Available online www.jsaer.com

***Journal of Scientific and Engineering Research*, 2019, 6(4):153-160**

Research Article

ISSN: 2394-2630
CODEN(USA): JSERBR

The Solution of Nurse Scheduling Problem with Simulated Annealing Algorithm

Bilgen AYAN-KOÇ[1], Mehmet AKTAN[2]

[1]Corresponding author, Necmettin Erbakan University, Dept. of Industrial Engineering, Konya, Turkey.
bilgenayan@hotmail.com
[2]Prof. Dr., Necmettin Erbakan University, Dept. of Industrial Engineering, Konya, Turkey
maktan@erbakan.edu.tr

Abstract Among the personnel scheduling problems, nurse scheduling problem (NSP) is more sensitive and important since it is specific and related to human health. In this study, it is intended to create the best work schedule for 15 nurses working in 2 shifts at a hospital operating 24 hours a day by using Simulated Annealing (SA) Algorithm. The annealing algorithm developed within the scope of the research was written from scratch by maintaining but developing the basic logic of the general annealing algorithm. The problem addressed by the algorithm which was expected to provide the best scheduling for the nurses by satisfying all 16 constraints were solved by the developed algorithm. The time to reach the best solution for 0-8 off (7-15 working) nurses in the given problem, on average, was calculated as 34 seconds (min.150-max.1,338 trials) in the case of 7-15 working nurses. The algorithm has also been tested for the case of 9 off (i.e. 6 working) nurses and the best solution in 195 seconds (min.456- max.4,864 iterations) providing all the constraints. In the study, a new temperature reduction technique and a new assignment technique were developed for the annealing algorithm. The new developed technique is called as "multiple simulated annealing with double assignment".

Keywords Annealing Algorithm, Assignment Problem, Nurse Scheduling, Shift Assignment, Staff Scheduling

1. Introduction

It is very difficult to assign work shifts to nurses for weeks. The fulfillment of tasks within an organization requires the creation of a schedule that encompasses each employee. The solution to this problem in the field of health is even more difficult because of the need for a number of different staffs on different days and shifts [1]. Scheduling nurse shifts that constitutes a multi-purpose problem due to adaptation problems, internal policies, personal preferences of nurses, necessity of different skills for patient care, nurse qualifications, patient sensitivity level, staff turnover is very important in terms of reducing the cost of employees and it constitutes 59% of total cost of a hospital as an average [2].

2. Theoretical Framework

2.1. Nurse Scheduling Problem

Nurse Rostering Problem (NRP) or Nurse Scheduling Problem (NSP), one of the types of staff scheduling, arose from the need to develop a decision-making tool for the appointment of nurses in line with nurse preferences and patient workload requirements [3]. The NSP represents a subclass of Constraint Satisfaction Problems (CSP), which includes a number of restraints. The problem is rather limited and difficult to solve. The aim is to ensure that nurses assign high quality shifts by providing restrictions, appropriate to the rules of employment contract, and the needs of nurses and employers in health care facilities. Restrictions in NSP are classified as hard and soft depending on their importance [4]. These restrictions may vary considerably according to individual preferences, as well as the legal arrangements that depend on institutions and countries. Hard constraints must be provided to obtain feasible solutions. On the other hand, it is desirable to have soft constraints, although not mandatory, and they can be violated [1].

Nurse scheduling can, in simple terms, be defined as determining the working order for each nurse. Nurse programs are typically developed for a period of 4 weeks. These programs can be flexible, be changed or be corrected each month [5]. Cyclic schedules are more widely used [6]. Fixed or cyclic schedules often provide good solutions, but they cannot easily meet staff preferences and fluctuating demand [5].

