ISSN 2029-7564 (online)
SOCIALINĖS TECHNOLOGIJOS
SOCIAL TECHNOLOGIES
2011, 1(1), p. 151–162.

ADAPTATION OF SHIFT SEQUENCE BASED METHOD FOR HIGH NUMBER IN SHIFTS ROSTERING PROBLEM FOR HEALTH CARE WORKERS

Mindaugas Liogys

Vilnius University Institute of Mathematics and Informatics, Lithuania, mliogys@gmail.com

Abstract

Purpose—is to investigate a shift sequence-based approach efficiency then problem consisting of a high number of shifts.

Research objectives:

- Solve health care workers rostering problem using a shift sequence based method.
- Measure its efficiency then number of shifts increases.

Design/methodology/approach—Usually rostering problems are highly constrained. Constraints are classified to soft and hard constraints. Soft and hard constraints of the problem are additionally classified to: sequence constraints, schedule constraints and roster constraints. Sequence constraints are considered when constructing shift sequences. Schedule constraints are considered when constructing a schedule. Roster constraints are applied, then constructing overall solution, i.e. combining all schedules.

Shift sequence based approach consists of two stages:

- Shift sequences construction,
- The construction of schedules.

ISSN 2029-7564 (online)
https://www.mruni.eu/lt/mokslo_darbai/st/apie_leidini/
https://www.mruni.eu/en/mokslo_darbai/st/apie_leidini/index.php

In the shift sequences construction stage, the shift sequences are constructed for each set of health care workers of different skill, considering sequence constraints. Shifts sequences are ranked by their penalties for easier retrieval in later stage.

In schedules construction stage, schedules for each health care worker are constructed iteratively, using the shift sequences produced in stage 1.

Shift sequence based method is an adaptive iterative method where health care workers who received the highest schedule penalties in the last iteration are scheduled first at the current iteration.

During the roster construction, and after a schedule has been generated for the current health care worker, 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 health care workers in the (partial) roster, as long as the swaps satisfy hard constraints and decrease the roster penalty.

Findings—Using shift sequence method for solving health care workers rostering problem is inefficient, because of large amount of shifts sequences (feasible shifts sequences are approximately 260 thousands).

In order to speed up roster construction process shifts are grouped to four groups: morning shifts, day shifts, night shifts and duty shifts. There are only 64 feasible shifts sequences, in this case.

After roster construction shift groups are replaced with the one of shift belonging to that group of shifts.

When all shifts are added to roster, computation of workload for each schedule is performed. If computed workload is equal to the one defined in working contract, then this schedule is complete, else begin shifts revision process. During revision process those schedules are considered which do not meet work contract requirements.

If computed workload is larger than the one defined in working contract, each shift is replaced with the shift, if it's possible, with lesser duration time. If computed workload is lesser than the one defined in working contract, each shift is replaced with the shift, if it's possible, with larger duration time.

This process continues while schedule does not meet workload requirement defined in working contract or no further improvement can be made.

Research limitations/implications—Problem dimension: 27 health care workers, 15 shifts, over 20 soft constraints, rostering period—one calendar month.

Practical implications – modifications made to shift sequence based approach allows to construct a roster for one of the major Lithuania's hospitals personnel in shorter time.

Originality/Value—modification of shift sequence based approach is proposed.

Keywords**:** shift, sequence, schedule, health care workers rostering.

Research type**:** research paper.

Introduction

The common objective of health care workers (further in this text, abbreviation HCW is used for term health care worker) rostering problem is to produce rosters with a balanced workload as well as to satisfy individual preferences as much as possible. Constraints are usually categorized into two groups: hard and soft constraints. Hard constraints must be satisfied to obtain feasible solutions. Soft constraints are desirable but not mandatory, and can be violated. Authors (Ikegami, Niwa 2003) additionally categorized constraints to: shift constraints, and nurse constraints. Authors (Brucker, Burke, Curtois, Qu, Berghe 2010) categorized to: shift constraints, schedule constraints and roster constraints. Their proposed approach uses shift sequences - not individual shifts as many other rostering models do: (Brusco, Jacobs 1995), (Burke, Curtois, Post, Qu, Veltman 2008), (Aickelin, Dowsland 2004).

