Nurse scheduling problem using Simplified Swarm Optimization

Wei-Chang Yeh$^1$, Chyh-Ming Lai$^2$, and Meng-Hang Tsai$^1$

$^1$Integration and Collaboration Laboratory, Department of Industrial Engineering and Engineering Management, National Tsing Hua University, Hsinchu 300, Taiwan

$^2$The Institute of Resources Management and Decision Science, Management College, National Defense University, Taipei 112, Taiwan

* Corresponding author: hang772244@gmail.com

Abstract. In the past, many researchers tried to find a high-quality nurse scheduling by using mathematical programming approach. However, mathematical programming approach is hard to solve the highly complex problem in the real world, because there are so many constraints. In this paper, we present Simplified Swarm Optimization (SSO), which is the stochastic optimization method proposed by Yeh [1], to solve the real world problem of Nurse Scheduling Problem (NSP). SSO designs the new update mechanism and aim to overcome the disadvantage of Particle Swarm Optimization (PSO) in discrete problem. The objective function is to maximize the fairness within all nurse’s scheduling.

1. Introduction

Manpower is the most important resource of an organization. Ensuring an abundant supply and distribution of human resources is an essential problem in some places, especially in hospitals. The operation of the hospital highly relies on manpower. Staff rostering is one of the steps in scheduling human resources processes, which assign a particular person in a particular day and time. Therefore, having a flexible scheduling and balancing the supply and the need of an organization become significant.

The Nurse Scheduling Problem (NSP) is a famous NP-hard problem and has been highly discussed for the past 30 years. The nurse scheduling is essentially the allocation of resources to the nursing staff of the medical institution. Not only allocating the appropriate manpower to the appropriate work at the appropriate time but also meet all the constraints which regulated by the regulations of Labor Laws. Scheduling becomes complicated due to many considerations including regulations of Labor Laws, needs of patient, coverage for every shifts, nurse preferences, and other considerations. Hospital wards must be staffed 24 hours for every time slot in a day. Scheduling needs to achieve high patient satisfaction where considering both legal and cost constraints at the same time. Consequently, researchers utilize such as mathematical programming, heuristic algorithm and artificial intelligence to achieve the goal.

One of the meta-heuristic algorithm that is suitable to solve this complex real world problem is Simplified Swarm Optimization (SSO). SSO was proposed by Yeh in 2009 [1].It was designed to solve the deficiencies of PSO in discrete problems originally, but the new update mechanism made SSO more simplicity, efficiency and flexibility. In this paper aim to design a SSO algorithm to solve
NSP without losing the original mechanism of SSO. It can not only find a high-quality solution but also save time for the scheduler.

2. Literature references
NSP is a highly constrained and complex problem of allocating working shifts and days for holiday to nurse’s schedule. In the process of scheduling, the hard limits specified by all laws and regulations must be met. Inappropriate scheduling may affect nurses working performance. In addition, inadequate nurses are also prone to over-shifting, leading to long hours of overtime or insufficient rest time. Many researchers adopt heuristic algorithm to solve NSP such as particle swarm optimization, Ant Colony Optimization or Genetic Algorithm.

Gutjahr [2] adopts Ant Colony Optimization (ACO) to solve regional dynamic scheduling problems. In this paper, work patterns, hospital preferences, nurse preferences, working hours, Nurses’ ability and many other restrictions have also joined the Decision Support Systems (DSS) to make the shifter’s preference for the hospital and the nurse's preference more flexible during the shift, and the results prove that ACO is highly restrictive. Both have good results and are superior to the solutions generated by greedy algorithms.

Kawanaka, H., et al. [3] proposes the Genetic Algorithm (GA) without adopting the conventional method which reduce the objective function by calculating the constraint of the violation. The previous method may make the adaptation function of the final result the best solution of the past generation, but still violate some restrictions. Therefore, the results still need to be manually fine-tuned to make it comply with the current regulations. This paper propose a new method of coding and genetic operations to follow all the hard constraints in the initial feasible solution. And in the process of updating and iterating, the generated solution is kept feasible, and the result of the GA operation is guaranteed to meet all the restrictions. The results show that the method proposed in this paper can greatly reduce the search range of GA solution and improve the efficiency of the solution.

