

# Using Bees Algorithm For The Nurse Scheduling Problem

ANDREA CAMIL NUESCA, Ateneo de Davao University

GENE MARCO MENDOZA, Ateneo de Davao University

---

**Abstract:** Making schedules for staff nurses is considered to be a complex problem that has great impact on hospitals on a day-to-day basis worldwide. In this work, the proponents have successfully applied Bees Algorithm (BA), a Swarm Intelligence metaheuristic approach, to solve the said problem and to generate an effective and efficient schedule. This approach practices a resemblance between the way in which bees in nature search for their food, and the way in which optimization algorithm search for an optimal solution in combinatorial optimization problems.

**General Terms:** nurse scheduling problem, bees algorithm

**Additional Key Words and Phrases:** nurse scheduling, nurse rostering, hospital personnel scheduling, staff scheduling

---

## 1. INTRODUCTION

### 1.1 Background

All health care institutions must constantly provide duty rosters for its personnel. A suitable nurse scheduling has a great impression on the quality of health care service, staff-and-patient satisfaction, and administrative workload.

Not like other organizations, hospitals are required to be operated 24 hours a day over seven days in a week. Generating a good nursing schedule is quite difficult, because it consists of different duty rosters on different days and shifts. In some cases, other staff nurses are permitted to have pre-set shifts requests. Usually, head personnel spend an extensive amount of time producing rosters particularly when a lot of requests have been appealed.

As of the moment, almost all nurse scheduling problems in hospitals in Davao City were solved manually. It is usually called “self-scheduling” in the literature. Making schedules by hand is considered a relatively time-consuming duty. Head personnel had no automatic instrument to check the quality of a constructed schedule.

It is reasonably important to use a systematic approach in generating good duty rosters because in health care institutions, it is not acceptable not to provide complete patient care needs and staff requirements. Automated approaches can offer a substantial prospective for refining the scheduling process and the quality of the duty rosters. One of the major contributions of automated nurse scheduling is a relatively efficient and time-conserving for the hospital’s managerial staff.

A high quality nurse duty roster can lead to further contentment and an operative workforce. Mathematical or heuristic approaches can straightforwardly provide a number of solutions, test on the schedule quality, and distribute the work uniformly among nurses. Many approaches such as Genetic Algorithms, Tabu Search and Simulated Annealing were used to solve this scheduling problem.

Moreover, in the literature, this complex problem can be answered by combinatorial optimization algorithms. One of the combinatorial optimization algorithms developed is the Bees Algorithm (BA). This metaheuristic algorithm is considered to be a new Swarm Intelligence

approach, which is inspired by the natural honey bees, and artificial bees are used as agents to communicate with each other and with the environment.

## 1.2 Problem Statement

The proponents will investigate BA in solving nurse scheduling problem. In addition, this study seeks to answer the following questions:

1. What are the hard and soft constraints to consider in generating schedules?
2. What are the objectives/criteria by which a schedule is considered to be optimal?
3. How schedules are generated using the algorithm?
4. How to present illustrative example schedules, showing the utility of the proposed algorithm?

## 1.3 Objectives

The main objective of the study is to implement BA in generating good schedules. In addition, this study desires to accomplish the following objectives:

1. To identify the hard and soft constraints to consider in generating schedules;
2. To identify the objectives/criteria by which a schedule is considered to be optimal;
3. To explain how the nurse scheduling are generated using the algorithm;
4. To present an automated scheduler program, showing the utility of the proposed algorithm.

## 1.4 Significance

The proposed approach in this study could contribute to the Bees Algorithm and to the healthcare community. First and foremost, it is timely and relevant. This study could yield answers that will contribute to health care domain, advanced knowledge and development. Since solving the problem is considered to be time-consuming, it could effectively help in improving human conditions and practices such as the manual scheduling process. For the Bees Algorithm, this study could also add further development and applications for future studies. It could also enhance the algorithm's capacity of handling large-scale problems, while providing good solutions. In essence, the said problem is worth studying and worth the time and effort to be spent on it.

## 1.5 Scope and Limitations

The proponents conduct a study on how BA can be implemented in generating good schedules for nurses. This study handles one (1) nurse station from Southern Philippines Medical Center (SPMC), located at Bajada, Davao City. The study also considers four (4) nursing shifts (day shift, *am* (7:00am-3:00pm); night shift, *pm* (3:00pm-11:00pm); graveyard shift, *g* (11:00pm-7:00am); off shift, *o*). Moreover, the scheduling involves 4-week planning. This study, however, will not cover all the hospitals in Davao City. In addition, for the implementation, PHP is the programming language is used for the automated program.

## 2. REVIEW OF RELATED LITERATURE

### 2.1 Nurse Scheduling Problem / Nurse Rostering Problem

#### 2.1.1 Nurse Scheduling Definition

Wren (1996) defined scheduling as “the allocation, subject to constraints, of resources to objects being placed in space-time, in such a way as to minimize the total cost of some set of the resources used” and rostering as “the placing, subject to constraints, of resources into slots in a pattern. One may seek to minimize some objective, or simply to obtain a feasible allocation. Often the resources will rotate through a roster. Once shifts have been produced showing the daily work of personnel, these shifts are placed into a roster to show in which shifts are worked by individuals on particular days.” He also classified rostering as distinct case of scheduling, which refers to generic scheduling problem that handles different decision levels. For the purposes of this research, nurse scheduling and rostering are considered as synonym.

#### Problem Definition and Constraints

The objective is to search for a schedule that satisfies a given set of constraints. Nevertheless, in practice, the preferences of nurses must also be satisfied as much as possible. As a result, two (2) categories of shift constraints are involved: hard constraints and soft constraints. Hard constraints consist of daily restrictions arising from legislative laws and are always satisfied in generating schedules; if these constraints fail, then the entire schedule is invalid. Soft constraints, however, arise from nurse preferences and are satisfied to the highest degree possible; meaning not meeting them does not make the entire schedule invalid. So, the main challenge is to incorporate these two (2) constraints into the scheduling problem, so as to improve the schedule quality (Mutingi and Mbohwa, 2014).