This article tackles HCWs rostering problem with high number in shifts using shift sequence based approach and introduces with improvements which made rostering problem solvable in shorter time.

1. Problem Formulation

The problem is that of creating monthly schedules for HCWs at a major Lithuania hospital. These schedules have to satisfy working contracts and meet as far as possible HCWs' requests.

A solution of rostering problem consists of a collection of personal schedules for each of the HCWs. A schedule for a HCW consists of shift sequences that usually have different lengths and different types (morning shifts, day shifts, etc.).

The shifts in a sequence must be performed on consecutive days, one shift per day. Between the sequences in a schedule there are days without shifts.

Constraints are categorized as sequence, schedule and roster constraints (Brucker, Burke, Curtois, Qu, Berghe 2010):

- Sequence constraints are applied when constructing shift sequences for each HCW with certain skills.
- Schedule constraints are applied when combining schedule for each HCW.
- Roster constraints are applied when constructing an overall solution – roster.

Hard and soft constraints of the problem are listed in the following tables. Last column describes which category of constraints listed above it applies.

Table 1. Hard constraints categorized to schedule and roster constraints

	Hard Constraint	Category
1.	The shift coverage requirements must be fulfilled	Roster
2.	After night shift must be at least for 24 hours rest time	Schedule
3.	Duty shift must be assigned on weekends	Schedule
4.	HCW cannot be assigned to different assignments on the same time	Schedule

Hard constraint No. 1 states that the total number of shifts on certain days must satisfy the coverage requirements. Hard constraint No. 2 states that there must be at least 24 hours time difference between night shift and any other shift. Hard constraint No. 3 states that duty shifts must be assigned only on weekends or on bank holidays. Hard constraint No. 4 states that if the HCW has more than one skill, his or her assignments must not overlap. If any of these hard constraints is not satisfied then created roster is considered as improper.

Table 2. Soft constraints categorized to sequence, schedule and roster constraints

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

Soft constraints not necessarily must be satisfied, however violations of soft constraint are penalized. Sum of penalties defines quality of roster – the lesser sum is the higher quality roster is. Objective of solving such problems is to minimize objective function (Aickelin, White 2004):

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

Where

n – Number of HCWs.

m – Number of shift sequences.

p_{ij} – Cost of HCW i working shift sequence j.

$F(i)$ – Set of feasible shift sequences for HCW i.

x_{ij} – decision variable, it is equal 1 if HCW i works shift sequence j, 0 – otherwise.

Problem dimension: 27 HCWs, 15 shifts (Table 3), rostering period – one calendar month.

Part of HCWs has full time work; part of HCWs has part time work; part of HCWs has more than full time work. There are HCWs who have more than one skill and in

order to construct correct roster have to be considered that his / her assignments does 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 same day.

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}

2. A Two-Stage Adaptive Approach Overview

Health care workers rostering problems are usually dealt with by constructing the schedules by generating assignments of HCWs to shifts for each day of the scheduling period.

This approach constructs schedules using shift sequences and it consists of two stages:

- A shifts sequences construction with respect to sequence constraints listed in Table 1.
- A construction of schedules for the HCWs which are combined into a roster with respect to the schedule and roster constraints listed in Table 1 and Table 2.

2.1. Shift Sequences Construction

In this stage, the shift sequences are constructed for each HCW, 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 HCW in the second stage of the approach. Shift sequence length is up to 5 shifts, because of 5 working days in a week, usually. If there is the need of 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.

2.2. The Construction of Schedules

In the second stage of the approach, schedules for each HCW are constructed iteratively, using the shift sequences produced in stage 1, described above. Only schedule constraints are under consideration then constructing schedule for HCW.

Basically, the construction of roster is made using two algorithms.

Algorithm 1. Construct_Roster()

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

Algorithm 2. Construct_Schedule(HCW, partial_roster)

set final threshold (f_threshold)
set current threshold (curr_threshold = 0)
while (curr_threshold ≤ f_threshold)
for each sequence ε ranked list for the HCW 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 cons
traints and the penalty ≤ curr_threshold
increase the value of f_threshold
return schedule

Algorithm 1 first operation is stage 1, the rest of the pseudo code is stage 2 of this approach. It is an adaptive iterative method where HCWs who received the highest schedule penalties in the last iteration are scheduled first at the current iteration.

