A pattern-based staff scheduling model and applications for part-time employees

Wei Wu · Atsuko Ikegami

Abstract In this paper, we introduce a mathematical programming model based on one-day duty (task patterns) of individual staff members for solving a staff scheduling problem. Our model can reflect various types of service and management policies as well as consider staff preferences.

We propose two heuristic methods: an iterated local search algorithm and a very large-scale neighborhood search approach. For the very large-scale neighborhood search, we propose a dynamic programming (DP) method that aims to find the most improved schedule when the preference of a staff member is changed. Our DP method can also be used in the rescheduling stage. The computational results show that the model and the proposed algorithms perform well for real-world instances in Japan.

Keywords Staff scheduling · Shift scheduling · Constrained optimization · Local search

1 Introduction

Scheduling staff in the service industry often takes considerable time and resources. Developing general-purpose scheduling software is also hard due to various constraints which depend on the type of the service and the management policy.

Staff scheduling has been one of the most active topics in scheduling and timetabling since the 1950s [1]. Various models and algorithms have been proposed for this problem [2].

In this paper, we focus on staff scheduling for part-time employees for medium-sized shops (10–50 staff members total). The constraints for this problem model factors such as the overall skill level of the team and consideration

W. Wu · A. Ikegami
Seikei University, Japan
Tel.: +81-0422-37-3440
Fax: +81-0422-37-3871
E-mail: wuwei@st.seikei.ac.jp

Proceedings of the 12th International Conference on the Practice and Theory of Automated Timetabling (PATAT-2018), Vienna, Austria, August 28–31, 2018

of staff preferences. According to a survey [3], a useful model for solving staff scheduling problem in Japan should consider the following points: 1. for the number of task types allowed to individual staff members for a day (upper and lower bounds); 2. length of each type of task (upper and lower bounds); 3. which task types can be split; 4. minimum working time for each part of a split task; 5. forbidden task pairs for a staff member in a day; 6. task pairs that cannot be performed in sequence by one person; 7. qualification/skill-level required to perform a task; 8. working time preferences for each staff member for each day; 9. number of staff who have a specific skill-level for each time that each task is needed (upper and lower bounds); 10. the maximum number of staff to be assigned during one day; 11. working hours of each staff member in the whole period (upper and lower bounds);

2 Pattern-based model

In this paper, a pattern represents an one-day duty for a staff member. To simplify the constraints described in Section 1, we consider an approach that first enumerates all patterns P_{shop} satisfying points 1–6 for the shop. Using this, the feasible pattern set P_i for staff member i is a subset of P_{shop} in which each pattern satisfies point 7 for member i. Similarly, a feasible pattern set P_{id} is a subset of P_i in which each pattern satisfies point 8 for member i on day d. Pattern set P_{id} is necessary for each scheduling period for each staff member; however, we can avoid generating P_{id} from scratch by using the pattern set P_i.

Using these patterns, we apply a constrained optimization model proposed by Tokunaga et al. [3] that aims to obtain a schedule that minimizes the penalty applied for violating points 9–11 (the soft constraints). The resulting model is a mixed integer programming model.

According to the terminology in [2], our problem is a combination of *duty generation* (pattern enumeration stage) and *shift scheduling* (the constrained optimization modeling stage).

3 Heuristic algorithms

3.1 Standard iterated local search

We consider a standard iterated local search (ILS) approach for the staff scheduling problem. This approach is based on *add*, *drop*, and *swap* neighborhood operations. During the add (resp., swap) operation, we add (resp., swap) a feasible pattern $p \in P_{id}$ to the current solution, while the drop operation drops a pattern from the current solution. Note that we can enhance the swap neighborhood by taking advantage of task-based demand. In our implementation, the drop and swap operators are performed simultaneously. That is, the process is repeated until no improvement is found by swapping or dropping neighborhoods. After we obtain a local optimal solution, the ILS applies a kick move (a local perturbation) to diversify the search region.

3.2 Very large-scale neighborhood search

We propose a very large-scale neighborhood search (VLSN), by considering two new neighborhoods: *day-opt* and *staff-opt*. The staff-opt (resp., day-opt) neighborhood of a solution is the set of solutions in which at most one staff member's schedule (resp., one day's schedule) differs from the base solution. Note that the sizes of these two neighborhoods can be exponential relative to the number of days (resp., staff members). We propose a dynamic programming method for searching for an improved solution in the staff-opt neighborhood. The resulting algorithm runs in pseudo-polynomial time to find a most improved solution in the staff-opt neighborhood.

The availability of part-time employees can change with little or no advance notice. Furthermore, many shops expect a rescheduling method that can quickly output a new schedule with few changes from the original schedule when changes occur. An advantage of the algorithms we propose here for solving the day-opt and staff-opt subproblems is that these same algorithms can be used for rescheduling when the requirements for a specific day or the preferences of one staff member have been changed.

3.3 Computational results

Two sets of instances, INS_1 and INS_2, are used in computational experiments. INS_1 is a dataset from a bakery chain constructing a two-week schedule for 36 staff members with 2 types of tasks and 5 types of skills; INS_2 is from a restaurant chain constructing a one-month schedule for 18 staff members with 3 types of tasks. We set the time limit to 10 minutes for both heuristic algorithms. In the pattern generation stage, we obtained 24,702 patterns and 10,824 patterns for INS_1 and INS_2, respectively.

We first examined the model in Section 2, and solved it by a mixed integer programming solver, Gurobi (Ver 7.02). The results of the heuristics show that the proposed ILS obtained a near-optimal solution (with an optimality gap $\leq 5\%$) for all instances, and the VLSN method obtained an optimal solution for all tested instances.

References

1. Ernst, A.T., Jiang, H., Krishnamoorthy, M., Owens, B., Sier, D.: An annotated bibliography of personnel scheduling and rostering. Annals of Operations Research **127**(1–4), 21–144 (2004)
2. Ernst, A.T., Jiang, H., Krishnamoorthy, M., Sier, D.: Staff scheduling and rostering: A review of applications, methods and models. European journal of operational research **153**(1), 3–27 (2004)
3. Tokunaga, T., Tanaka, Y., Takafumi, K., Yuki, K., Ikegami, A.: Development of a staff scheduling model for part-time employees and a support system. Transactions on Mathematical Modeling and Its Applications **8**, 57–65 (2015). [In Japanese]