There are many studies in the literature on nurse scheduling and the solution of NSP. Tein and Ramli [7] reviewed recent developments in nurse scheduling. Cheang et al. [8] and Clark et al. [9] evaluated nursing scheduling and re-scheduling studies. Warner [10] discussed the problem of nursing planning and its solutions. Tien and Kamiyama [11], Bradley and Martin [12] conducted extensive research on personnel planning and scheduling. Sitompul and Randhawa [13], Cheang et al. [8] and Burke et al. [14] discussed various approaches to NSP in the literature. Basically, these approaches range from traditional mathematical programming methods [15-17] to proprietary intuitive methods [18-19]. One of the most important research aspects in nurse scheduling since the 2000s has been the study of meta-heuristic methods, especially evolutionary methods [20-21]. Simulation of annealing [18, 22-23], tabu search [24-28], Memetic algorithms [29-30], genetic algorithm [21, 31-32], variable neighborhood search [33] and other meta-heuristic methods including Bayesian optimization [34] have been studied. Many of these advanced intuitive approaches attempt to solve models that address the complexity needed and variety of demands in modern hospital environments [35].

Manual and automatic methods can be used for scheduling, and nurse scheduling can be organized in a “cyclic” or “non-cyclic” schedule [5]. Commonly used solution approaches for nurse scheduling are solution methods based on mathematical programming and heuristic methods. On the other hand, the nurse timing problem is a stochastic problem in which the demand is uncertain [3]. Methods and approaches developed for NSP can be discussed in three groups which are "optimization approaches", "artificial intelligence methods" and "heuristic methods". Among these methods and approaches, the annealing simulation algorithm, which is one of the meta-heuristic methods, is the method used in the solution of NSP discussed in this study.

2.2. Simulated Annealing Algorithm

Simulated Annealing (SA) is a stochastic approach based on the simulation of the statistical process of growing crystals using the annealing process to reach the overall minimum internal energy configuration [36]. More specifically, it is based on the imitation of the physical annealing process, which enables to increase the quality of a solid material by heating it to a certain temperature and then cooling [37].

The function to control the cooling process in the annealing simulation, i.e. to adjust the speed of cooling, refers to the possibility of acceptability of the local solutions that will help to find the global optimum solution, even if they are not better when searching for the optimum solution. At the beginning of searching, this probability needs to be set so that non-optimal solutions are accepted, and in this direction, it is high enough to be able to find the global optimum solution by avoiding local optimum solutions. During the search for the optimum solution, it is seen that the probability is decreasing and thus it is difficult to escape from the local optimum solution. Here, the aim is to try to reach the global optimum solution rather than finding a new optimum solution by searching the neighbors of the current local optimum solution [37].

The simulated annealing algorithm starts with a random solution and produces the next solution with the local neighborhood in each iteration. The new solution that improves the E function of the system (which reduces the energy of the matter/system) that optimizes the energy level of the system is always accepted. On the other hand, temporary, non-optimal solutions that allow for increasing the temperature of the system or allow for a certain level of degradation in the system are also accepted. The basic operation of the algorithm can be shown as in Figure 2.1 [38].

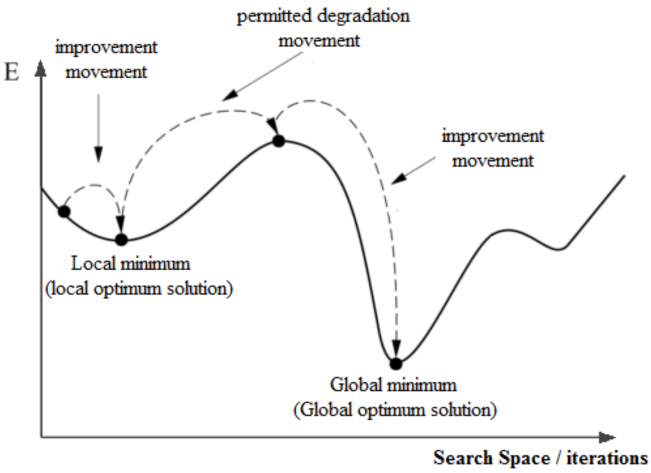

Figure 2.1: Basic Operation of Simulated Annealing Algorithm [38]

