Comparison of Shift Sequence Based and Simulated Annealing Methods for Highly Constrained Medical Staff Rostering Problems

Mindaugas Liogys [a]
Vilnius University, Institute of Mathematics and Informatics, Akademijos st. 4, Vilnius, Lithuania

Received 1 June 2012, accepted 7 July 2012

Abstract. The aim of this article is to compare two different approaches (simulated annealing and shift sequence based method) used in solving medical staff rostering problem. During comparison stage two dimensions were considered: roster quality and roster building time. Test results showed that simulated annealing method is more efficient than shift sequence based method in both ways - builds a better roster in shorter time.

Citations: Mindaugas Liogys. Comparison of Shift Sequence Based and Simulated Annealing Methods for Highly Constrained Medical Staff Rostering Problems – *Innovative Infotechnologies for Science, Business and Education,* ISSN 2029-1035 – **1(12)** 2012 – Pp. 3-6.

Keywords: Shift Sequence Based Methods; Simulated Annealing Methods; Medical Staff Rostering.
Short title: Comparison of methods.

Introduction

The common objective of medical staff rostering problem is to produce rosters with a balanced workload as well as to satisfy individual preferences as much as possible.

Many researchers are interested in medical staff rostering problem. Large number of articles have appeared presenting different approaches to this problem [1-9]. Still, not many comparisons have been carried out and they focus on the comparison of two or more approaches developed by the same authors [10]. This article is devoted to compare two different approaches (shift sequence based method and simulated annealing) in solving medical staff rostering problems.

1. Problem Formulation

The problem is that of creating monthly schedules for cardiologists at a major Lithuania hospital. These schedules have to satisfy working contracts and meet as far as possible cardiologists' requests. Working contract regulations usually are called hard constraints and personal preferences - soft constraints [9]. Hard constraints and software constraints that are considered in this problem are listed in the Table 1 and Table 2 respectively.

Table 1. Hard Constraints	
1.	The shift coverage requirements must be fulfilled.
2.	After night shift must be at least for 24 hours rest time.
3.	Duty shift must be assigned only on weekends.
4.	Cardiologist cannot be assigned to different assignments at the same time.
5.	Only duty shifts can be assigned on weekends.

Hard constraint N1 states that the total number of shifts on certain days must satisfy the coverage requirements. Hard constraint N2 states that there must be at least 24 hours time difference between night shift and any other shift. Hard constraint N3 states that duty shifts must be assigned only on weekends or on bank holidays. Hard constraint N4 states that if the cardiologist has more than one skill, his or her assignments must not overlap. Hard constraint N5 states that no morning, day and night shifts are allowed to be assigned on weekends. If any of these hard constraints is not satisfied then created roster is considered as improper.

Soft constraints (see Table 2) must not necessarily be satisfied; however, violations of soft constraint are penalized. Sum of penalties defines quality of roster: if the lesser sum is obtained it means the roster of higher quality is prepared. Objective of solving such problems is to minimize objective function [3]:

Table 2. Soft Constraints	
1.	Maximum number of shift assignments.
2.	Maximum number of consecutive work days.
3.	Minimum number of consecutive work days.
4.	Maximum number of consecutive non-working days.
5.	Minimum number of consecutive non-working days.
6.	Maximum number of a certain shift worked.
7.	Maximum number of consecutive working weekends.
8.	Maximum number of working weekend in a month.
9.	Requested days off.
10.	Requested days on.
11.	Requested shifts on.
12.	Requested shifts off.
13.	Requested shifts for each weekday.

