.[1], p.296) consider the following network (V, A). Associated with each task j are two vertices v_j^1 and v_j^2 which correspond to the starting time s_j and the finishing time f_j of task j, respectively. Additionally, there are two dummy vertices denoted by 0 and $*$. The set A contains the following arcs:

- arcs $\left(v_j^2, v_k^1\right)$ for tasks j, k if $f_j \le s_k$, i.e. if task k can be processed after task j,
- arcs $\left(0, v_j^1\right)$ and $\left(v_j^2, *\right)$ for all tasks j,and
- a return arc $(*, 0)$.

With the exception of the return arc $(*, 0)$ all arcs have a lower capacity 0, an upper capacity 1, and cost 0. Arc $(*, 0)$ has a lower capacity 0, the upper capacity is equal to the number of employees, and the cost is equal to 1. Associated with the vertices v_j^1 and v_j^2 is a demand and supply, respectively, of one unit. Finally vertices 0 and $*$ are neither supply nor demand nodes. One has to find a minimal cost flow (if it exists).

Given an optimal flow the shifts of the minimal number of employees can be constructed easily.

The result described in the following theorem has been used in the solution procedure proposed in [31] for solving Problem 3.

Theorem 4 *([31]) For the intraday part of Problem 3, a feasible assignment of tasks to employees (if it exists) can be calculated by a maximum flow algorithm if the shifts of the employees are given.*

Proof We assume that the day is fixed and drop the index d in the data. Let $\mathcal{T}$ be the set of all R_j- and D_j- values, and all block starting and finishing times for all employees working on the day (blocks are sets of maximal length of consecutive working periods of a shift). Denote by $t_1 < t_2 < ... < t_s$ the ordered sequence of all elements in $\mathcal{T}$.

The network (V, A) can be constructed as follows. The set V of nodes consists of

- all tasks j,
- all intervals$[t_i, t_{i+1}[$ $(i = 1, \cdots, s-1)$, and
- a source s and a sink t.

There are three different types of directed arcs:

- arcs (s, j) with upper capacity p_j,
- arcs $([t_i, t_{i+1}[, t)$ with upper capacity $(t_{i+1} - t_i)N_i$ where N_i is the number of employees available in time period $[t_i, t_{i+1}[$,
- there is an arc between a task node j and an interval node $[t_i, t_{i+1}[$ if and only if $[t_i, t_{i+1}[\subseteq [R_j, D_j[$. The upper capacity of this arc is $t_{i+1} - t_i$.

The network is shown in Figure 1.

$$\begin{array}{ccccccc} & & \text{tasks} & & \text{intervals} & & \\ & \leq p_j & \vdots & \leq t_{i+1} - t_i & \vdots & \leq (t_{i+1} - t_i)N_i & \\ \text{s} & \longrightarrow & \text{j} & \xrightarrow[\text{iff } [t_i, t_{i+1}[\, \subseteq [R_j, D_j[]{\hspace{6em}} & [t_i, t_{i+1}[& \xrightarrow{\hspace{5em}} & \text{t} \\ & & \vdots & & \vdots & & \end{array}$$

Fig. 1 Network for the assignment of tasks to employees

A flow in an arc $(j, [t_i, t_{i+1}[)$ may be interpreted as working time assigned to task j in the interval $[t_i, t_{i+1}[$.

There exists a feasible task assignment if and only if the value of a maximal flow is equal to $\sum_{j=1}^{m} p_j$.

If there is a maximal flow with this property then in each task node j the processing time p_j is distributed to the time intervals $[t_i, t_{i+1}[$ in which j can be processed and the time j processed in $[t_i, t_{i+1}[$ cannot exceed $t_{i+1} - t_i$. Furthermore, due to the flow-balance constraints in the interval nodes $[t_i, t_{i+1}[$ the sum of these processing times cannot exceed $(t_{i+1} - t_i)N_i$. It is well known (see e.g. [11] p. 108) that under these conditions it is possible to process the parts of tasks assigned to $[t_i, t_{i+1}[$ by N_i employees if task changes are allowed. □

In the problem formulated by Theorem 4 the role of tasks and employees can be switched if the breaks are ignored. This leads to the following problem which again can be formulated as a maximum network flow problem: given the total demand $D(t_i)$ of all tasks in the time interval $[t_i, t_{i+1}[$, find a feasible shift assignment covering the demands under the restriction that employee e has to work at least m_e time units in the interval $[S_e, F_e[$. Again we assume that task changes are allowed. The corresponding network is shown in Figure 2. The label $\geq m_e$ on arcs (s, e) indicate that m_e is a lower bound for the flow in these arcs. The labels on the other arcs indicate upper bounds for the flows. The demand can be covered if and only if there exists a maximum flow with value $\sum_{i=1}^{s-1} D(t_i)$.