The energy difference (ΔE) between the current and the new optimum solution shows the difference between the current value of the objective function and the new solution created through the neighboring solution of the objective function, that is, the difference between the first and the next optimum solutions. The likelihood of a less optimal solution is based on the Boltzmann distribution, as shown in Equation 2.1:

$$P(\Delta E, T) = e^{\frac{\Delta E}{T}} = e^{\frac{E(n+1)-E(n)}{T}} \qquad \text{(Equation 2.1)}$$

Here, $E(n)$ represents the objective function (current energy level) of the current solution, $E(n+1)$ represents the energy level of the new solution obtained by the local neighborhood, and T represents the actual temperature of the system. When the T value (i.e. temperature) increases, the P value (i.e. the probability of accepting a less optimal neighboring solution) is reduced. At this point, the number of new solutions that will be investigated (permitted) at a certain temperature level is determined by the researcher. However, when a stable state is reached (where there is no further improvement in solutions), the temperature is generally reduced. In this method, both current and all of the better solutions should be recorded [38]. In case that $T_k = T_0/logk$ logarithmic cooling where T_k is the actual temperature (instead of T), T_0 is the first temperature and k is the control parameter, it has been proven that the annealing simulation algorithms are close to the global optimum solution [39].

The main benefit of the Simulated Annealing procedure is the ability to escape from the local optimum solution, thanks to its ability to accept less optimal solutions. The likelihood of moving to less optimal solutions is increased when the temperature is high, but decreases with the degradation of the objective function (E). The probability of accepting the worse alternative is $1/[1+P(\Delta E,T)]>Z$ where $P(\Delta E,T)$ is $e^{\Delta E/kT}$, ΔE is the difference in the objective function, T is the current temperature, k is the Boltzmann constant and Z is a randomly generated number between 0 and 1. In the first stage, when the temperature of the system is high, the probability of accepting degradation or worse solutions is higher, and with the decrease in temperature, the algorithm becomes stable and allows only smaller decreases in the quality of the solution. The choice of cooling rate governs the behavior of the annealing and proceeds in convergence by determining the movements of the temperature during iterations. The performance of the annealing simulation algorithm depends largely on the variables included in the cooling selection, such as the initial temperature, the ending criteria, and the cooling function. For example, if the initial temperature T_0 is too high, the algorithm is more prone to random local search, and vice versa, i.e., at low temperature, it leads to simple search for local improvements [38]. The expression of the energy difference as the objective function is as in Equation 2.2:

$$\Delta E = \Delta f = f_{i+1} - f_i \equiv f(x_{i+1}) - f(x_i) \qquad \text{(Equation 2.2)}$$

A predetermined number of new points x_{i+1} are tested at any specific value of temperature T to simulate obtaining the thermal equilibrium at each temperature. The initial temperature T plays an important role in the successful convergence of the Simulated Annealing algorithm. For example, if the initial temperature (T) is too high, it needs more temperature drops for convergence. In contrast, if the initial temperature is selected to be too low, the search may be missing and the algorithm may be stuck at the local minimum, i.e. it may not find the global minimum [40].

Burke, Li and Qu [35] presented the approach of simulating annealing in order to solve the problem of multi-purpose nurse shift scheduling. First of all, they have developed legal shift patterns for all shifts of the nurses in the schedule. After that, each of the nurses were assigned to these shifts and a structure that could be adapted to the solution which can satisfy all hard constraints was obtained. Subsequently, a function with two options based on evaluating weighted sum and diversification was used to meet the soft constraints. The results obtained by Burke et al. [35] suggest that the method they proposed can be easily applied in nurse scheduling in modern hospitals. The studies of Isken and Hancock [18], Brusco and Jacobs [23], Kundu, Mahato, Mahanty and Acharyya [41], and Ko, Kim, Jeong, Jeon, Uhmn and Kim [42-43] are other examples of the studies using the Simulated Annealing algorithm in the solution of the nurse scheduling problem.

3. Materials and Methods

3.1. Method

As the method of research, one of the heuristic methods developed for the solution of NSP, the Simulated Annealing Algorithm (SAA) was used.

3.2. Problem

The problem of research is to create the best working schedule that provides objective functions by using the simulated annealing algorithm in the context of the default constraints (8 hard, 8 soft, 16 in total) for 15 nurses working in 2 shifts (8 hours between 08:00-16:00 and 16 hours between 16:00-08:00).