Altamirano [4] utilizes Particle Swarm Optimization (PSO) to solve Anesthesiology Nurse Scheduling Problem (ANSP). Dividing the day into four time slots and set limits on every slots. The assignment has to consider all the constraints such as regulations by law, coverage for every time slot, individual preference. These restrictions allow nurses to have enough rest. The objective function is to minimize the fairness among all the nurses. This paper also proposes a new evaluation for NSP: Ideal Work Distribution (IW) , Current Work Distribution (CW) , Fair Work Distribution (FW). These new evaluation are designed to distinguish the differences between very similar solutions. And the result shows that it can spend less time to find a quality solution.

3. NSP description

| Type of Shift | Start-End Time |
|---------------|----------------|
| Day (D)       | 7:00 – 16:00   |
| Evening (E)   | 13:00 – 21:00  |
| Night (N)     | 20:00 – 4:00   |
| Late Night (LN)| 2:00 – 10:00  |

In this paper, consider a 7 days scheduling, the nurses are assigned to one of these shifts each day (Day (D), Evening (E), Night (N) or, day off (O)). By referring Table 1 for Day, Evening and Night shift, the period is 8 hours. NSP limits can be separated into hospitals, laws and nurse preferences. There are several constraints must to be satisfied are the following:

- There can be no more than three different shift type in a week.
- At least one day off in a week.
- Each nurse take only one shift in a day.
- Coverage of shifts must be meet the least number of nurse’s demand every day.
- Isolated working days is prohibited.
There must be no more than two days off per week.

- If a nurse works a LN, the following day could be either Day Off or Late Night Shift.
- If a nurse works an N, the following day could be either Night Shift, Late Night Shift or Day Off.
- Each nurse cannot take consecutive Late Night Shift more than four days a week.

3.1. Objective function

The purpose of this study is to maximize the fairness between every nurse. In previous research,[5] the authors proposed a fitness function for NSP. It could be formulated as following:

$$\text{Minimize } Z = P_{\text{max}} - P_{\text{min}}$$

where $P_{\text{max}}$ is the maximum penalty values among all nurses and $P_{\text{min}}$is the minimum. The concept of this objective function is to provide a fair scheduling for all nurses. The penalty value are shown in Table 2. By calculating the sum of the penalty value associate to the shifts for the nurse. The closer the difference between the maximum and the minimum, the higher the fairness of the nurse’s scheduling.

$$\text{Table 2. Penalty associated to the shifts.}$$

| Type of Shift | Day Shift (D) | Evening Shift (E) | Night Shift (N) | Late Night Shift (LN) |
|---------------|---------------|-------------------|-----------------|----------------------|
| Monday        | 1             | 1.2               | 1.4             | 1.6                  |
| Tuesday       | 1             | 1.2               | 1.4             | 1.6                  |
| Wednesday     | 1             | 1.2               | 1.4             | 1.6                  |
| Thursday      | 1             | 1.2               | 1.4             | 1.6                  |
| Friday        | 1             | 1.2               | 1.4             | 1.6                  |
| Saturday      | 1.2           | 1.4               | 1.6             | 1.8                  |
| Sunday        | 1.2           | 1.4               | 1.6             | 1.8                  |

3.2. Simplified Swarm Optimization (SSO)

SSO is a population-based optimization algorithm proposed by yeh in 2009[1]. It was initially design to fix the deficiencies of PSO[6] in solving discrete problems. The advantages of SSO include simplicity, efficiency and flexibility [7, 8]. The update mechanism uses the ladder function to change the update method for different types of problems. With the adjustment of the update method, it can also deal with continuous problems [9].