Algorithm 2 presents the schedule construction process. It builds a schedule for the HCW based on the partial roster built so far for other HCWs 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 HCW, 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 HCW, the shift sequence with the second lowest penalty is considered and so on. The sequences are assigned for the current HCW 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 HCW, an improvement method based on an efficient greedy local search (Peyro, Ruiz 2010) is carried out on the partial roster. It simply swaps any pair of shifts between two HCWs 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 HCWs' 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.

3. Adaptation of Shift Sequence Approach to Problem with High Number of Shifts

Using shift sequence method for solving HCWs' rostering problem is inefficient, because of large amount of shifts sequences (feasible shifts sequences are over 260 thousands).

Regardless of possibility to limit number of shifts sequences used in stage 2, it still does not help to solve the problem. If we choose only those with penalty 0, we still get large amount of sequences, if we choose only *n* best sequences we may get to situation there is no suitable shift sequence to assign to HCWs.

In order to limit amount of sequences and yet do not lose any of them, the shifts listed in Table 3 are grouped into four groups (Table 4): morning shifts (M), day shifts (D), night shifts (N), duty shifts (Dt).

Table 4. Grouped shifts

Shifts group label	Corresponding Shift	Time Period
M	R1	07^{30}–15^{12}
	R2	07^{30}–09^{18}
	R3	09^{18}–15^{12}
	R4	07^{30}–11^{06}
	R5	09^{18}–14^{24}
	R6	09^{18}–14^{36}
	R7	11^{06}–15^{12}
D	D1	15^{12}–17^{00}
	D2	15^{12}–18^{48}
	D3	17^{00}–20^{36}
	D4	15^{12}–20^{36}
N	N1	18^{48}–24^{00}
	N2	00^{00}–09^{12}
Dt	Dt1	08^{00}–24^{00}
	Dt2	00^{00}–08^{00}

In this case we get only 64 feasible shifts sequences.

Hard constraint No. 2 (Table 1) must be redefined as "After night shift must be day-off."

Then roster is constructed using method described above, the task is to add the real shifts listed in Table 3 to corresponding places of constructed roster. Shift M replaced with the one of the shifts belonging to the morning shifts group. Shift D replaced with the one of the shifts belonging to the day shifts group. Shift N is replaced with both of the shifts belonging to the night shifts group. Shift Dt replaced with both of the shifts belonging to the duty shifts group. Then replacing night shifts or duty shifts, the shifts N1 or Dt1 is replaced with N or Dt, the shifts N2 or Dt2 is added to the next day.

When all shifts are added to roster, computation of workload for each schedule is performed. If computed workload is equal to the one defined in working contract, then this schedule is complete, else begin shifts revision process. During revision process those schedules are considered which do not meet work contract requirements.

If computed workload is larger than the one defined in working contract, each shift is replaced with the shift, if it's possible, with shorter duration time. If computed workload is lesser than the one defined in working contract, each shift is replaced with the shift, if it's possible, with longer duration time.

This process continues while schedule does not meet workload requirement defined in working contract or no further improvement can be made.

Shifts replacement and revision procedures can be formulated as the following:

Algorithm 3. Shifts_Replacement()

for each schedule ɛɛ *roster do*
for each days-on in schedule
shifts group label replace with corresponding shift belonging to that group

Algorithm 4. Balance_Workload()

for each HCW ɛ *HCWs list do*
compute schedule workload (computed_workload)
if computed_workload != desired_workload
Revise_Schedule(schedule, computed_workload, desired_workload)

Algorithm 5. Revise_Schedule(schedule, computed_workload, desired_workload)

```
if computed_workload < desired_workload
        for each shift, except night and duty shifts ɛ schedule do
                if shift is not longest in duration
                        replace shift to shift with longer duration time from
                        the same shifts group
                else
                        skip to next shift
        compute schedule workload (computed_workload)
        if computed_workload = desired_workload
                revision complete
else
        for each shift, except night and duty shifts ɛ schedule do
                if shift is not shortest in duration
                        replace shift to shift with shorter duration time from
                        the same shifts group
                else
                        skip to next shift
        compute schedule workload (computed_workload)
        if computed_workload = desired_workload
                revision complete
```

After shifts replacement and revision procedures if there are still left schedules which duration time does not meet HCWs' workload defined in his/her working contract greedy local search is performed again to substitute shifts between two schedules with larger and lesser workloads if it does not violate hard constraints and not increase the penalty of roster.