#### Typical Constraints of Nurse Scheduling Problem

<i>Constraints</i>	<i>Description of Constraints</i>
<i>Daily Restrictions</i>	<i>Assign each nurse at most one shift per day</i>
	<i>The assigned <math>d</math>, <math>n</math>, or <math>l</math> shifts <math>&gt;</math> required <math>d</math>, <math>n</math>, or <math>l</math> shifts respectively</i>
	<i>A <math>(d-n)</math>, <math>(n-l)</math>, <math>(l-d)^*</math> shift combination (sequence) is not permissible</i>
	<i>Assigned legal holidays = number of legal holidays</i>
	<i>Interval between night shifts should be at least 1 week</i>
<i>Nurse Preferences</i>	<i>Preferred or desired day off or holidays</i>
	<i>Fairness or equality of shifts for each nursing staff</i>
	<i>Congeniality – compatible or preferable shift assignments among work mates</i>

Table 2.1 Mutingi and Mbohwa’s study (2014) –  $d$ (am),  $n$ (night),  $l$ (late night),  $o$ (off)\*

#### 2.1.2 Approaches to Nurse Scheduling/Rostering

At the present moment, there are two (2) common approaches towards staff scheduling. Fixed rostering is where staffs are allocated a fixed shift pattern over a long or even indefinite time period. Flexible rostering is where each rostering period is planned individually (typically 4-6 weeks at a time).

### 2.1.2.1 Departmental Scheduling/Rostering

Departmental rostering is conducted by the head staff/nurse. The roster may be authorized by a senior manager before the schedule will be released to the staff.

### 2.1.2.2 Team Scheduling/Rostering

The staffs are divided into teams and a nominated member of each team has responsibility for rostering. In consultation with team members, it is usual for team leaders to meet and discuss the overall roster design; a more senior manager may take responsibility for the final roster.

### 2.1.2.3 Self- Scheduling/Rostering

In self-scheduling, the roster is prepared by the ward staff. A more senior manager will usually authorize the final roster.

Based on Silvestro and Silvestro's evaluation on nurse rostering practice (2000), it has recommended that departmental scheduling is more appropriate in large wards with complex rostering problems, while team scheduling is appropriate for medium-sized wards, and self-scheduling is for small wards.

## 2.1.3 Common Nurse Scheduling Problem Solutions

The solution approaches that have been commonly proposed to NSPs are classified in three categories: optimization, metaheuristics, and artificial intelligence.

### 2.1.3.1 Optimization Approach

Optimization approaches generally perform optimization concepts which either maximizing or minimizing some objectives through Mathematical Programming (MP). Mathematical programming methods are suitable for generating lowest cost optimal outcomes for scheduling problems. Optimization using mathematical programming is categorized as linear/single-objective or multi-objective/goal programming.

#### 2.1.3.1.1 Linear/Single-Objective Mathematical Programming

Linear/single-objective mathematical programming, such as Integer Programming (IP), linear programming (LP), and mixed-integer programming (MIP), is a technique that only maximizes a goal which is preferred by the decision-maker. However, based on Ernst et al.'s work (2004), mathematical programming formulations are more limiting compared to constraints and objectives which are simply expressed. Therefore, these approaches are commonly applied to simplified real world scheduling problem. Moreover, implementing an integer programming method is quite hard and time-consuming.

#### 2.1.3.1.2 Multi-Objective/Goal Programming

Multi-objective/goal programming outlines a target level for each objective and relative priorities to attain all goals provided. This approach aims at finding a solution which is closest to each of the goals in the order of the objectives given. The major advantage of this approach is the flexibility it permits in choosing goals and priorities; it can further account such vital factors such as the nurse's personal preferences and desired staffing requirements.

Burke et al. (2002) used multi-objective approach for nurse rostering in Belgian hospital facilities. In the study, there is a standard for each time-related constraint. One of the main advantages of this method is the possibility for users to use their preference for definite constraints, instead setting up abstract cost parameters. The compensation of constraints has been controlled by the weights. A foremost advantage of the multi-objective approach is that it allows a better handling of different constraints by taking the likely ranges for the criteria into attention. However, this approach required additional progress (to handle the complete range of constraints required by large hospitals) before it could be engaged straight into a hospital ward.

### 2.1.3.2 Metaheuristics Approach

In recent years, modern metaheuristics such as Tabu Search, Genetic Algorithm and Simulated Annealing, have been proved to be effective in finding near-optimal solutions for a variety for real world problems. Metaheuristic approaches are frequently used to solve hard and combinatorial optimization problems. In Ernst et al.'s work (2004), they provided three (3) metaheuristic characteristics. First, these approaches are considered to be relatively robust, which means they can provide a reasonable and good feasible solution. Second, these approaches are easy to implement. Lastly, metaheuristics provide an easy approach to handle complex objectives.

#### 2.1.3.2.1 Tabu Search

Tabu Search (TS) is searching iteratively from one solution to another by moving in a neighborhood space with the assistance of an adaptive memory.

Bellantini et al. (2004) discussed algorithms to resolve a scheduling problem in a hospital ward in Italy. The research engaged with a huge set of detailed constraints, which they split into coverage constraints and contractual and operational requirements. Certain inevitable reductions have been combined in the proposed model, for example allowing a deviation among the coverage requirements and the definite number of scheduled nurses. As a result, a software system is presently in use in the hospital. The study's outcomes improved upon those manually generated schedules and the computation time is remarkably low.

#### 2.1.3.2.2 Simulated Annealing

Simulated annealing (SA) is a generalization of a Monte Carlo method for examining the equations of state and frozen states of n-body systems (Metropolis et al., 1953).