3.3. Objective Function and Constraints

The objective function of the nursing shift scheduling problem is given below:

1) The best (optimum) working schedules for nurses should be created according to the constraints given.
2) The total shifts of each nurse should be as close as possible.

In the study, the constraints of the shift scheduling problem where assigning 2 shifts to 15 nurses during the day were given in two groups as hard constraints (HC) and soft constraints (SC).

There are totally 8 Hard Constraints (HC) given as follows:

1) The hospital serves 24 hours a day.
2) Shift times: Day (08.00-16.00), Night (16.00-08.00)
3) Each nurse should work at least 40 hours per week.
4) Nurses are allowed to work for a maximum of 2-night shifts consecutively.
5) A nurse cannot be assigned to a day shift after the night shift.
6) A nurse can work only one shift in a day.
7) The supervisor nurse cannot be assigned to a night shift and cannot work at the weekend.
8) A nurse who is over 50 years old and pregnant cannot be assigned to any shift.

There are totally 8 Soft Constraints (SC) which are given as follows:

1) A minimum of 48 consecutive hour break must be given at least once a week for each nurse.
2) An equal number of weekend breaks should be assigned to each nurse.
3) The total number of day shifts assigned to each nurse should be greater than or equal to the sum of the night shifts in the schedule.
4) The nurse should not be assigned to the shifts on the days when an excuse is indicated and accepted.
5) The nurses cannot have consecutive 72 hours break or more in any week.
6) The number of weekly nurses needed during planning is always constant.
7) The same nurse should not work for 2 days at the weekend.
8) 4 hours of shifts out of normal shifts can be made between the hours of 19.00 and 23.00 when the emergency service is intensive (this restriction will permit to assign for each day, but the duty of overtime assignment will be provided by the emergency service physician working that evening. No overtime shift will be on non-busy days).

4. Findings

4.1. Mathematical expression of the sub-objective functions and writing the appropriate MATLAB codes

In order to solve the problem, objective functions (OF1-2), hard constraints (HC1-8) and soft constraints (SC1-8) were defined and a sample table was created in Microsoft Excel. Calculations of the mathematical expressions which cannot be created directly were formulated in Excel with formulas. A table that complies with all constraints has been created and all formulas in Excel were checked whether they give the correct results for the objective functions and constraints. Then, for each of them, the code was written separately in the MATLAB program, and eventually these codes were combined into a single script.

After the 500 trials, the temperature equation which gave the fastest solution in the nursing scheduling problem was found $T=(T_0*d^{MainCounter})/log_2(1+k)$. Where d is the temperature reduction constant $d=0.80+(0.02*NumberOfNursesOff)$. For example, in case that no nurse is off i.e. all nurses are working, d value takes 0.80, while 8 nurses are off (which is the maximum number of nurses off in our study) i.e. 7 nurses are working (which is the minimum number of working nurses in this study) *d* value takes 0.96.

4.2. A new Assignment Suggestion

In this section, the assignment method which was developed in this study is described. According to this method, till the mainobjective function value falls to a certain critical value, the selection of the assignment technique changes randomly among a number of weekly shift patterns where all weekly shifts in the schedule is changed. However, when the main objective function value falls below the determined critical value, only one random week in the schedule is replaced by one randomly selected from the weekly shift patterns.

The purpose of using these two different assignment techniques together, is the prolongation of the time to reach the optimum solution when only one of these techniques is used. Algorithm in which sub-objective and main objective function values is displayed on the screen in each iteration while working, has been found to be stuck at certain values of the main objective function and faced difficulty in going below that critical value for a long time.