4. Results and Findings

The experiments were undertaken on an Intel Core 2 Duo 2.16 GHz machine. Testing software is built using C# programming language.

The results of experiments showed that shift sequence based method is very slow in solving rostering problem with high number of shifts—execution time of roster construction process (1 iteration) is approximately 15 minutes. Proposed approach solved problem in much shorter time - execution time of roster construction process (1 iteration) is approximately 20 seconds.

Number of shifts has no direct impact to the roster construction time using proposed approach as long as it can be grouped into groups listed in section 4.

Conclusions

The testing is done using shift sequence based (Brucker, Burke, Curtois, Qu, Berghe 2010) and modified shift sequence based approaches. The results of experiments showed that shift sequence based method is very slow in solving rostering problem with high number of shifts - execution time of roster construction process (1 iteration) is approximately 15 minutes. Proposed approach solved problem in much shorter time—execution time of roster construction process (1 iteration) is approximately 20 seconds.

Number of shifts has no direct impact to the roster construction time using proposed approach as long as it can be grouped into groups listed in section 4.

According to the results of research on implementing Bayesian Network in scheduling problems (Jingpeng, Aickelin 2006; Aickelin, Jingpeng 2007), implemented Bayesian Network can speed up roster construction process to several times. Implementation of Bayesian Network to shift sequence based approach could be promising direction of research for the future work.

Literature

Aickelin, U., Dowsland, K. 2004. An *Indirect Genetic Algorithm for a Nurse Scheduling Problem*, *Computers and Operations Research*, 31(5), p. 761-778.

Aickelin, U., Jingpeng, L. 2007. *An Estimation of Distribution Algorithm for Nurse Scheduling, Annals of Operations Research*, 155(1), p. 289 – 309.

Aickelin, U., White, P. 2004. *Building Better Nurse Scheduling Algorithms*, *Annals of Operations Research*, 128(1-4), p. 159 - 177.

Brucker, P., Burke, E., Curtois, T., Qu, R., Vanden Berghe, G. 2010. *A shift sequence based approach for nurse scheduling and a new benchmark dataset*, *Journal of Heuristics*, 16(4), p. 559 – 573.

Brusco, M., Jacobs, L. 1995. *Cost analysis of alternative formulations for personnel scheduling in continuously operating organisations*, *European Journal of Operational Research*, 86(2), p. 249 – 261.

Burke, E., Curtois, T., Post, G., Qu, R., Veltman, B. 2008. *A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem, European Journal of Operational Research*, 188(2), p. 330 – 341.

Ikegami, A., Niwa, A. 2003. *A subproblem-centric model and approach to the nurse scheduling problem*. *Mathematical programming*, 97(3), p. 517 – 541.

Jingpeng, L., Aickelin, U. 2006. *Bayesian Optimisation Algorithm for Nurse Scheduling*, *Scalable Optimization via Probabilistic Modeling*, Springer, p. 315 – 332. ISBN 978-3-540-34953-2.

Peyro, L. F., Ruiz, R. 2010. *Iterated greedy local search methods for unrelated parallel machine scheduling, European Journal of Operational Research,* 207, p. 55 – 69.

