Personnel scheduling with time windows and preemptive tasks

Robert Robinson, Ronald Sorli, and Yakov Zinder

Dept. of Mathematical Sciences, University of Technology, Sydney
{Yakov.Zinder,Ron.Sorli}@uts.edu.au

1 The problem

The personnel scheduling problem being considered is concerned with a situation which often arises in a flexible working environment where employees, for example casual staff, are available only at certain times, which may differ from employee to employee and from day to day. The planning horizon is seven days. Operating hours are such that there is sufficient rest time between days. This condition holds for example if work is not allowed during the night. For each day, a set of tasks is specified, which are to be completed during this day. All parameters are measured in some time units, which following the tradition of the literature on personnel timetabling will be referred to as periods. For example, 15-minute periods are commonly used in practice. It is assumed that all employees are equally capable of performing any task. Each employee can only perform one task at a time and at most one employee can be assigned to a task at a time. If an employee is assigned to a task for a particular period (time unit), then this employee performs this task from the beginning till the end of this period.

Each employee e is available on each day d only during some time interval (time window) $[S_{ed}, F_{ed}]$, where S_{ed} and F_{ed} are the earliest and latest periods when employee e is available. Employee e may not be available on some day at all, then the corresponding time window is not specified and the employee is considered to be having a day off. A shift of employee e on a working day d is a time interval $[V_{ed}, W_{ed}]$ such that $S_{ed} \leq V_{ed} < W_{ed} \leq F_{ed}$ and $W_{ed} - V_{ed} \geq m_e$, where m_e is the given minimal shift length. In practice, m_e often depends on the employee's status – casual or full-time, or on the employee's age.

During each period within a shift, an employee either performs a task, or is idle, or has a break. There are two types of breaks: a short rest break and a longer meal break. Each shift contains a possible meal break and at most two rest breaks depending on the shift length. When scheduling breaks, there are minimal and maximal restrictions on the period between:

- the start of the shift and the start of the first break;
- the end of the last break and the finish of the shift;
- the end of one break and start of the next.

All breaks are not preemptive.

Further, there are requirements regarding days off and restrictions on the total shift duration over the week. It is desirable that each employee has at least two days off per week. In addition, the total shift duration accumulated over a week has a minimal and a maximal number of periods. These two thresholds may also vary from employee to employee.

Each task j has a duration P_j and a time window $[R_j, D_j]$ within which the task is to be completed, where R_j is the earliest period in which task j can be processed and D_j is the latest period in which task j can be processed. In the literature on scheduling, the parameters R_j and D_j are often referred to as a task's release time and deadline respectively. A distinctive feature of the considered personnel scheduling problem is the assumption that tasks are preemptive – different employees can perform the same task during different periods and any two periods in which a task is processed may be separated by a period when nobody is assigned to this task. Preemptions do not increase the task's duration. Thus, task j is completed when its total processing time becomes equal to P_j .

A feasible schedule is an assignment of shifts to employees and a pattern of processing of the tasks which results in the completion of all tasks, the scheduling of all entitled breaks and the fulfillment of the minimal shift duration m_e . The objective here is to find a feasible schedule that minimizes the total cost over the planning horizon. The cost is exclusively employee wages for the shifts where meal breaks are unpaid but rest breaks are compensated. Each employee has a given base rate of pay. In addition each period has a penalty weight with respect to the base rate. Thus, an employee's wage for a single day is calculated as the sum of (i) the product of the shift duration, less the meal break duration, and the employee's base rate of pay and (ii) the sum of the periods' penalties over the payable periods. If the duration of a shift exceeds a given limit M, then the employee is paid at an overtime rate for all periods in excess of M. If an employee is not given at least two days off, then there is additional pay.

Although related problems were considered in the literature (see for example [1–3]), to the authors' knowledge, none is concerned with preemptive tasks. The presence of preemptive tasks allows a solution algorithm which combines the ideas of tabu search and network optimization.

2 Algorithm

The meta-heuristic tabu search was chosen as the principal optimization algorithm [4, 5]. An innovation is the use of a network model within the tabu search framework to check the eligibility of any proposed schedule, where a schedule is eligible if it completes all tasks.

2.1 Tabu search

An initial solution is constructed by scheduling all employees for their maximum availability without any breaks. The cost of this, possibly infeasible, initial solution will be an upper bound on the value of the objective function. Subsequently, other schedules are constructed by making a change (a move) to one shift of one employee at a time.

Five types of move are proposed:

- delete shift;
- create shift;
- change start – by changing V_{ed} but not W_{ed};
- change finish – by changing W_{ed} but not V_{ed};
- reassign shift – by changing $[V_{ed}, W_{ed}]$ but not $W_{ed} - V_{ed}$.

A neighbourhood of a schedule is defined as the set of schedules which is constructed by applying a single type of move to a schedule.

The objective function not only includes the total cost, as described above, but also an artificial penalty cost which enables the withholding of entitled breaks and the violation of the total shift durations over the week.

To prevent cycling of schedules and being trapped at a local optimum, a bounded tabu list of altered shifts is maintained. If an altered shift is in the tabu list, it cannot be considered as an admissible move unless the schedule results in an objective function value that is better than *any* already obtained (an overall best). This is referred to as the global aspiration criterion.

