

School of Computer Science and Information Technology  
University of Nottingham  
Jubilee Campus  
NOTTINGHAM NG8 1BB, UK

Computer Science Technical Report No. NOTTCS-TR-SUB-0906301611-4177

## **Personnel Scheduling: Models and Complexity**

*Peter Brucker, Rong Qu and Edmund Burke*

*First released: June, 2009*

© Copyright Peter Brucker, Rong Qu and Edmund Burke

In an attempt to ensure good-quality printouts of our technical reports, from the supplied PDF files, we process to PDF using Acrobat Distiller. We encourage our authors to use outline fonts coupled with embedding of the used subset of all fonts (in either Truetype or Type 1 formats) except for the standard Acrobat typeface families of Times, Helvetica (Arial), Courier and Symbol. In the case of papers prepared using TEX or LATEX we endeavour to use subsetted Type 1 fonts, supplied by Y&Y Inc., for the Computer Modern, Lucida Bright and Mathtime families, rather than the public-domain Computer Modern bitmapped fonts. Note that the Y&Y font subsets are embedded under a site license issued by Y&Y Inc.

For further details of site licensing and purchase of these fonts visit <http://www.yandy.com>

---

# Personnel Scheduling: Models and Complexity

Peter Brucker · Rong Qu · Edmund Burke

**Abstract** Due to its complexity and challenging features, personnel scheduling has been heavily investigated in decades. However, there is the lack of study on models and complexity in these important problems. In this paper, we review and present recent general and specific models in the personnel scheduling literature. In particular, we address the complexity issues and present polynomial solvable and NP-complete cases for these problems.

## 1 Introduction

De Causmaecker et al. [5] classified companies according to different personnel scheduling problems. The corresponding scheduling problems lead to different models. The classification is:

- permanence centered planning
- fluctuation centered planning
- mobility centered planning
- project centered planning.

The number of personnel needed, for example, for police services and hospitals, is usually defined in advance. Corresponding personnel scheduling problems are called *permanence centered*. On the contrary for warehouses or distribution centers, as well as for call centers and fast food restaurants, the personnel planning is based on *fluctuating* demand. *Mobility centred* planning occurs when duties involve transportation. Examples are duties of health and safety boards or in connection with home health care. Also transportation companies face similar planning problems. *Project centered*

---

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

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 a widely recognised challenging problem in the literature, personnel scheduling has attracted significant research attention in both research and practice. For example, as just one of personnel scheduling problems, nurse rostering has been heavily investigated in the last few decades by using a variety of techniques from heuristic algorithms to exact methods [3].

In this paper a general model which covers permanence and fluctuation centred planning is presented. As an example for 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 employees have to work on which days of a period of several days. Such decisions have an influence on the ability to cover the demand for employees in each period of each day. For the development of suitable methods to solve personnel scheduling problems, it is useful to get some insight on the complexity of special cases. For this purpose we identify cases which are either NP-hard or polynomially solvable. All polynomially solvable cases can be solved efficiently by network flow techniques. These techniques can be used as subroutines in heuristics which solve more general problems.

This paper is organized as follows. The models are presented in Section 2, followed by the complexity results in Section 3. In Subsection 3.1 the network flow formulations are presented while Section 3.2 contains the NP-completeness results. The last section contains some conclusions.

## 2 Models

### 2.1 A General Model

A general personnel scheduling problem can be formulated as follows.

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

There are  $n$  employees  $e = 1, \dots, n$ . Associated with each employee  $e$  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  $w_e(t) = 1$ .

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 a subset of employees  $e$  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
- relevant assignment costs are minimized.

Additional so called "soft constraints" may be imposed onto the working pattern in  $P_e$ . In this case a penalty  $u(\pi)$  may be defined which measures how much pattern  $p$  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 general 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. 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 the model 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  $[L_j, R_j] \subseteq [0, T]$  with  $R_j - L_j \geq p_j$ , i.e.  $D_j(t) \in \{0, 1\}$  for  $t = L_j, L_j + 1, \dots, R_j - 1$  where  $\sum_{t=L_j}^{R_j-1} D_j(t) = p_j$ .

In the next section special cases of this general model are presented. Problem 1 is a nurse rostering problem where one task (shift) is scheduled 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.

## 2.2 Some Special Problems

### 2.2.1 Problem 1([2]): 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. The following table taken from ([2]) shows the demand for one type of nurses during a week.

Shift type	Start time	End time	Demand						
			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.

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 60 hours 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)
8. A night shift after an early shift should be avoided. (penalty 1)

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 general 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.

### 2.2.2 Problem 2([4]): A problem with restricted task changes

Another problem has been formulated as follows.

There is a planning horizon of  $T$  days  $d = 1, \dots, T$ .  $n_d$  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 pattern 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 has to perform.

### 2.2.3 Problem 3([8]): 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 intra-day scheduling problem is a problem with feasible 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  $j = 1, \dots, n$ . 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. Also interruptions and later processings of a task are possible. However, the total processing of task  $j$  must be equal to  $p_j^d$ .

*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.

### 2.2.4 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 task which can be performed by the employee.

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 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 model has 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.

### 3 Complexity

#### 3.1 Polynomially Solvable Cases

Next we describe the special case of the general model which are polynomially solvable. These problems can be formulated in connection with the first and second parts of a decomposition of Problem 4 in Section 2. However, they are also of interest by their own.