After the 500 trials, the temperature equation which gave the fastest solution in the nursing scheduling problem was found to be $T=(T_0*d^{MainCounter})/log_2(1+k)$, where d is the temperature reduction constant $d=0.80+(0.02*NumberOfNursesOff)$. For example, in case that no nurse is off, i.e., all nurses are working, *d* is 0.80, whereas when 8 nurses are off (which is the maximum number of nurses off in our study), i.e., 7 nurses are working, (which is the minimum number of working nurses in our study) *d* is 0.96. During developing the algorithm, the critical value was observed to differ depending on the number of nurses off. In case of 0-8 nurse off, it was determined that the equation $CriticalValue=140+70*log2[(2*NumberOfNursesOff)+1]$ would be appropriate for the critical values observed in the first stage. Thus, this equation gave the critical values of 140 – 250.94 – 302.53 – 336.51 – 361.89 – 382.16 – 399.03 – 413.48 – 426.12 for 0-8 nurses off respectively. However, as the algorithm was stuck in the values above 140 for a long time, the critical value was set to 251 as exceptional for the cases where the number of nurses off was 0-2, and was set to $CriticalValue=251+(NumberOfNursesOff-2)*100$ for the cases where the number of nurses off was 7 and above. The critical values in the change of assignment technique were 251 - 251 - 251 – 336.51 – 361.89 – 382.16 – 399.03 - 751 - 851 - 951, respectively, according to the 0-9 nurse off. The algorithm found schedules within 30 seconds for 0-2 nurse off, 45 seconds for 3-5 nurses off, and 60 seconds for 6-8 nurses off. The algorithm provided an average time of 34 seconds for 0-8 nurses off. The algorithm was tested for the challenging situation which was the 9 nurses off (i.e. 6 working nurses), and it was found that the algorithm could reach the optimum solution under 5,000 iterations (min.456-max.4,864 iterations), 1,953 iterations (appx. 195 seconds) as an average in all tests.

5. Results and Discussion

Considering that he algorithm reached a solution under 600 iterations (appx. 60 seconds) in the case of 0-8 nurses off, under 5,000 iterations (appx. 500 seconds) in the case of 9 nurses off (1953 iterations, appx. 195 seconds as an average), also when considering that the number of constraints (16 constraints) and the number of days (28 days) in the scheduling problem discussed are quite large, and that the algorithm could always find a solution for all constraints in every run, it can be concluded that the number of iterations for a solution is successful. In cases where the number of constraints is less than 5 and the number of days is 7 or 14, the solution can be reached more quickly. However, there are studies in which the solution could not be reached within 10,000 iterations for 14-nurse and 14-day scheduling problem. For example; Chen, Lu, Lu and Zhu [45] reported that they reached the solution of the 14-day and 15-nurse scheduling problem including the condition of providing only 3 constraints, within 100,000 iterations by using the cooling once at each 1,000 iterations with 0.99 (slow annealing) or 0.95 (rapid annealing) cooling rate.

References

[1]. Kumar, B. S., Nagalakshmi, G., Kumaraguru, S. (2014). A shift sequence for nurse scheduling using linear programming problem, *IOSR Journal of Nursing and Health Science (IOSR-JNHS)*, 3(6):24-28.

[2]. Wang, B. A. (2016). Using linear programming to flex nursing staff. Retrieved from http://floridahfma.squarespace.com/s/Xiao-Angelica-Wang.pdf

[3]. Karpuz, E. (2015). *Nurse scheduling and rescheduling problem under uncertainty.* Master's Dissertation, Institute of Science and Technology, Middle East Technical University, Turkey.

[4]. Kundu, S., Acharyya, S. (2008). A SAT approach for solving the nurse scheduling problem, *TENCON 2008 - 2008 IEEE Region 10 Conference*, 1–6. Retrieved from http://doi.org/10.1109/TENCON.2008.4766380.

[5]. Tekin, E. (2017). *Multi-objective nurse scheduling with shift preferences in a surgical suite*. Master's Dissertation, Institute of Science and Technology, Kadir Has University, Turkey.

[6]. Trilling, L., Guinet, A., Le Magny, D. (2006). Nurse scheduling using integer linear programming and constraint programming. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, *12* (Part 1). Retrieved from https://www.sciencedirect.com/science/article/pii/S1474667015360602/pdf?md5=653a4e10b24c81251ebf19996a3ebcf3&pid=1-s2.0-S1474667015360602-main.pdf

[7]. Tein, L., Ramli, R. (2010). Recent advancements of nurse scheduling models and a potential path, *Proceedings of the 6th IMT-GT Conference on Mathematics, Statistics and its Applications (ICMSA2010).* Kuala Lumpur, Malaysia.