Suman and Kumar (2006) provided a survey of three metaheuristics (SA, SA with TS, and chaos SA) in solving nurse rostering problem. Mainly, SA took less CPU time than GA. Nevertheless, SA is more time-consuming than TS. The researchers found out that SA alone is less successful to find feasible solution in NSP, and concluded that the performance of SA can be improved by using SA with another algorithm and other algorithmic parameters.

#### 2.1.3.2.3 Genetic Algorithms (GA)

Genetic Algorithm is a metaheuristic approach based on natural selection and genetic recombination. The algorithm works by choosing solutions from the current population and then applying genetic operators – such as mutation and crossover – to

create a new population. The algorithm then efficiently exploits historical information to speculate on new search areas with improved performance. The loop continues until a stopping criteria is met (Goldberg, 1989; Webster et al., 1998)

Aickelin and Downsland (2004) improved a proposed GA to solve a nurse scheduling problem in UK. Rather than working straight with sets of possible solutions and treating the constraints by applying penalty functions or repairs, the researchers proposed an indirect approach in which the duty of balancing optimization and constraint consummation is common between a greedy heuristic and the genetic algorithm. The individuals are signified by permutations of the available staff and the heuristic is used to generate timetables by scheduling nurses to their right shifts in the specified order. GA was compared to TS to test efficiency. With the proposed GA, the worth of the results proved that GA is more flexible and robust than TS.

#### 2.1.3.2.4 GA vs. TS Comparison

In Dias et al. (2003)'s study, two (2) metaheuristics, GA and TS, were established and applied in order to construct nursing schedules at large hospitals in Brazil. Both demonstrated very efficient results and constructed schedules in all real test instances delivered by the hospital. The study concluded that GA was able to provide a better quality of schedule than TS. However, TS is more time efficient than GA in solving NRP.

#### 2.1.2.3 Artificial Intelligence Approach

Artificial Intelligence (A.I.) approach develops computational methods that meet or exceed the abilities of the human mind in areas such as perception, reasoning, learning, and planning (<http://www.cs.duke.edu/research/artificial/>, The Department of Computer Science at Duke University).

##### 2.1.2.3.1 Constraint Programming

Constraint programming (CP) has the advantage of describing constraint logic easily. According to Ernst et al.'s work (2004), CP is considered to be suitable for highly constrained problems. In addition, optimization with CP is inefficient and more research and development is necessary to identify the best way to combine CLP and optimization techniques used in solving scheduling problems.

Muslija et al. (2000) tried to produce cyclical solutions for a workforce scheduling problem by using constraint programming. Rotating workforce timetables are useful for the workers' health and satisfaction, and therefore increase their work performance. The proposed method generated acceptable shift sequences, which satisfied the legal constraints, in a one week planning period. Significant schedule characteristics were work blocks lengths and optimal weekend characteristics. Individual preferences and further constraints could be applied when generating this workforce rotating schedule. The proposed approach could help in generating better schedules fast but is probably too basic to be of use in large healthcare settings.

##### 2.1.2.3.2 Expert Systems – Decision Support Systems

Expert systems aim to imitate how an expert human scheduler would generate a good staff roster.

Petrovic et al. (2002) tackled a case-based reasoning method to a nurse scheduling problem. The researchers attempted to automate the self-scheduling method that is applied in a specific UK hospital. Guidelines that were considered when generating a schedule are the following: provide the appropriate skill combination, meet the coverage, nurse-to-patient ratios, and a set of time-related constraints. Hospital organizers mixed and repaired incomplete rosters that have been constructed from the individual preferences and requests. This is quite remarkable that the practice duplicates the human style of reasoning in which problems are solved using past experience, on the evidence that similar problems need alike solutions. Moreover, the researchers recommended an approach in which it could be combined with a metaheuristic technique.

#### 2.1.4 Existing Approach - Swarm Intelligence (Metaheuristic) Approach

The Swarm Intelligence (SI) approach mimics the collective exploration technique of the swarms in the nature to solve problems. It is used as a population based approach to solve the combinatorial problems. Some common Swarm algorithms are Evolutionary Algorithms, Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and the Bees-inspired Algorithms such as Bees Algorithm. Based on Ahmed and Glasgow's work (2012), SI's general advantages include scalability, adaptability, and collective robustness. SI systems' control mechanisms are not too dependent on swarm size, as long as it is not too small. SI systems answer well to rapidly varying environments. The fault-tolerance ability of SI systems is remarkably high, as these systems have no single point of failure. This single point of failure is a part of any system that positions the whole system into risk of a complete failure, if it ceased to function.

##### 2.1.4.1 Related Works

###### 2.1.4.1.1 A Fuzzy-Based Particle Swarm Optimization Algorithm for Nurse Scheduling (2014)

PSO, a Swarm Intelligence approach, is developed by Dr. Eberhart and Dr. Kennedy in 1995 and inspired by the social behavior of the groups of population in nature like animal herds, flocking birds, and a school of fish. This approach consists of a population called swarm and each member of the swarm is called a particle.

