Constructing Cyclic Staff Schedules by Iterated Local Search

Nysret Musliu

Vienna University of Technology
musliu@dbai.tuwien.ac.at

Abstract

Construction of cyclic staff schedules is of high practical relevance in a broad range of workplaces. Typically, such schedules must fulfill different hard constraints regarding the workforce requirements, sequences of shifts, and length of work and days-off blocks. We investigate the application of iterated local search to solve this problem. The current version of our algorithm has been experimentally evaluated on a set of benchmark instances from the literature.

1 Introduction

We investigate solving the rotating workforce scheduling problem. The formulation of the problem we solve in this paper is given in [5]:
Instance:

- Number of employees: n.
- Set A of m shifts (activities) : $a_1, a_2, \ldots, a_m$, where a_m represents the special day-off "shift".
- w: length of schedule. The total length of a planning period is $n \times w$ because of the cyclic characteristics of the schedules.
- A cyclic schedule is represented by an $n \times w$ matrix $S \in A^{nw}$. Each element $s_{i,j}$ of matrix S corresponds to one shift. Element $s_{i,j}$ shows on which shift employee i works during day j or whether the employee has time off. In a cyclic schedule, the schedule for one employee consists of a sequence of all rows of the matrix S. The last element of a row is adjacent to the first element of the next row, and the last element of the matrix is adjacent to its first element.
- Temporal requirements: The requirement matrix R $((m-1) \times w)$ (we use here $m-1$, because shift a_m represents the day-off), where each element $r_{i,j}$ of the requirement matrix R shows the required number of employees for shift i during day j.

The aim is to find a cyclic schedule (assignment of shifts to employees) that satisfies the requirement matrix, and the constraints about sequences of shifts not permitted to be assigned to employees, maximum and minimum length of periods of consecutive shifts, and maximum and minimum length of blocks of workdays

To generate rotating workforce schedules several methods have been proposed in the literature [1], [5], [2], [4]. In this paper we propose an iterated local search based algorithm to solve this problem. The proposed method is experimentally evaluated and compared with other existing aproaches.

2 Iterated Local Search for Cyclic Staff Scheduling

The proposed algorithm is based on the iterated local search framework [3]. Our algorithm includes the previously proposed local search which is based on the min conflicts heuristic, and the iterative process with the perturbation mechanism and the acceptance criteria. Although the local search procedure is very important, selecting the appropriate perturbation in each iteration as well as the mechanism for acceptance of solution are also very important to obtain good results using an iterative local search algorithm.

Initially our algorithm constructs a random initial solution. The iterative phase consists of the local search method, perturbation, and the acceptance criteria test that decides if the solution produced with the local search fulfils the acceptance criteria. Regarding the local search technique we apply one of methods proposed in [4] which is based on min conflicts heuristic. During each iteration a conflicted slot (or a block of slots) in schedule is selected randomly. A slot is conflicted if it appears in one of violated constraints. The selected slot is swapped with all slots in same column and the best solution is selected such that the number of conflicts is minimized. Additionally, the tabu list is used to avoid cycles during the search.

The solution returned from the local search phase is accepted for the next iteration if it fulfils the specific criteria regarding the quality of the solution. We experimented with different possibilities for the acceptance of the solution returned from the local search: (1)solution returned from the local search phase is accepted only if it is better than the best current existing solution, (2) solution returned is always accepted, (3) solution is accepted if its quality is not much worse than the best yet found solution.

During the perturbation phase the accepted solution is perturbed and the newly obtained solution is used as an input for the next call of the local search technique. We apply two different perturbation mechanisms: (1) N fixed swaps (see description of swap moves in [5]) of cells are performed randomly; (2) An adaptive perturbation mechanism that takes into consideration the feedback from the search process. The number of cells to be swapped N varies from 2 to some number x (determined experimentally), and the algorithm begins with a small perturbation. If during the iterative process (for a determined number of iterations) the fitness of the solution remains same, the size of perturbation is increased by 1, otherwise the size of N will be decreased by 1.

3 Computational Results

We report on preliminary results of described algorithm for 20 problems which were proposed previously in the literature [1]. We experimented with proposed perturbation variants and acceptance criteria, and investigated different stopping criteria for the local search. In this paper are presented the best current results. The experiments were performed with an Intel Core 2 CPU 2GHz, 1GB RAM. For each problem 20 independent runs are executed.

In Table 1 results for 20 problems are shown. Columns 3-6 present results of existing approaches from the literature regarding the time needed to reach the first solution (in [4] the average time over 10 runs is given). Note that for algorithms proposed in [1] and [2] the results for problems 4-20 are not available. Last column (ILS) presents preliminary results obtained with iterated local search (the average time over 20 runs) proposed in this paper. Iterated local search could solve all problems and gave a feasible solution in each run. Although the experiments were performed on different computers and the results are not totally comparable we can conclude from the table that the iterated local search gives promising results for the benchmark problems in the literature. Most problems could be solved in a very short amount of time and results are competitive with other techniques.

For the future work, we are planning a more extensive evaluation of parameters and investigation of the impact of each component in the iterated local search algorithm.

4 Acknowledgments

The research herein is partially conducted within the competence network Softnet Austria II (www.soft-net.at, COMET K-Projekt) and funded by the Austrian Federal Ministry of Economy, Family and Youth (bmwfj), the province of Styria, the Steirische Wirtschaftsfrderungsgesellschaft mbH. (SFG), and the

[1] http://www.dbai.tuwien.ac.at/staff/musliu/benchmarks/

Table 1: Comparison between ILS and other aproaches in literature

Example	Groups	[5] (time in sec)	[1] (sec)	[2] (sec)	[4] (sec)	ILS (sec)
1	9	0.9	73.54	3.78	0.07	0.09
2	9	0,4	310.84	0.03	0.07	0.08
3	17	1.9	457.98	10.26	0.42	0.32
4	13	1.7			0.11	0.05
5	11	3.5			0.43	0.56
6	7	2			0.08	0.17
7	29	16.1			52.79	138.27
8	16	124			0.74	0.54
9	47	>1000(?)			15.96	3.17
10	27	9.5			0.60	0.47
11	30	367			13.15	6.93
12	20	>1000(?)			1.17	1.07
13	24	>1000(?)			0.87	0.37
14	13	0.54			0.76	0.74
15	64	>1000(?)			159.04	243.99
16	29	2.44			0.54	0.38
17	33	>1000(?)			2.16	0.41
18	53	2.57			6.83	4.03
19	120	>1000(?)			75.83	22.91
20	163	>1000(?)			71.38	43.56

city of Vienna in terms of the center for innovation and technology (ZIT). Additionally, this work was partially supported by the Austrian Science Fund (FWF), project P20704-N18.

References

[1] Nagraj Balakrishnan and Richard T. Wong. A network model for the rotating workforce scheduling problem. *Networks*, 20:25–42, 1990.

[2] G. Laporte and G. Pesant. A general multi-shift scheduling system. *Journal of the Operational Research Society*, 55/11:1208–1217, 2004.

[3] Helena Lourenço, Olivier Martin, and Thomas Stützle. Iterated local search. In *Handbook of Metaheuristics*, volume 57, pages 320–353. 2003.

[4] Nysret Musliu. Heuristic methods for automatic rotating workforce scheduling. *International Journal of Computational Intelligence Research*, 2(4):309–326, 2006.

[5] Nysret Musliu, Johannes Gärtner, and Wolfgang Slany. Efficient generation of rotating workforce schedules. *Discrete Applied Mathematics*, 118(1-2):85–98, 2002.