In initialization phase, multiple candidate feasible solutions in SSO are generated randomly. For any solution $X_i^t = (x_{i1}^t, x_{i2}^t, x_{i3}^t, ..., x_{in}^t)$ in the $t^{th}$ iteration. Every elements in $X_i^t$ are updated successively. Each element are selected one of the updated mechanism of $t g_i, p\text{Best } p_{ij}^{t-1}$, current value $x_{ij}^{t-1}$ or a new random value $x$ depending on a uniform random feasible number $\rho$ between 0 and 1. $C_w, C_p$ and $C_g$ are predetermined parameter to construct four intervals to decide the updated way of $X_i^t$ as shown in Figure 1.
The initial feasible solution is designed to satisfy all the constraints. It randomly selects a feasible nurse to and assigns to a shift. The assignment procedure is completed until satisfying the shift least demand of all coverage. In case the assignment cannot find a feasible solution, the procedure tried another assignment until it becomes feasible as shown in Figure 2.

### 4. Experiment result

A numerical case are showed in Table 3. The proposed algorithm has been implemented in Python 3.7.0 language programming and executed on Intel(R) Core(TM) i5-4960 CPU 3.5GHz with RAM 12GB and window 10 64 bit operation system. The setting of SSO parameters is as follow: particle number is 100; three parameters used in the update mechanism are $c_w = 0.45$, $C_p = 0.75$ and $C_g = 0.9$ to construct four intervals. The termination criteria for SSO evolution would stop until the maximum iteration. Table 4 are the result of the experiment.

#### Table 3. Parameters of the experience tests.

| Nurses | Days | Day | Evening | Night | Late Night |
|--------|------|-----|---------|-------|------------|
| P1     | 10   | 7   | 2       | 1     | 1          |
| P2     | 14   | 7   | 2       | 2     | 2          |
| P3     | 22   | 7   | 4       | 3     | 2          |
Table 4. Experimental Results, CPU in seconds.

|       | Pmax | Pmin | Best Obj | CPU |
|-------|------|------|----------|-----|
| SSO   |      |      |          |     |
| P1    | 7.6  | 7    | 0.6      | 82  |
| P2    | 7.8  | 6.8  | 1.0      | 196 |
| P3    | 8    | 6.8  | 1.2      | 137 |
| PSO   |      |      |          |     |
| P1    | 7.8  | 7    | 0.8      | 65  |
| P2    | 8    | 6.8  | 1.2      | 500 |
| P3    | 8.4  | 6.8  | 1.6      | 387 |

5. Conclusions
In this paper, we propose a Simplified Swarm Optimization algorithm for dealing with NSP problem. The objective function is to maximize the nurse fairness. By using SSO to avoid scheduler spent too much time while arranging the schedules. Not only reduce the time-consuming manual shifts, but also meets the preferences of nurses and achieves the fairness of scheduling among all the nurses. And the results show that SSO can find a better solution than PSO. Our approach can adjust the nurse's class while maintaining the feasibility of scheduling. The SSO update mechanism is also more suitable for solving discrete problems in the real world than PSO. Figure 3 ~Figure 5 are the results of our research.

Figure 3. Solution for P1.

Figure 4. Solution for P2.
6. References

[1] W. C. Yeh, Expert Syst Appl 36, 5(2009)
[2] W. J. Gutjahr, M. S. J. C. Rauner, O. Research 34, 3, 642-666 (2007)
[3] H. Kawanaka, K. Yamamoto, T. Yoshikawa, T. Shinogi, and S. Tsuruoka, Trans. Evol. Comput. (2001)
[4] L. Altamirano, M. C. Riff, I. Araya, and L. Trilling, L. J. I. J. o. C. I. S., 5, 1, 111-125 (2012)
[5] L. Trilling, A. Guinet, and D. Le Magny, IFAC, 39, 3, 671-676 (2006)
[6] R. Eberhart and J. Kennedy, International Symposium on Micro-machine and Human Science (1995)
[7] W. C. Yeh and C. M. Lai, PloS one, 10, 9(2015)
[8] W. C. Yeh, C. M. Lai, and K. H. Chang, Neurocomputing, 173, part3, 15 (2016)
[9] W.C. Yeh, KNOWL-BASED SYST, 82, page60 – 69(2015)