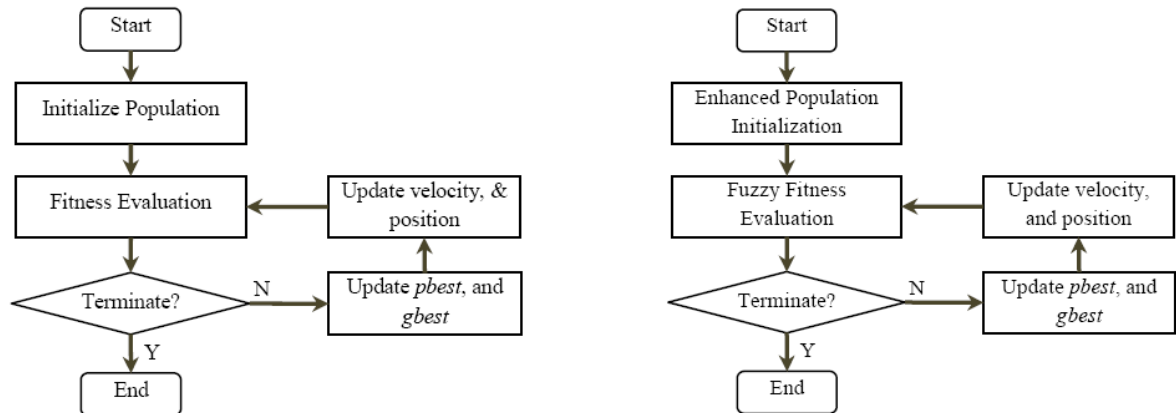


Figure 2.1: Flowcharts for basic PS and proposed PSO-based algorithm structure

Mutingi and Mbohwa (2014) presented the PSO metaheuristic technique, and utilized the fuzzy set theory concepts for considering nurse personal preferences to produce near-optimal solutions. It incorporated hard constraints to improve the computational efficiency of the algorithm. The study was able to enhance initialization based on hard constraints and apply fuzzy set theory on fitness evaluation of soft constraints. An enhanced solution generation heuristic is developed for better efficiency. The researchers believed the proposed algorithm forms a platform for further of decision support system for decision makers in the application area. The experimental results indicated that the proposed algorithm is effective and efficient, even over large scale problems, in the range of 83%.

#### 2.1.4.1.2 Multi-Objective Ant Colony Optimization Algorithm to Solve a Nurse Scheduling Problem (2013)

ACO, a Swarm Intelligence approach, was developed by Gambardella Dorigo in 1997 and is inspired by the pheromone-based strategy of ants, foraging in nature. The foraging behavior of ants is based on searching the shortest path between source and their nests.

Category	MOACO				GA			
	Initial Solution	Final Solution	% Im	CPU	Initial Solution	Final Solution	% Im	CPU
	Vmoy	Vmoy			Vmoy	Vmoy		
C1	0.25	0.011	96	48.7	1.53	0.07	96	49.81
C2	1.41	0.33	77	55.7	3.34	0.39	88	71.71
C3	1.68	0.31	82	60.07	3.31	0.32	90	87.26
C4	1.9	0.27	86	54.2	3.04	0.27	92	85.60

<b>C5</b>	2.43	0.33	86	<b>66.13</b>	3.17	0.34	<b>90</b>	98.80
<b>C6</b>	2.54	0.84	45	<b>67</b>	5.48	0.93	<b>83</b>	139.25

Table 2.1: Results of MOACO and GA (Saji et al., 2013)

Saji et al. (2013) proposed an adaptation of a multi-objective mathematical programming model and ACO algorithm. They used real data from two (2) units: Intensive Care Unit (ICU) and Emergency Unit, and further defined six (6) categories of test problems. Each category resembles to a particular shift (Day, Evening, or Night). To identify effectiveness of the proposed search method, it was compared with GA, and tested on limited population from different units of a large hospital. In Table 2.1, MOACO was able to generate a feasible schedule, in terms of execution time (CPU), however GA provided a significant effectiveness with remarkable percentage improvement from the initial solution (%Im). As a result, the researchers recommended to use other variants to provide good quality schedule compared to genetic algorithm with a large population and within a reasonable time.

## 2.2 Proposed Approach: Bees Algorithm – Swarm Intelligence (Metaheuristic)

### 2.2.1 Biological Background – The Foraging Behavior of Honey Bees and The Waggle

#### Dance

Bees Algorithm (BA) is a metaheuristic Swarm Intelligence approach that belongs to the class of nature-inspired algorithms. It is developed by Pham in 2005 and inspired by a number of biological and natural processes. This search technique utilizes a resemblance between the way in which bees in nature search for food, and the way in which optimization algorithms search for an optimal solution in difficult combinatorial optimization problems.

The BA mimics the food foraging behavior of colony of honey bees. The foraging process starts when the bee colony has send scout bees to collect nectar from flower patches relative to the amount of food available at each patch. Moreover, the bees communicate with each other at the hive through a waggle dance that informs other bees in the hive as to the direction in which it will be found, distance from the hive, and quality rating or fitness of food sources. This waggle dance is quite essential for colony communication. The said information supports the colony to send its scout bees to flower patches accurately, without using guides or maps. In addition, more follower bees are sent to more promising patches. With this communication, it allows the colony to gather food quickly and efficiently.

### 2.2.2 Applications

BA has been successfully applied to various engineering and management optimization problems. Moreover, it has been also applied in some problems like multi-objective optimization (Pham & Ghanbarzadeh, 2007), neural network training for wood defects (Pham et al., 2006), manufacturing cell formation (Pham et al., 2007), job shop scheduling for a machine (Pham et al., 2007), data clustering (Pham et al., 2007), optimizing the design of mechanical components (Pham et al., 2007), image analysis (Yuce, 2012), and supply chain network optimization (Mastrocinque et al., 2013).

### 2.2.3 Related Works

#### 2.2.3.1 The Bees Algorithm – A Novel Tool for Complex Optimisation Problems (2006)

Pham et al. (2006) work provided results obtained for a number of benchmark problems signifying the efficiency and robustness of BA while being compared to two (2) metaheuristics, GA and ACO.

