An Efficient Method for Nurse Scheduling Problem using Simulated Annealing

Young-Woong Ko, DongHoi Kim, Minyeong Jeong, Wooram Jeon, Saangyong Uhmn, Jin Kim*

Dept. of Computer Engineering
Hallym University
Chuncheon, Gangwondo 200-702 Republic of Korea
{yuko,kdh,jmy,jwr,suhmn,jinkim}@hallym.ac.kr

Abstract. We applied simulated annealing algorithm to nurse scheduling problem. For time complexity problem of simulated annealing, we suggested an efficient transition rule using cost matrix for simulated annealing. The experimental results showed that the suggested method generated a nurse scheduling faster in time and better in quality compared to traditional simulated annealing.

Keywords: nurse scheduling problem, simulated annealing, cost matrix

1 Introduction

In nurse scheduling problem (NSP), nurses are assigned into a set of shifts and rest days in a timetable called nurse roster in such a way that several constraints are satisfied. The constraints may be set up by staffing requirements, the rules by the administration and the labor contract clauses. It was proven to be NP-hard even if it was with only a subset of real world constraints [1].

In early days, the problem was formulated as selection of a timetable that minimized an objective function that balanced trade-off between staffing coverage and nurses' schedule preferences [2, 3]. The Constraint Logic Program (CLP) framework, genetic algorithm and Bayesian optimization algorithm were applied [4-8].

In this paper, we suggested a cost matrix based simulated annealing (CMSA) to find a schedule that optimizes a set of constraints. In the next section, we introduced NSP briefly and its cost function and in section 3, SA and a cost matrix and a transition rule in CMSA. Section 4 is concerned with experiments and results. Finally, conclusion and further work are discussed in section 5.

2 Problem Description

2.1 Nurse Scheduling Problem

NSP is to create weekly or monthly schedules for *n* nurses by assigning one out of possible shift patterns to each nurse. These schedules have to satisfy working con-

AST 2013, ASTL Vol. 20, pp. 82 - 85, 2013

tracts and meet demands for the number of nurses of different grades on each shift, while being seen to be fair by the staff concerned. Therefore, NSP is essentially a scheduling problem which suits a number of constraints. Constraints are usually categorized into two categories: soft and hard. While hard constraints must be met, soft ones must be satisfied as much as possible. While a schedule which does not satisfy hard constraints cannot be feasible, one satisfying hard constraints but not soft can be considered as a solution. There are various kinds of hard and soft constraints.

Because the main objective of this study is to show a fast and efficient SA approach, we confined the constraints as follows.

(a) Hard constraints
 (i) Number of nurses for each shift per day. The number of nurses for each shift should be within the predefined range.
 (ii) Working patterns. Morning after night shift, evening after night, morning after evening shift and three consecutive night shifts should be avoided.

(b) Soft constraints
 There are constraints for the total number of off(*o*) days, night(*n*), morning(*m*) and evening(*e*) shifts during the certain periods for each nurse.

2.2 Cost Function

We have to define a cost function to evaluate schedules. Let *N* and *D* be the number of nurses and days, and *s* be one of the three shifts or a day-off. Then, NSP is represented as a problem to decide an $N \times D$ matrix whose element x_{ij} means the shift of nurse *i* on day *j* where $x_{ij} = \{m, e, n, o\}$.

(a) To evaluate the violation of hard constraint (i), we define m_j, e_j, n_j be the total number of nurses for morning, evening, and night shift on day *j*. If any of these numbers are not between minimum and maximum number of nurses for each shift ($m_{\min}$, $m_{\max}$), ($e_{\min}$, $e_{\max}$), and ($n_{\min}$, $n_{\max}$) cost c_1 will be incremented by 1.

(b) To evaluate the violation of hard constraint (ii), working patterns will be examined. Any violation of working patterns will increment cost c_2 by 1.

(c) To evaluate the violation of soft constraint, we define M_i, E_i, N_i, O_i be the total number of shifts *m, e, n, o* for nurse *i* during the period *D* and M_{req}, E_{req}, N_{req}, O_{req} be the required number of shifts, *m, e, n, o* for nurses during the period *D*. If any of these numbers M_i, E_i, N_i, O_i does not match the required numbers M_{req}, E_{req}, N_{req}, O_{req} respectively, cost c_3 will be incremented by 1.

Different weights can be assigned for the costs *c1*, *c2*, and *c3*. Then, the final cost function is

$$f = c_1 * w_1 + c_2 * w_2 + c_3 * w_3$$

where w_1, w_2, w_3 are weights for c_1, c_2, and c_3.

Our final goal is to minimize a cost function *f* so as to find an optimal nurse schedule. The simplest method to find an optimal nurse schedule is a brute force approach to evaluate all possible nurse schedules. It guarantees a feasible nurse schedule with the minimum cost. The number of all possible nurse schedules is 4^{DxN}. If *D* and *N* increase, this approach is intractable. To overcome this problem, we use simulated annealing which is an approximation algorithm. SA provides an acceptable good solution in a fixed amount of time, rather than an optimal solution.