[8]. Cheang, B., Li, H., Lim, A., Rodrigues, B. 2003, Nurse rostering problems -A bibliographic survey. *European Journal of Operational Research*, 151:447–460.

[9]. Clark, A., Moule, P., Topping, A., Serpell, M. (2013), Rescheduling nursing shifts: Scoping the challenge and examining the potential of mathematical model-based tools. *Journal of Nursing Management*, 23(4):411–420.

[10]. Warner, M. (1976). Scheduling nursing personnel according to nursing preferences: A mathematical programming approach, *Operations Research*, 24(5):842–856.

[11]. Tien, J., Kamiyama, A. (1982). On manpower scheduling algorithms. *SIAM Review*, 24(3):275-287.

[12]. Bradley, D., Martin, J. (1991). Continuous personnel scheduling algorithms: A literature review. *Journal of the Society for Health Systems*, 2(2):8-23.

[13]. Sitompul, D., Randhawa, S. (1990). Nurse scheduling models: A state-of-the-art review, *J Soc Health Syst.*, 2(1):62-72.

[14]. Burke, E. K., De Causmaecker, P., Berghe, G. V., Landeghem, H. (2004b). The state of the art of nurse rostering. *Journal of Scheduling*, 7(6): 441-499.

[15]. Warner, M., J. Prawda. (1972). A mathematical programming model for scheduling nursing personnel in a hospital. *Management Science*, 19:411-422.

[16]. Beaumont, N. (1997). Scheduling staff using mixed integer programming. *European Journal of Operational Research*, 98(3):473-484.

[17]. Jaumard, B., Semet, F., Vovor, T. (1998). A generalized linear programming model for nurse scheduling. *European Journal of Operational Research*, 107(1):1–18.

[18]. Isken, I., Hancock, W. (1990). A heuristic approach to nurse scheduling in hospital units with non-stationary, urgent demand and a fixed staff size. *J Soc Health Syst.*, 2(2):24-41.