[a]Corresponding author, email: *m.liogys@eif.viko.lt*, cell: +370(698)87485

Table 3. Shift types

Shift label	Shift type	Time period
R1	Morning	07:30 - 15:12
R2	Morning	07:30 - 09:18
R3	Morning	09:18 - 15:12
R4	Morning	07:30 - 11:06
R5	Morning	09:18 - 14:24
R6	Morning	09:18 - 14:36
R7	Morning	11:06 - 15:12
D1	Day	15:12 - 17:00
D2	Day	15:12 - 18:48
D3	Day	17:00 - 20:36
D4	Day	15:12 - 20:36
N1	Night	18:48 - 24:00
N2	Night	00:00 - 09:12
Dt1	Duty	08:00 - 24:00
Dt2	Duty	00:00 - 08:00

$$y = \sum_{i=1}^{n} \sum_{j \in F(i)}^{m} p_{ij} x_{ij} \quad (1)$$

where n represents number of cardiologists; m - number of shift sequences; p_{ij} - cost of cardiologist i working shift sequence j; $F(i)$ - set of feasible shift sequences for cardiologist i; x_{ij} - decision variable which is equal to 1 if cardiologist i works shift sequence j, 0 - otherwise.

There are 17 different shifts available, according to various lengths of working hours for cardiologists - see Table 3. Rostering period is one calendar month.

Part of cardiologists has full time workload; part of cardiologists has part time workload; part of cardiologists has more than full time workload. There are cardiologists who have more than one skill and in order to construct correct roster have to be considered that his / her assignments do not overlap. Best case scenario is then one assignment ends and starts another for those who have several skills, i.e. no time interval between assignments on the same day.

2. Overview of methods

A solution of rostering problem consists of a collection of personal schedules for each of the cardiologists. A schedule for a cardiologist consists of shifts that usually are different in lengths and types (morning shifts, day shifts, etc.).

Table 4. Hard constraints categorized to schedule and roster constraints

N	Hard constraint	Category
1.	The shift coverage requirements must be fulfilled.	Roster
2.	After night shift must be at least 24 hours of rest time.	Schedule
3.	Duty shift must be assigned on weekends.	Schedule
4.	Cardiologist cannot be assigned to different assignments on the same time.	Schedule
5.	Only duty shifts can be assigned on weekends.	Schedule

Table 5. Soft constraints categorized to sequence (SE), schedule (SHE) and roster constraints

N	Soft constraint	Category
1.	Maximum number of shift assignments.	SHE
2.	Maximum number of consecutive work days.	SE / SHE
3.	Minimum number of consecutive work days.	SE / SHE
4.	Maximum number of consecutive non-working days.	SHE
5.	Minimum number of consecutive non-working days.	SHE
6.	Maximum number of a certain shift worked.	SHE
7.	Maximum number of consecutive working weekends.	SHE
8.	Maximum number of working weekend in a month.	SHE
9.	Requested days off.	SHE
10.	Requested days on.	SHE
11.	Requested shifts on.	SE
12.	Requested shifts off.	SE
13.	Requested shifts for each weekday.	SHE

Chosen methods use different approach on building the schedules of a roster. Shift sequence based method builds the schedules using shift sequences, simulated annealing - using individual shifts.

2.1. Shift Sequence Based Method

This method consists of two stages: generation of shift sequences and schedule construction according to generated shifts, that are discussed in sections 3.1.1 and 3.1.2. Hard and soft constraints are additionally categorized to sequence, schedule and roster [4]:

i) sequence constraints are applied when constructing shift sequences for each cardiologist;
ii) schedule constraints are applied when combining schedule for each cardiologist;
iii) roster constraints are applied when constructing an overall solution - roster.

Categorized constraints are listed in the Table 4 and Table 5. Last column describes which category of constraints listed above it applies to.

2.1.1. Shift Sequences Construction

In this stage, the shift sequences are constructed for each cardiologist, considering sequence constraints. Shifts sequences are ranked by their penalties for easier retrieval in later stage.

To decrease the complexity, it is possible to limit the number of possible valid shift sequences by either considering only sequences with a penalty below a certain threshold, or by selecting the certain amount of the best sequences for each cardiologist in the second stage of the approach. Shift sequence length is up to 5 shifts. If there is a need for constructing sequences of length greater than 5, such sequences are constructed using combination of sequences of length up to 5 shifts. This combination is performed in the schedule and roster construction stage.

2.1.2. The Construction of Schedules

In the second stage of the approach, schedules for each cardiologist are constructed iteratively, using the shift sequences produced in shift sequences construction stage. Only schedule constraints are under consideration then constructing schedule for cardiologists. Roster constraints are applied then schedule is added to roster.