Benchmark Function	GA		ACO		BA	
	Mean No. of Evaluations (Number of Iterations)	Success %	Mean No. of Evaluations (Number of Iterations)	Success %	Mean No. of Evaluations (Number of Iterations)	Success %
<b>De Jong</b>	10160	100	6000	100	<b>868</b>	<b>100</b>
<b>Goldstein &amp; Price</b>	5662	100	5330	100	<b>999</b>	<b>100</b>
<b>Branin</b>	7325	100	1936	100	<b>1657</b>	<b>100</b>
<b>Martin &amp; Gaddy</b>	2844	100	1688	100	<b>526</b>	<b>100</b>
<b>Rosenbrock</b>	10212	100	6842	100	<b>631</b>	<b>100</b>
<b>Hyper Sphere</b>	15468	100	22050	100	<b>7113</b>	<b>100</b>
<b>Greiwangk</b>	200000	100	50000	100	<b>1847</b>	<b>100</b>

Table 2.2: Test Functions: GA, ACO and BA (Pham et al, 2006)

Table 2.2 results were quite promising since BA has revealed significant robustness, generating a 100% success percentage in totally. Based on De Jong's test function, BA was 120 times faster than ACO and 207 times faster than GA. With Goldstein & Price test function, BA was 5 times faster than ACO and GA. Through Branin's test function, a 15% BA improvement was noted when compared to ACO and 77% when compared to GA. In Martin & Gaddy & Rosenbrock's test functions, BA brought 100% success and significant improvement over GA and ACO. By means of Hyper Sphere test function, BA required half of the mean number of evaluations compared to GA and one-third of that required for ACO. Lastly, in Greiwangk test function, BA was found 10 times faster than GA and 25 times faster than ACO. In other words, BA outperformed ACO and GA in terms of speed of optimization and accuracy of the results obtained and could be considered a tool for complex optimization problems.

## 2.2.3.2 Forming Student Groups Using Bees Algorithm (2012)

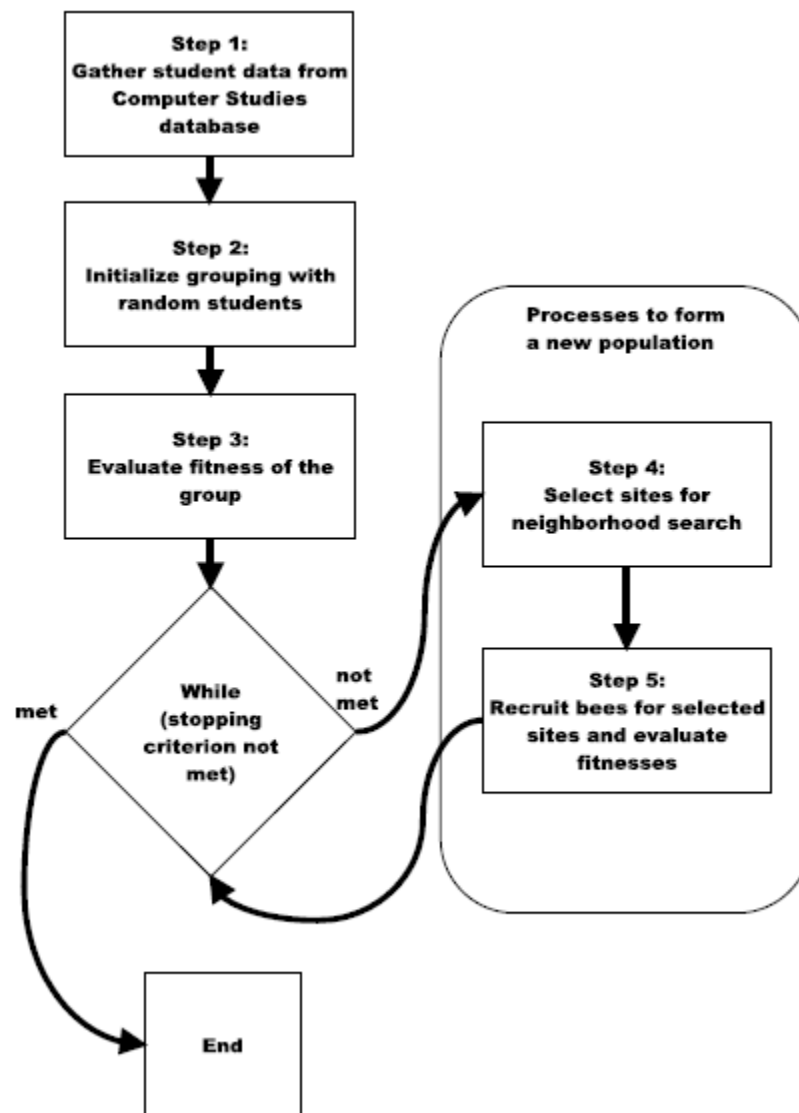


Figure 2.2: The Conceptual Framework (Ablazo et al., 2012)

Ablazo et al. (2012) utilized BA and objective function to form 3-student group where the students' academic grades on major subjects were the only basis. The results delivered a good measure of heterogeneity and fitness and all objectives of the study were considered accomplished; however the proponents believed that grades are not the sole basis for forming groups, as they have identified through their data gathering. They have recommendations such as finding a new mathematical approach to cater a 2-student group formation, and adding some other criteria to consider like the minor subjects, work ethic, personal preferences, etc.

### 2.2.3.3 Using The Bees Algorithm To Schedule For A Machine (2007)

Pham et al. (2007) presented a study concerning scheduling of jobs with a common due date for a machine to minimize the penalties related with early or late completion. BA was also compared with those of existing optimization techniques such as DPSO (Discrete PSO), TS and GA. Since BA is considered stochastic (randomly determined, not precisely predicted), its minimum, maximum, average and standard deviation of runs should be given to evaluate its performance.