[19]. Randhawa, S. U., Sitompul, D. (1993). A heuristic-based computerized nurse scheduling system, *Computers & Operations Research*, 20(8):837–844.
[20]. Aickelin, U., Dowsland, K. (2000). Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem, *Journal of Scheduling*, 3(3):139-153.
[21]. Aickelin, U., Dowsland, K. A. (2004). An indirect Genetic Algorithm for a nurse-scheduling problem. Computers & Operations Research, 31(5):761–778.
[22]. Abramson, D. (1991). Constructing school timetables using simulated annealing: Sequential and parallel algorithms, *Management Science*, 37(1):98–113.
[23]. Brusco, M. J., Jacobs, L. W. (1993). A simulated annealing approach to the cyclic staff-scheduling problem. *Naval Research Logistics*, 40(1): 69–84.
[24]. Burke, E. K., De Causmaecker, P., Vanden Berghe, G. (1998). A hybrid tabu search algorithm for the nurse rostering problem. *SEAL'98 Selected papers from the Second Asia-Pacific Conference on Simulated Evolution and Learning on Simulated Evolution and Learning*, 1585:187-194.
[25]. Dowsland, K. A. (1998). Nurse scheduling with tabu search and strategic oscillation. *European Journal of Operational Research*, 106(2-3):393–407.
[26]. Dowsland, K. A., Thompson, J. M. (2000). Solving a nurse scheduling problem with knapsacks, networks and tabu search. *Journal of the Operational Research Society*, 51(7):825–833.
[27]. Valouxis, C., Housos, E. (2000). Hybrid optimization techniques for the work shift and rest assignment of nursing personnel. *Artificial Intelligence in Medicine*, 20(2):155–175.
[28]. Ikegami, A., Niwa, A. (2003). A subproblem-centric model and approach to the nurse scheduling problem. *Mathematical Programming*, 97(3): 517–541.
[29]. Burke, E. K., Cowling, P., De Causmaecker, P., Vanden Berghe, G. (2001). A memetic approach to the nurse rostering problem," *Applied Intelligence*, 15(3):199-214.
[30]. Ozcan, E. (2005). *Memetic algorithms for nurse rostering*. Retrieved from https://www.researchgate.net/profile/Ender_Ozcan/publication/221579196_Memetic_Algorithms_for_Nurse_Rostering/links/02e7e524026ab4cee6000000/Memetic-Algorithms-for-Nurse-Rostering.pdf.
[31]. Kawanaka, H., Yamamoto, K., Yoshikawa, T., Shinogi, T., Tsuruoka, S. (2001). Genetic algorithm with the constraints for nurse scheduling problem. *Proceedings of the 2001 Congress on Evolutionary Computation*.
[32]. Tanomaru, J. (2001). Staff scheduling by a genetic algorithm with heuristic operators. *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*. Retrieved from https://www.ufsj.edu.br/portal2-repositorio/File/nepomuceno/00489191.pdf
[33]. Burke, E. K., De Causmaecker, P., Petrovic, S. and Vanden Berghe, G. (2004a). *Variable neighborhood search for nurse rostering problems*, in M. G. C. Resende, J. P. De Sousa (eds.), *Metaheuristics: Computer Decision-Making* (Combinatorial Optimization Book Series), Kluwer, Chapter 7, 153-172.
[34]. Li, J., Aickelin, U. (2006). *Bayesian Optimization Algorithm for nurse scheduling*, In Pelican, M., Sastry, K., Cantú-Paz, E. (Eds), Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications, Springer, Chapter 17, 315-332.
[35]. Burke, E. K., Li, J., Qu, R. (2012). Pareto-based optimization for multi-objective nurse scheduling. In: Boros, E. (ed.) *Annals of Operation Research*, 196(1): 91-109, Springer, US.
[36]. Arora, J. S. (2004). Introduction to Optimum Design, Elsevier Academic Press.
[37]. Le, K. N. (2011). *A study of evolutionary multiobjective algorithms and their application to Knapsack and nurse scheduling problems*. Doctoral Dissertation. University of Nottingham, UK.
[38]. Rosocha, L., Vernerova, S., Verner, R. (2015). Medical staff scheduling using Simulated Annealing, *Quality Innovation Prosperity*, 19(1):1–11.
[39]. Aarts, E., Korst, J., Michiels, W. (2007). *Theoretical aspects of local search*. Springer, Berlin.
[40]. Gulmez, E. (2014). *Minimum weight design of carbon/epoxy laminated composites for maximum buckling load using simulated annealing algorithm*. Master's Dissertation, Graduate School of Engineering and Sciences, İzmir Institute of Technology, Turkey.

[41]. Kundu, S., Mahato, M., Mahanty, B., Acharyya, S. (2008). Comparative performance of simulated annealing and genetic algorithm in solving nurse scheduling problem. *IMECS 2008 -International Multiconference of Engineers and Computer Scientists (19-21 March 2018)*.

[42]. Ko, Y.-W., Kim, D.H., Jeong, M., Jeon, W., Uhmn, S., Kim, J. (2013a). An efficient method for nurse scheduling problem using simulated annealing, 20:82–85. Retrieved from http://onlinepresent.org/proceedings/vol20_2013/20.pdf

[43]. Ko, Y.-W., Kim, D.H., Jeong, M., Jeon, W., Uhmn, S., Kim, J. (2013b). An improvement technique for simulated annealing and its application to nurse scheduling problem. *International Journal of Software Engineering and Its Applications*, 7(4):269–278.

[44]. GNU Octave. 2018. *GNU Octave Version 4.4*.1. File name: octave-4.4.1-w64-installer.exe. Retrieved from https://ftp.gnu.org/gnu/octave/windows/octave-4.4.1-w64-installer.exe

[45]. Chen, F., Lu, S., Lu, T., Zhu, S. (2017). *Nurse Scheduling*. Columbia University.