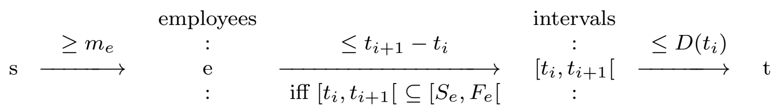

Fig. 2 Network for assigning employees to tasks

3.2 NP-complete Cases

In this section special cases of the problems introduced in Section 2.3 are described which are already NP-hard if formulated as optimization problems or NP-complete if formulated as decision problems.

Special Case 1
There is only one task which needs one employee during each period $t = 0, \cdots, T-1$ where $T = 3q$, and q is the number of employees available. The working pattern of each employee consists of exactly 3 time periods which need not to be consecutive.

Can the q employees cover the demand of the task?

Garey & Johnson show (e.g. [24], p. 243) that there is a reduction from the exact covering by 3-sets problem (X3C). This result implies that the problem to minimize the number of employees needed to perform only one task is already NP-hard.

Special Case 2
Task j needs exactly one employee for a time interval $[s_j, t_j[\subseteq [0, T]$. Each employee is available the whole planning horizon $[0, T]$ but cannot perform all tasks.

Can q employees cover the demand of the task?

The problem is NP-complete even if task changes are allowed. This has been shown by Kroon et al. [27] by a reduction from the 3-dimensional matching problem. By Theorem 3, the problem is polynomially solvable if each employee can perform all tasks.

Special Case 3
There are two tasks. Each task must be performed during the whole planning horizon $[0, T]$ by one employee in each period. There are n employees $e = 1, \cdots, n$. Any working pattern $(w_e(t))_{t=0}^{T-1}$ with $\sum_{t=0}^{T-1} w_e(t) = m_e$ can be assigned to employee e. Each employee can perform each task. Furthermore, $\sum_{e=1}^{n} m_e = 2T$ holds.

Are there working patterns for the employees such that both tasks are covered and each employee performs only one task (no task changes)?

Thus, the problem of minimizing the task changes is NP-hard even when each employee can perform each task.

NP-hardness follows from the fact that the problem is equivalent to the partitioning problem. Indeed, both tasks can be covered if and only if there is a subset $I \subseteq \{1, \cdots, n\}$ with $\sum_{e \in I} m_e = T$. In this case the employees $e \in I$ can be assigned to one task while the remaining employees are assigned to the other task.

4 Concluding Remarks

Personnel scheduling problems have attracted significant research attention. There was a lack of, however, a general mathematical model which accommodates various characteristics in the problem. In this paper we developed such a general mathematical model and specific models for personnel scheduling problems. Polynomially solvable and NP-hard cases are presented. In [21], it is said that

"New models need to be formulated that provide more flexibility to accommodate individual workplace practices. This can then lead to the development of more general algorithms that will be more robust to changes in the rostering requirements."

Our aim is to underpin the theory of personnel scheduling, which unlike in traditional scheduling, needs theoretical studies on models and complexity.

Personnel scheduling problems can be formulated as integer linear programs. For some small sized instances, LP-solvers can be used to solve the problem. Complex problems can be solved by heuristics which combine local search and network flow techniques.

The problem of assigning shifts to employees and employees to tasks to cover the demand can be efficiently solved by network flow algorithms if task changes are allowed. This can be exploited in heuristics for personnel scheduling problems. A side effect is that employees have to switch between tasks (working places) during their shifts. These switches may be unavoidable. Unfortunately the problem of minimizing the number of task changes is NP-hard. Heuristics which

- assign feasible shifts to employees,
- construct (directly) schedules which allow task changes, and
- take care of the number of task changes

need to be developed and numerically tested. The results presented here could be useful for such investigations.

References

1. Ahuja, R.K., Magnanti, T.L. and Orlin J.B. (1993), Network Flows: Theory, Algorithms, and Applications. Prentice Hall.
2. Bailey, J. (1985), Integrated days off and shift personnel scheduling. Comput. & Indus. Engng, 9(4): 395–404.
3. Bailey, J., Alfares, H. and Lin, W.Y. (1995), Optimization and heuristic models to integrate project task and manpower scheduling. Computers ind. Engng, 29(1–4): 473–476.
4. Bard J.F. (2004), Selecting the appropriate input data set when configuring a permanent workforce. Computers & Industrial Engineering, 47: 371-389.