3 A Cost Matrix-based Simulated Annealing (CSMA)

Due to the limit of the length, we refer the details of simulated annealing to [9] and will briefly describe our method. The very first nurse schedule s_{first} which is an $N \times D$ matrix is obtained by randomly assigning each nurse to one of the three shifts or day-off on each day. The cost E_{first} is obtained by calculating c_1, c_2, and c_3. This cost becomes current cost $E_{current}$. And new schedule s_{new} is generated by applying the transition rule to the current schedule.

In this study we use a cost matrix v which is also an $N \times D$ matrix for the transition rule. The value of each cell in v is assigned to 0 or 1 depending on the cost of new schedule. Initially the value in each cell in v is 0. When the cost E_{new} is calculated for s_{new}, the value of the corresponding cell in v is set to 1 if it is increased. Hereafter, we apply tradition rule to the cell x_{ij} with certain probability p only if the corresponding cell v_{ij}=1.

4 Experiments and results

We implemented our proposed CMSA and traditional SA (TSA) in C. The goal was to check whether our algorithm could actually generate an acceptable NSP and compare it with TSA. Identical problem instances were solved by both methods. Both programs were given 100 problem instances. The primitive parameters were as follows: N=15, D={1, 2, 3, 4}, w_1=5, w_2=5, w_3=1, $m_{\min}$=4, $m_{\max}$=6, $e_{\min}$=3, e_{max}=5, $n_{\min}$=3, $n_{\max}$=5. The soft constraints were proportional to the periods (M_{req}=2, E_{req}=2, N_{req}=2, O_{req}=1 for 1 week). The probability p is 2 percent.

Table 1. Comparison of CMSA and TSA.

Period	Method	E_{opt}=0	E_{opt}	I_{opt} / k	T_{opt} (sec)
1week	CMSA	79/100	0.42	305612 / 1x10^6	1.1
	TSA	4/100	5.06	546054 / 1x10^6	1.9
2 weeks	CMSA	54/100	1.26	2900782 / 5 x10^6	23.9
	TSA	0/100	8.88	2312737 / 5 x10^6	30.6
3 weeks	CMSA	72/100	0.66	12192435 / 20 x10^6	155
	TSA	0/100	13.14	9588709 / 20 x10^6	121
4 weeks	CMSA	97/100	0.06	51631978 / 100 x10^6	926
	TSA	0/100	17.28	56574940 / 100 x10^6	1607

The methods were compared on the basis of four criteria: the number of problem solved with cost=0(E_{opt}=0), the average cost of the solution, and the average number of iterations to reach the final state s_{opt} and the time T_{opt}. Table 1 showed the results.

Both methods solved all the given problem instances whose hard constraints were not violated at all. CMSA generated schedules with optimal cost in all periods whereas TSA could not generate schedules with optimal cost except 1 week. The average costs of CMSA were smaller and the time T_{opt} of CMSA was faster than those of TSA. In every aspect, the quality of the solution by CMSA was very impressive.

5 Conclusion and Future Work

In this paper, we proposed an improved SA for NSP in which a cost matrix was used for the transition rule. This approach generated a solution for nurse schedule faster in time and better in quality than traditional SA. Although we have presented this work in terms of nurse scheduling, it should be noticed that the main idea could be applied to many other scheduling problems. Future research is aimed at experiments on real hospital data with more constraints and diversity of requirements.

Acknowledgements

This research was supported by Hallym University Research Fund, 2011(HRF-201208-004) and Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (No. 2009-0077545).

References

1. 1. Osogami, T., Imai, H.: Classification of Various Neighborhood Operations for the Nurse Scheduling Problem. In: ISAAC '00: Proceedings of the 11th International Conference on Algorithms and Computation, Springer-Verlag (December 2000) 72-83.
2. Miller, H.E., Pierskalla, W.P., Rath, G.J.: Nurse Scheduling using Mathematical Programming. Operations Research 24(5) (1976) 857-870.
3. Warner, D.M., Prawda, J.: A Mathematical Programming Model for Scheduling Nursing Personnel in a Hospital. Management Science 19(4-Part-1) (December 1972) 411-422.
4. Abdennadher, S., Schlenker, H.: Nurse Scheduling using Constraint Logic Programming. In: Proc. AAAI ’99/IAAI ’99 (1999) 838-843
5. Jan, A., Yamamoto, M., Ohuchi, A.: Evolutionary Algorithms for Nurse Scheduling Problem. In: on Evolutionary Computation, 2000. Proc. The 2000 Congress on (2000) 196-203
6. Aickelin, U., Dowsland, K.A.: An indirect Genetic Algorithm for a Nurse-Scheduling Problem. Computers & Operations Research 31(5) (April 2004) 761-778.
7. Li, J., Aickelin, U.: A Bayesian Optimization Algorithm for the Nurse Scheduling Problem. Evolutionary Computation, 2003. In: CEC '03. The 2003 Congress on 3 (2003).
8. Kundu, S., Mahato, M., Mahanty, B., Acharyya, S.: Comparative Performance of Simulated Annealing and Genetic Algorithm in Solving Nurse Scheduling Problem. In: Proc. Int'l Multi Conference of Engineers and Computer Scientists 2008 1 (January 2008) 1-5.
9. Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P.: Optimization by Simulated Annealing. Science 220(4598) (1983) 671–680.