Basic algorithm of *Shift Sequence* (Algorithm 1, see Table 6) is written using method described in Ref. [4]. It is an adaptive iterative method where cardiologists who received the highest schedule penalties in the last iteration are scheduled first at the current iteration.

Schedule construction process is presented in Algorithm 2, see Table 7. It builds a schedule for the cardiologist based on the partial roster built so far for other cardiologists and returns its penalty to Algorithm 1. The basic idea of this algorithm is to generate a schedule with a low penalty value for the nurse, using low penalty shift sequences. Variable `curr_threshold` points what kind of sequences to use, i.e. if its value is 0, then are used only those sequences that has penalty equal to 0. If no valid assignment can be made for the current cardiologist, the shift sequence with the second lowest penalty is considered and so on. The sequences are assigned for the current cardiologist if the penalty of assigning them is under the current threshold (`curr_threshold`).

During the roster construction, and after a schedule has been generated for the current cardiologist, an improvement method based on an efficient greedy local search is carried out on the partial roster. It simply swaps any pair of shifts between two cardiologists in the partial roster, as long as the swaps satisfy hard constraints and decrease the roster penalty.

After all the schedules have been constructed and a roster has been built, there may still be some shifts for which the coverage is not satisfied. To repair this, a greedy heuristic is used. Each extra shift to be assigned is added to the nurse's schedule whose penalty decreases the most (or increases the least if all worsen) on receiving this shift. After this repair step, the local search is applied once more to improve the quality of the overall roster.

2.2. Simulated Annealing Method

The simulated annealing method is used to solve combinatorial optimization problems. A combinatorial optimization problem is a minimization (maximization) problem consisting of three parts: a set of instances; a finite set of candidate solutions for each instance; and a cost function that assigns to each candidate solution for each instance a positive number called cost. The optimal solution to an instance of a minimization (maximization) problem is the candidate solution having the minimum (maximum) cost.

In the simulated annealing method (Algorithm 3, see Table 8), the cost function to be minimized is identified with the energy of a physical system, and the solution space is identified with the state space. The solution space of the optimization problem is explored by a probabilistic hill climbing search, whose step size is controlled by a parameter T that plays the role of the temperature in a physical system.

By slowly lowering the temperature towards zero according to a properly chosen schedule, one can show that the globally optimal solutions are approached asymptotically.

Table 6. Algorithm 1. `Construct_Roster()`

```
construct and rank the shifts sequences for each cardiologist
iteration = 0
set max no. of iterations (MaxNoIter)
randomly order cardiologists
while (iteration < MaxNoIter)
   for "each cardiologist"  in "ordered list of cardiologists"
        Construct_Schedule(cardiologist, partial_roster)
        greedy local search to improve partial roster
        store the best roster constructed so far
        calculate the penalty for the schedule of "each cardiologist"
        sort the cardiologists by their schedule's penalty in a non-increasing order
increase iteration counter
```

Table 7. Algorithm 2. `Construct_Schedule(cardiologist, partial_roster)`

```
set final threshold (f_threshold)
set current threshold (curr_threshold = 0)
while (curr_threshold <= f_threshold)
    for each sequence in  ranked list for the cardiologist do
        for each day from the first day in the planning period
            assign the sequence's  corresponding shifts based on the partial_roster
    if it does not violate any hard constraints and the penalty <= curr\_threshold
    increase the value of f\_threshold
return schedule
```

Table 8. Algorithm 3. Simulated annealing

```
Build initial roster Current_Roster
Initialize starting temperature T
LOOP
    New_Roster = Neighbour of Current_Roster
    Calculation of Current_Cost
    Calculation of New_Cost
    If (Current_Cost - New_Cost <= 0)
    Then
    Current_Roster = New_Roster
    Else
        If (f > Random(0, 1))
        Then
            Current_Roster = New_Roster
        Else
            Do Nothing
    Decrease Temperature
END LOOP When Stop Criterion Is Met
```

Functional parameter f depending on temperature T was calculated using following equation:

$$f = \exp\left[\frac{Current_{Cost} - New_{Cost}}{T}\right] \quad (2)$$

Simulated annealing chooses a random move from the neighbourhood - if the move is better than its current position then simulated annealing will always take it. If the move is worse then it will be accepted based on some probability.