5. Bard, J. F. and Purnomo, H. W. (2004), Real-time scheduling for nurses in response to demand fluctuations and personnel shortages. In Trick, M. & Burke, E. (Eds.), Proceedings of the 5th international conference on the practice and theory of automated timetabling, pp. 67–87. Pittsburgh.
6. Bard, J.F., Binici, C. and deSilva A.H. (2003), Staff scheduling at the United States Postal Service. Computers and Opertions Research, 30: 745-771.
7. Bilgin, B., De Causmaecker, P., Rossie, B. and Vanden Berghe, G. (2008), Local search neighbourhoods to deal with a novel nurse rostering model. Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2008), 18-22 August 2008, University of Montreal, Canada.
8. Blochliger, I. (2004) Modeling staff scheduling problems. A tutorial. European Journal of Operational Research, 158: 533-542.
9. Bodin, L., Golden, B., Assad, A. and Ball, M. (1983), Routing and scheduling of vehicles and crews – The state of the art. Computers and Opertions Research, 10(2): 63–211.
10. Brucker, P. and Knust, S. (2006), Complex Scheduling, Springer, Berlin.
11. Brucker, P. (2007), Scheduling algorithms, Springer, Berlin.
12. Brucker, P., Qu, R., Burke, E., Post, G. (2005), A decomposition, construction and post-processing approach for a specific nurse rostering problem. Proceedings of the 2nd Multidisciplinary Conference on Scheduling: Theory and Applications (MISTA 2005), 18-21 July: 397–406.
13. Burke E. K., De Causmaecker P., Vanden Berghe G., and Van Landeghem H. (2004), The state of the art of nurse rostering. Journal of Scheduling, 7(6): 441–499.
14. Caprara, A., Monaci, M., Toth, P. (2003), Models and algorithms for a staff scheduling problem. Mathematical Programming, Ser. B 98: 445–476.
15. Dantzig, D. (1954) A comment on Edie's traffic delay at toll booths, Operations Research, 2: 339–341.
16. Dantzig, M.L. and Fulkerson, D.R. (1954), Minimizing the number of tankers to meet a fixed schedule. Naval Res. Logist. 1: 217–222.
17. De Causmaecker, P., Demeester. P., Vanden Berghe, G., Verbeke, B. (2005), Analysis of real-world personnel scheduling problems. Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling, 18th August - 20th August 2004, Pittsburgh, PA USA: 183–198.
18. De Causmaecker P., Vanden Berghe G. (2010), Towards a reference model for timetabling and rostering. To appear at Annals of Operations Research, DOI: 10.1007/s10479-010-0721-2, 2010.
19. Dowsland, K. A. (1998), Nurse scheduling with tabu search and strategic oscillation. European Journal of Operational Ressearch, 106: 393–407.
20. Ernst, A.T., H. Jiang, M. Krishnamoorthy, B. Owens and D. Sier (2004), An Annotated Bibliography of Personnel Scheduling and Rostering. Annals of OR, 127: 21–144.
21. Ernst, A. T., Jiang H., Krishnamoorthy M., and Sier D. (2004), Staff scheduling and rostering: A review of applications, methods and models. European Journal of Operational Research, 153(1): 3-27.
22. Ertogral, K. and Bamuqabel B. (2008), Developing staff schedules for a bilingual telecommunication call center with flexible workers. Computers & Industrial Engineering, 54: 118-127.
23. Franz, L.S., Baker, H.M., Leong, G. K. and Rakes T.R. (1989), A mathematical model for scheduling and staffing multiclinic health regions. European Journal of Operational Research, 41: 277–289.
24. Garey, M.R., Johnson, D.S. (1979), Computers and Intractability: A Guide to the Theory of NP-completeness. W.H. Freeman, San Francisco, 1979.
25. Gutjahr, W.J., Katzensteiner, S., Reiter, P., Stummer, C. and Denk, M. (2010), Multi-objective decision analysis for competence-oriented project portfolio selection. European Journal of Operational Research, 205(3): 670–679.
26. Haspeslagh, S., De Causmaecker, P., Stolevik, M. and Schaerf, A. (2010), First International Nurse Rostering Competition 2010. Proceedings of the 8th International Conference for the Practice and Theory of Automated Timetabling, Northern Ireland, 10th - 13th August, 2010, 498–501.
27. Kroon, L.G., Salomon, M., Van Wassenhove, L.N., (1997), Exact and approximation algorithms for the tactical fixed interval scheduling problem. Operations Research, 45: 624–638.
28. Lau H.C., (1996), On the complexity of manpower shift scheduling. Computers and Operations Research, 23(1): 93–102.

29. Merciera, A., Cordeau, J.-F. and Soumis, F. (2005), A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. Computers & Operations Research, 32: 1451-1476.
30. Orden A. (1956), The Transhipment Problem. Management Science, 2(3): 276–285.
31. Robinson, R., Sorli, R., Zinder, Y. (2005), Personnel scheduling with time windows and preemptive tasks. Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling, 18th August - 20th August 2004, Pittsburgh, PA USA: 561–566.
32. Rodrigues, M.M., de Souza, C.C. and Moura, A.V. (2006), Vehicle and crew scheduling for urban bus lines. European Journal of Operational Research, 170(3): 844–862.
33. Segal, M. (1974), The operator scheduling problem: a network flow approach. Operations Research 22: 808–823.
34. Tiwari, V., Patterson, J.H., Mabert, V.A. (2009), Scheduling projects with heterogeneous resources to meet time and quality objectives. European Journal of Operational Research, 193: 780-790.
35. Weide, O., Ryan, D. and Ehrgott M. (2010), An iterative approach to robust and integrated aircraft routing and crew scheduling. Computers & Operations Research, 37: 833–844.