For each iteration of the tabu search, all possible moves are investigated. The cost of a proposed move is calculated. If this cost is better than the best-so-far then the eligibility of the proposed move is checked by seeing whether the move results in a feasible schedule.

Several stopping criteria for the tabu search are implemented. These include exceeding the limit on the overall number of iterations performed and exceeding the limit on the number of iterations since the overall best was last updated (an indication of the likelihood of further improvements). There is also the possibility that no eligible move exists and so the search must terminate.

2.2 Network model

Using the ideas of scheduling theory [6], the eligibility of a proposed schedule is checked by solving a series of maximum flow, bipartite network problems, one for each day in the planning horizon. In the bipartite network one set of nodes corresponds to the set of tasks. The other set represents time intervals where a time interval is defined as the longest period of time during which there is no change in the status quo.

For each task node there is an arc from a source node to the task node with an upper bound on capacity given by the task duration. For each interval node there is an arc from it to a sink node with an upper bound on capacity given by the product of the time interval duration and the number of employees available during that interval. Finally, there will be a task-to-interval arc whenever the task can be processed during the interval. Since each task requires only a single employee, the upper bound on capacity of such an arc is the corresponding interval duration.

The Ford and Fulkerson out-of-kilter algorithm [7] is used to obtain a maximum flow. The eligibility of the proposed schedule is verified by checking whether the maximum flow found equals the sum of the task durations. If needed, the actual processing of tasks over time is given by the flows in the task-to-interval arcs of the network. To find an allocation of particular employees to particular tasks requires additional processing (e.g. using McNaughton's algorithm [8]).

Each move results in a limited transformation of the current network. Consequently, the solution for the current network provides useful information for the process of finding a solution for the transformed network which speeds up the checking of eligibility.

3 Computational experiments

The preliminary computational experiments, aimed at an initial testing of the idea, used a straightforward procedure for problem generation. This procedure generated fairly small problems which allowed for manual verification of each operation and also the subsequent comparison of the solution produced according to the presented tabu search algorithm and an optimal one, obtained by a mixed integer linear programming model implemented in CPLEX. Larger problems were also generated by scaling up the small ones: by repeatedly doubling the number of tasks and employees, but keeping the distributions of task duration and window length unchanged. It was found that if the length of the tabu list is increased proportionally, the tabu search algorithm demonstrates similar behaviour.

The success of preliminary experiments justified the main, ongoing, round of computational experiments. For this main round a different method is employed for generating random problems which are known to be feasible. This was achieved by using a three stage generation process. The input parameters for the process are the number of employees and parameters appropriate for the probability distributions utilized in the process of data generating.

- At the first stage, the method generates shifts for each employee by selecting randomly working days, the left endpoint of an availability window in each working day, the duration of each of these windows, the shift starting time within each window, the duration of each shift, and if the shift duration requires, the starting times and durations of all necessary breaks.
- At the second stage, the active time of each shift is split randomly into several disjoint intervals, where the term "active time" refers to those parts of the shift which are not breaks. Again randomly each disjoint interval is designated to be either idle time or working time.
- At the third stage, the disjoint working time intervals in each day are randomly liked together to form tasks. So each task is a chain of several working time intervals. This process results in generating, for each task, a time window and a duration. The task time window is specified by randomly selecting left and right endpoints, where the left endpoint does not exceed the left endpoint of the first working time interval, constituting the considered task, and

the right endpoint is not less than the right endpoint of the last working time interval of this task.

Obviously, only information on employees' and tasks' time windows and tasks' durations form the input data for the tabu search routine. The described above method produces as a by-product a feasible schedule, which is used as a benchmark in computational experiments.

References

1. Chiarandini, M., Schaerf, A., Tiozzo, F.: Solving employee timetabling problems with flexible workload using Tabu search. In: Proceedings of the 3rd International Conference on the Practice and Theory of Automated Timetabling (PATAT-2000), Konstanz, Germany (2000) 298–302
2. Mattfeld, D.C., Branke, J.: Task scheduling under gang constraints. In Kendall, G., Burke, E., Petrovic, S., eds.: Proceedings of the 1st Multidisciplinary International Conference on Scheduling: Theory and Applications, MISTA 2003. Volume 2. (2003) 506–522
3. Schaerf, A., Meisels, A.: Solving employee timetabling problems by generalized local search. In: Proceedings of the 6th Italian Conference on Artificial Intelligence (AIIA-99). (1999) 493–502
4. Glover, F.: Tabu search – Part I. ORSA Journal on Computing **1** (1989) 190–206
5. Glover, F., Laguna, M.: Tabu Search. Kluwer Academic Publishers, Dordrecht (1997)
6. Brucker, P.: Scheduling Algorithms. Third edn. Springer-Verlag, Berlin (2001)
7. Ford, L., Fulkerson, D.: Flows in Networks. Princeton University Press, N.J. (1962)
8. McNaughton, R.: Scheduling with deadlines and loss functions. Management Science **6** (1959) 1–12