$\Delta min$						
$n$	$h 0.2$					
	DPSO	TS	GA	HTG	HGT	BA
<b>10</b>	0.00	0.25	0.12	0.12	0.12	<b>0.00</b>
<b>20</b>	-3.84	-3.84	-3.84	-3.84	-3.84	<b>-3.84</b>
<b>50</b>	-5.70	-5.70	-5.68	-5.70	-5.70	<b>-5.70</b>
<b>100</b>	-6.19	-6.19	-6.17	-6.19	-6.19	<b>-6.19</b>
<b>200</b>	-5.78	-5.76	-5.74	-5.76	-5.76	<b>-5.78</b>
<b>500</b>	-6.42	-6.41	-6.41	-6.41	-6.41	<b>-6.43</b>
<b>1,000</b>	-6.76	-6.73	-6.75	-6.74	-6.74	<b>-6.76</b>
<b>Avg.</b>	-4.96	-4.91	-4.92	-4.93	-4.93	<b>-4.96</b>
$n$	$h 0.8$					
	DPSO	TS	GA	HTG	HGT	BA
<b>10</b>	0.00	0.00	0.00	0.00	0.00	<b>0.00</b>
<b>20</b>	-0.41	-0.41	-0.28	-0.41	-0.41	<b>-0.41</b>
<b>50</b>	-0.24	-0.24	-0.19	-0.23	-0.23	<b>-0.24</b>
<b>100</b>	-0.18	-0.15	-0.12	-0.08	-0.11	<b>-0.18</b>
<b>200</b>	-0.15	-0.04	-0.14	0.26	0.07	<b>-0.15</b>
<b>500</b>	-0.11	0.21	-0.11	0.73	0.13	<b>-0.11</b>
<b>1,000</b>	-0.06	1.13	-0.05	1.28	0.40	<b>-0.05</b>
<b>Avg.</b>	-0.16	0.07	-0.13	0.22	-0.02	<b>-0.16</b>

Table 2.3: Minimum Deviation Computational Results

In Table 2.3, the minimum deviation of 6 metaheuristics, including BA, DPSO (Discrete PSO), TS, and GA, were computed. It showed that BA, PSO and GA have a similar tendency to yield negative percentage relative deviation, which means they outperform Biskup and Feldmann benchmark. The other metaheuristics, however, are considered inferior.

$\Delta_{max}$		
$h$	DPSO	BA
0.2	-4.90	-4.95
0.4	-3.18	-3.26
0.6	-0.03	-0.22
0.8	-0.16	-0.16
Avg.	-2.07	-2.15

Table 2.4: Maximum Deviation Computation Results

In Table 2.4, the maximum deviation of 2 metaheuristics, BA and DPSO were also computed. BA exhibited a significant result than DPSO, including the total average.

$n$	$h$	$\Delta_{min}$		$\Delta_{max}$		$\Delta_{avg}$			$\Delta_{std}$	
		DPSO	Bees	DPSO	Bees	DPSO	Bees	DE	DPSO	Bees
10	0.2	0.00	0.00	0.11	0.00	0.01	0.00	0.00	0.03	0.00
	0.4	0.00	0.00	0.15	0.00	0.02	0.00	0.00	0.05	0.00
	0.6	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.00	0.00
	0.8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
20	0.2	-3.84	-3.84	-3.79	-3.83	-3.83	-3.84	-3.84	0.02	0.00
	0.4	-1.63	-1.63	-1.57	-1.63	-1.62	-1.63	-1.63	0.02	0.00
	0.6	-0.72	-0.72	-0.66	-0.72	-0.71	-0.72	-0.72	0.03	0.00
	0.8	-0.41	-0.41	-0.41	-0.41	-0.41	-0.41	-0.41	0.00	0.00
50	0.2	-5.70	-5.70	-5.61	-5.69	-5.68	-5.70	-5.69	0.03	0.00
	0.4	-4.66	-4.66	-4.52	-4.66	-4.63	-4.66	-4.66	0.05	0.00
	0.6	-0.34	-0.34	-0.23	-0.34	-0.31	-0.34	-0.32	0.04	0.00
	0.8	-0.24	-0.24	-0.24	-0.22	-0.24	-0.24	-0.24	0.00	0.01
100	0.2	-6.19	-6.19	-6.15	-6.19	-6.18	-6.19	-6.17	0.02	0.00
	0.4	-4.94	-4.94	-4.82	-4.93	-4.90	-4.94	-4.89	0.04	0.00
	0.6	-0.15	-0.15	0.26	-0.14	-0.09	-0.14	-0.13	0.14	0.00
	0.8	-0.18	-0.18	-0.18	-0.17	-0.18	-0.18	-0.17	0.00	0.00
200	0.2	-5.78	-5.78	-5.74	-5.77	-5.77	-5.78	-5.77	0.01	0.00
	0.4	-3.75	-3.75	-3.68	-3.74	-3.72	-3.75	-3.72	0.02	0.01
	0.6	-0.15	-0.15	0.56	-0.15	-0.03	-0.15	0.23	0.27	0.00
	0.8	-0.15	-0.15	-0.15	-0.15	-0.15	-0.15	0.20	0.00	0.00
500	0.2	-6.42	-6.43	-6.40	-6.42	-6.41	-6.43	-6.43	0.01	0.00
	0.4	-3.56	-3.57	-3.51	-3.56	-3.54	-3.57	-3.57	0.01	0.00
	0.6	-0.11	-0.11	-0.11	-0.11	-0.11	-0.11	1.72	0.00	0.00
	0.8	-0.11	-0.11	-0.11	-0.11	-0.11	-0.11	1.01	0.00	0.00
1000	0.2	-6.76	-6.76	-6.73	-6.74	-6.75	-6.75	-6.72	0.01	0.01
	0.4	-4.37	-4.35	-4.32	-4.33	-4.35	-4.34	-4.38	0.01	0.01
	0.6	-0.06	-0.05	-0.03	-0.05	-0.04	-0.05	1.29	0.01	0.00
	0.8	-0.06	-0.05	-0.06	-0.05	-0.06	-0.05	2.79	0.00	0.00
Avg.		-2.15	-2.15	-2.07	-2.15	-2.14	-2.15	-1.87	0.03	0.00

Table 2.5: Comparison Between BA, DPSO and DE

In Table 2.5, the results between BA and DPSO's minimum, maximum, and average percentage of relative deviations and standard deviation. The average percentage of relative deviation of BA was compared to DPSO and Differential Evolution (DE). The results showed that BA again outperformed DPSO and DE. The standard deviation for BA is nearly zero, which means BA is slightly more robust than DPSO. All the statistics found show that the performance of BA is superior to all existing approaches.