Basic algorithm of *Simulated Annealing* is written using method presented in Ref. [10]. Neighborhood rosters where created using the following strategies [11].

Single shift-day. The simplest neighborhood of a schedule includes all the feasible solutions that differ in the position of one scheduled shift.

Overtime - Undertime neighborhood. This neighborhood only considers moving shifts from people with overtime to people with undertime.

Personal requests neighborhood. This neighborhood includes personnel personal requirements like shift on or off, day on or off and etc.

Shuffle neighborhood. Instead of moving duties (as in the simple single shift-day neighborhood), all the duties, which are scheduled in a period from one day to a number of days equal to half the planning period, are switched between the person with the worst schedule and any other person.

3. Simulations.

Experiments there held under same conditions: hardware - Double Core CPU 2.16 GHz, amount of iterations - 500, tests were ran separately 100 times. As we see from Table 9, *Simulated Annealing* method creates better quality rosters in shorter time than *Shift Sequence Based* method.

4. Conclusion

This article presents comparison of two methods used in solving real-world medical staff rostering problem. Test results shows that simulated annealing method is more efficient in matter of roster creation time and roster quality.

Table 9. Test results.
E - Average Execution time (s);
R - Average Roster Quality (rel. un.)

Method	E	R
Shift Sequence Based	250	9655
Simulated Annealing	23	8890

References

1. U. AICKELIN; K. DOWSLAND. An Indirect Genetic Algorithm for a Nurse Scheduling Problem – *Computers and Operations Research* 2004, 31(5), p. 761-778.
2. U. AICKELIN, L. JINGPENG. An Estimation of Distribution Algorithm for Nurse Scheduling – *Annals of Operations Research* 2007, 155(1), p. 289 - 309.
3. U. AICKELIN, Uwe; WHITE, Paul. Building Better Nurse Scheduling Algorithms – *Annals of Operations Research* 2004, 128(1-4), p. 159 - 177.
4. P. BRUCKER, E. BURKE, T. CURTOIS, R. QU, G. VANDEN BERGHE. A shift sequence based approach for nurse scheduling and a new benchmark dataset – *Journal of Heuristics* 2010, 16(4), p. 559 - 573.
5. M. BRUSCO, L. JACOBS. Cost analysis of alternative formulations for personnel scheduling in continuously operating organizations – *European Journal of Operational Research* 1995, 86(2), p. 249 - 261.
6. E. BURKE, T. CURTOIS, G. POST, R. QU, B. VELTMAN. A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem – *European Journal of Operational Research* 2008, 188(2), p. 330 - 341.
7. A. IKEGAMI, A. NIWA. A subproblem-centric model and approach to the nurse scheduling problem – *Mathematical programming* 2003, 97(3), p. 517 - 541.
8. L. JINGPENG, U. AICKELIN. Bayesian Optimisation Algorithm for Nurse Scheduling, Scalable Optimization via Probabilistic Modeling – *Springer* 2006, p. 315 - 332. – ISBN 978-3-540-34953-2.
9. M.N. AZAIEZ, S.S. AL SHARIF. A 0-1 goal programming model for nurse scheduling – *Computers and Operations Research* 32, p. 491-507.
10. S. KUNDU, M. MAHATO, B. MAHANTY, S. ARCHARYYA. Comparative Performance of Simulated Annealing and Genetic Algorithm in Solving Nurse Scheduling Problem – *Proceedings of the International MultiConference of Engineers and Computer Scientists* 2008 Vol I, 2008, 96-100 p. – ISBN 978-988-98671-8-8.
11. E. BURKE, P. DE CAUSMAECKER, S. PETROVIC, G.VANDEN BERGHE. Variable Neighbourhood Search for Nurse Rostering Problems – *Book Metaheuristics* 2004, 153-172 p. – ISBN 1-4020-7653-3.