**Theorem 1** *The special case of the general model in which each employee is available in all periods, i.e.  $w_e(t) = 1$  for all  $t = 0, \dots, T - 1$  can be formulated as a minimal cost network flow problem.*

*Proof* The problem can be formulated as a transshipment problem. The underlying network has

- a node  $e$  with supply 1 for each employee  $e$ , and
- a node  $j$  with demand  $d_j = \max_{t=1}^{T-1} D_j(t)$  for each task  $j$ .

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

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

**Theorem 2** *The following decision version of the general model 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  $t_1 = 0 < t_2 < \dots < t_r = T$  be the time instances where the demand profiles of tasks or the availability of employees are changing. For each time interval  $I_k = [t_k, t_{k+1}[$  ( $k = 0, \dots, r-1$ ) 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 is a special case of the general model in which only those employees are considered which are available in the interval  $I_k$ .  $\square$

The following theorem is due to Segal ([9]).

**Theorem 3** *The special case of the general model 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  $s_e \leq t \leq f_e$ , and  $w_e(t) = 0$  otherwise) can be formulated as a minimal cost flow problem.*

*Proof* Let  $T$  be 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  $t_1 = 0 < t_2 < \dots < t_s = T$  the ordered sequence of all elements in  $T$ .

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  $i = (1, \dots, s-1)$  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  $(i, j)$  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 the number of employees which have a corresponding shift.

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

Problem 3 can also be formulated as a network flow problem.

**Theorem 4** ([8]) *For Problem 3, a feasible assignment of tasks to employees (if it exists) can be calculated by a maximum flow algorithm.*

*Proof* Because the day is fixed we drop the index  $d$  in the data. Let  $T$  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  $t_1 < t_2 < \dots < t_s$  the ordered sequence of all elements in  $T$ .

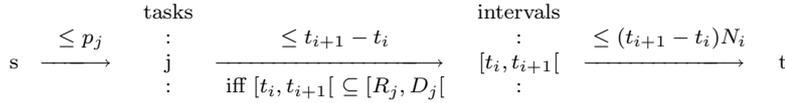
The network  $(V, A)$  can be constructed as follows. The set  $V$  of nodes consists of

- all tasks  $j$ ,
- all intervals  $[t_i, t_{i+1}[$  ( $i = 1, \dots, s-1$ ), 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.



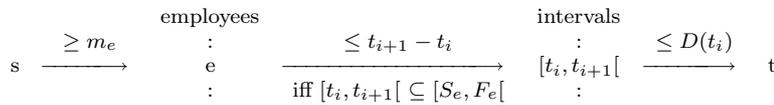
**Fig. 1** Network for the assignment of tasks to employees

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  $\sum_{j=1}^n p_j$ .

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. [1] 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 preemption is allowed.  $\square$

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 flows 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  $\sum_{i=0}^s D(t_i)$ .



**Fig. 2** Network for assigning employees to tasks

### 3.2 NP-complete Cases

In this section special cases of the problems introduced in Section 2.2 which are already NP-hard are described. Furthermore, corresponding reductions are indicated and references are given.

#### Special Case 1

There is only one task which needs one employee on each period  $t = 0, \dots, T - 1$  where  $T = 3q$ . The working pattern of each employee consists of exactly 3 time periods which need not to be consecutive.

*Does a subset of  $q$  employees exist which cover the demand of the task?*

Garey & Johnson show (e.g. [6], p. 243) that there is a reduction from the exact covering by 3-sets problem (X3C).

#### 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 all tasks?*

The problem is NP-complete even if task changes are allowed. This has been shown by Kroon et al. [7] by a reduction from the 3-dimensional matching problem.

#### 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  $e = 1, \dots, n$ . Any  $w_e(t)_{t=0}^{T-1}$  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  $I \subseteq \{1, \dots, n\}$  with  $\sum_{e \in I} m_e = T$ .

We conclude that

- One has to develop heuristics to solve realistic problems.
- Good heuristics may use the fact that some subproblems are polynomially solvable. Such subproblems have been discussed in the last section.

## 4 Concluding Remarks

In this paper we described a general and specific models for personnel scheduling problems. Polynomially solvable and NP-hard cases are presented.

Personnel scheduling problems can be formulated as integer linear programs. For some small sized instances, LP-optimizers 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 working place changes is NP-hard. Heuristics which assign feasible shifts to employees and construct (directly) schedules also allow task changes but also take care of the number of changes needed to be developed and numerically tested. The results presented here could be useful for such investigations.

## References

1. Brucker, P. (2007), *Scheduling Algorithms*, Springer, Berlin.
2. 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 Multi-disciplinary Conference on Scheduling: Theory and Applications (MISTA 2005)*, 18-21 July: 397-406.
3. 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.
4. Caprara, A., Monaci, M., Toth, P. (2003), Models and algorithms for a staff scheduling problem, *Mathematical Programming, Ser. B* 98: 445-476.
5. De Causmaecker, P., Demeester, P., Vanden Berghe, G., Verbeke, B. (2005), Analysis of real-world personnel scheduling problems, In: E. K. Burke and M. Trick (editors), *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling*, 18th August - 20th August 2004, Pittsburgh, PA USA: 183-198.
6. Garey, M.R., Johnson, D.S. (1979). *Computers and Intractability: A Guide to the Theory of NP-completeness*. W.H. Freeman, San Francisco, 1979.
7. 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.
8. Robinson, R., Sorli, R., Zinder, Y. (2005), Personnel scheduling with time windows and preemptive tasks, In: E. K. Burke and M. Trick (editors), *Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling*, 18th August - 20th August 2004, Pittsburgh, PA USA: 561-566
9. Segal, M. (1974), The operator scheduling problem: a network flow approach, *Operations Research* 22: 808-823.