### 2.3 Theoretical Frameworks

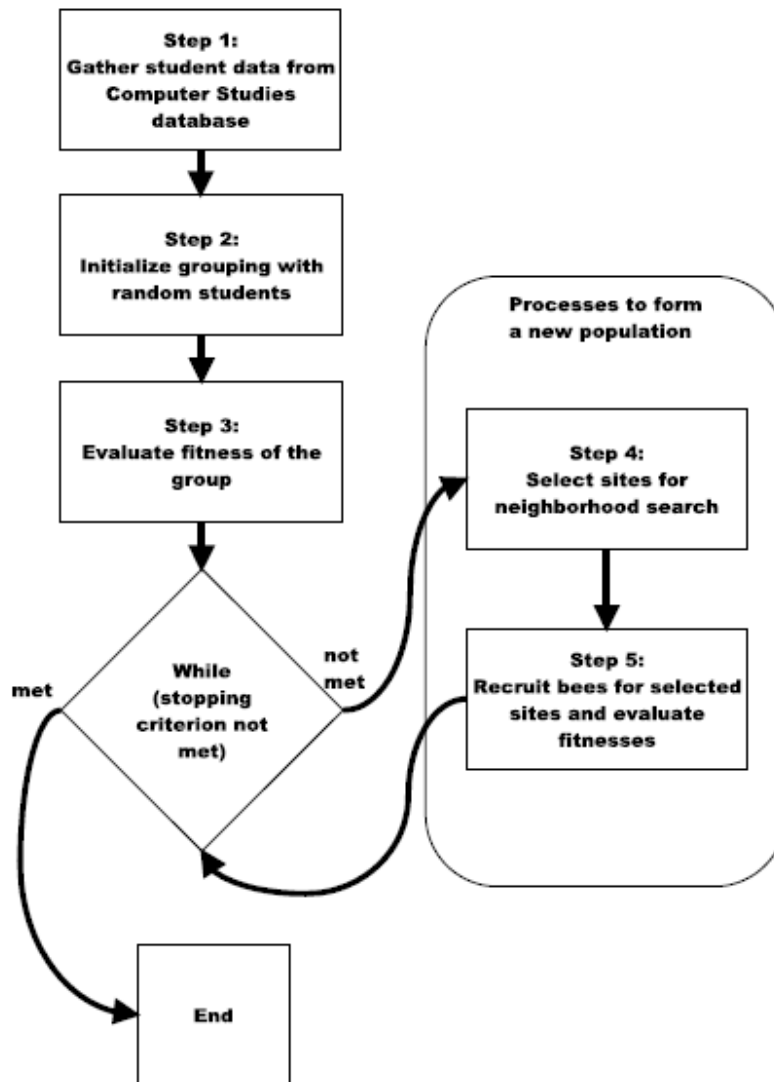


Figure 2.1: Theoretical Framework (BA Group Formation – Ablazo et al., 2012)

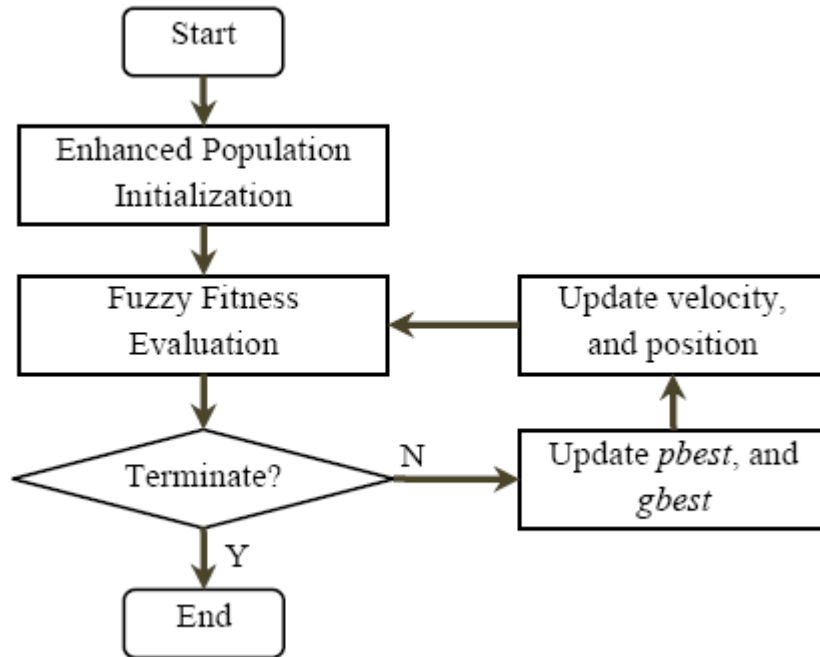


Figure 2.2: A Fuzzy-Based Particle Swarm Optimization Algorithm for Nurse Scheduling  
(Mutingi and Mhobwa, 2014)