SVEIKATOS PRIEŽIŪROS ĮSTAIGOS PERSONALO TVARKARAŠČIŲ SUDARYMAS MODIFIKUOTU PAMAINŲ SEKŲ METODU, KAI DIDELIS PAMAINŲ SKAIČIUS

Mindaugas Liogys

Vilniaus universitetas, Matematikos ir informatikos institutas, Lietuva, mliogys@gmail.com

Santrauka. *Šiame straipsnyje nagrinėjamas pamainų sekų metodo efektyvumas sudarant didelių ligoninių gydytojų darbo tvarkaraščius, kai yra daug (iki 15) pamainų.*

Nagrinėjamas pavyzdys, kai tvarkaraštis sudaromas vienam kalendoriniam mėnesiui 27-iems gydytojams. Dėl skirtingo gydytojų darbo krūvio (dalis gydytojų dirba visu etatu, dalis – puse etato, trečdalis – ketvirčiu etato) susidaro 15 skirtingų pamainų. Sudarant tvarkaraštį susiduriama su daugiau nei 20 ribojimų.

Ankstesni tyrimai parodė, kad šis metodas gana efektyvus sudarant darbo tvarkaraštį, kai pamainų skaičius nedidelis (iki 4 pamainų). Sprendžiant gydytojų tvarkaraščių sudarymo uždavinį pastebėta, kad didėjant pamainų skaičiui, pailgėja tvarkaraščio sudarymo laikas. Kai yra keturios pamainų, šiuo metodu tvarkaraštis gali būti parengiamas per 20 sekundžių (pasirinkus 1 iteraciją). Kai pamainų skaičius išauga iki 15, tvarkaraščio sudarymo trukmė yra 15 minučių (pasirinkus 1 iteraciją). Straipsnyje pasiūlyta modifikacija leidžia sudaryti tvarkaraštį per apytikriai 45 kartus trumpesnį laiką.

Pasiūlytoje modifikacijoje pamainos skirstomos į keturias grupes: rytinės pamainos, dieninės pamainos, naktinės pamainos ir budėjimo pamainos. Pamainų sekos sudaromos

naudojant ne pačias pamainas, o jų grupes. Taip sumažinamas tinkamų pamainų sekų skaičius (nuo 260 000 pamainų sekų iki 64 pamainų sekų).

Remiantis gautomis pamainų sekomis sudaromas kiekvieno gydytojo darbo tvarkaraštis vietoje pamainų sekų grupės įrašant konkrečią pamainą. Jei apskaičiuojamas darbo krūvis neatitinka darbo sutartyje apibrėžto darbo krūvio, pamaina, atsižvelgiant į tai, ar gautas darbo krūvis yra mažesnis, ar didesnis už apibrėžtą darbo sutartyje, keičiama į kitą ilgesnę arba trumpesnę tai pačiai grupei priklausančią pamainą. Pamainų keitimo procedūra kartojama, kol darbo krūvis atitinka darbo sutartyje apibrėžtą krūvį arba pasiekiama paskutinė mėnesio diena.

Atlikus pamainų keitimo procedūrą atliekama dviejų skirtingų gydytojų pamainų apkeitimo procedūra, jei apkeitimas nepažeidžia būtinųjų ribojimų ir nepablogina tvarkaraščio, t. y. parengto tvarkaraščio baudos taškų suma nepadidėja (už kiekvieną tvarkaraščio neatitiktį uždavinio ribojimams skiriamas tam tikras baudos taškų skaičius. Baudos taškų suma rodo tvarkaraščio kokybę – kuo suma mažesnė, tuo tvarkaraštis kokybiškesnis). Pamainų apkeitimo procedūrai naudojamas godusis lokalios paieškos algoritmas.

Atlikus skirtingų gydytojų pamainų apkeitimo procedūrą vykdoma kita iteracija gydytojus išrikiuojant surinktų baudos taškų mažėjimo tvarka, t. y. pradedama nuo to gydytojo, kurio darbo grafikas ankstesnėje iteracijoje surinko daugiausiai baudos taškų.

Tvarkaraščio rengimas baigiamas, kai įvykdomas iš anksto pasirinktų iteracijų skaičius. Kuo didesnis iteracijų skaičius, tuo kokybiškesnis (mažesnis baudos taškų skaičus) tvarkaraštis, tačiau kuo didesnis iteracijų skaičius, tuo ilgesnis tvarkaraščio rengimo laikas.

Raktažodžiai: *pamaina, seka, tvarkaraštis.*