### 3. RESEARCH DESIGN AND METHODOLOGY

#### 3.1 Conceptual Framework

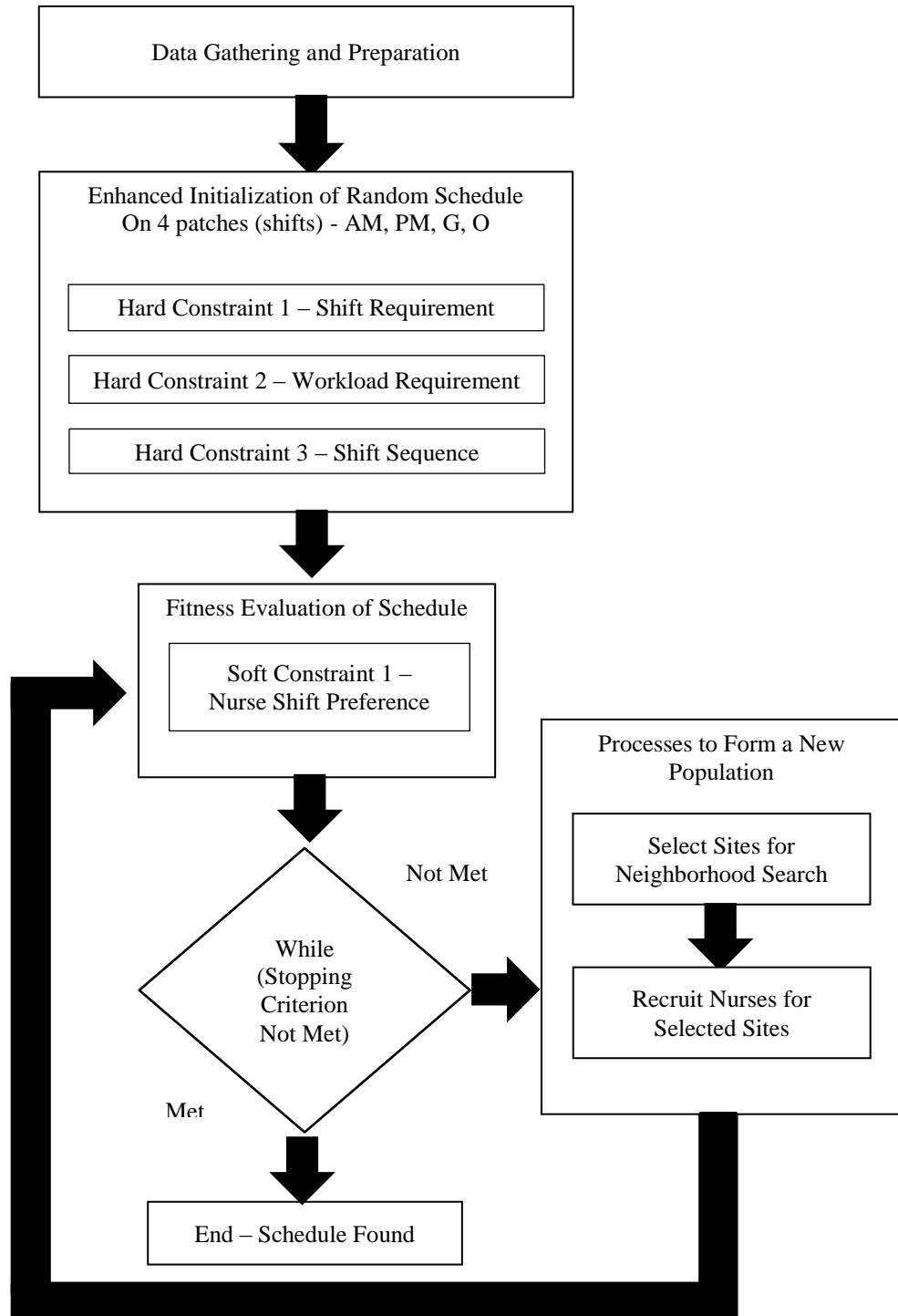


Figure 3.1: Conceptual Framework

### 3.1.1 Data Gathering and Preparation

The first phase is gathering nurse data, such as nurse name, their preferences and specified hospital unit from a department.

### 3.1.2 Enhanced Initialization of Random Schedule on Four (4) Patches

The proposed initialization is considered to be enhanced since this seeks to satisfy all given hard constraints by the head nurse. This procedure also followed by random assignments of the four (4) shifts (patches). Nurses (bees) are randomly placed in the search space with a schedule.

### 3.1.3 Fitness Evaluation of Schedule

The fitness values of the sites visited by the scout bees (nurses) are evaluated based on soft constraint. Through this, the quality of the schedule is improved. The fitness function offers the decision maker an opportunity to incorporate choices reflecting the preferences of the management and the nurses.

### 3.1.4 Select Sites for Neighborhood Search

Bees (nurses) that have highest fitness values are chosen as “selected” bees and sites visited by them are chosen for neighborhood search.

### 3.1.5 Recruit Bees for Selected Sites

Searches are being conducted in the neighborhood of the selected sites, assigning more bees (nurses) to search near to the best e sites. The bees (nurses) could be straightly selected which is permitted by the fitness evaluation with the sites they are visiting. On the other hand, the evaluated fitness values are used to determine the probability of the bees (nurses) being selected.

Searches in the neighborhood of the best e sites which signify more favorable solutions are made more detailed by recruiting more bees (nurses) to follow them than the other selected bees (nurses).

### 3.1.6 End - Schedule Found

When the constraints and criteria are met, the search is already stopped and an optimal solution is obtained.

## 3.2 Methodology

### 3.2.1 Data Gathering and Preparation

The proponents will gather nurse data, such as nurse name, their preferences and specified hospital unit from a department.

### 3.2.2 Develop Criteria

The proponents will develop a criterion for efficient and effective schedule. These include a fitness function and will be utilized.

#### 3.2.2.1 Fitness Function

Fitness Function (FF) is an indicator on how much a program satisfies the soft constraint, nurses' preferences.

### 3.2.3 Implement BA Using an Automated Nurse Scheduling Program

The proponents will produce a front-end interface that would include BA and the developed criteria.

### 3.2.4 Testing and Evaluation

The automated program is tested and evaluated by the ward and its staff nurses.

## 4. THEORETICAL BACKGROUND

### 4.1 Bees Algorithm

#### 4.1.1 Bees in Nature

The foraging process starts when the bee colony has send scout bees to collect nectar from flower patches relative to the amount of food available at each patch. Moreover, the bees communicate with each other at the hive through a waggle dance that informs other bees in the hive as to the direction in which it will be found, distance from the hive, and quality rating or fitness of food sources. This waggle dance is quite essential for colony communication. The said information supports the colony to send its scout bees to flower patches accurately, without using guides or maps. In addition, more follower bees are sent to more promising patches. With this communication, it allows the colony to gather food quickly and efficiently.

#### 4.1.1 Bees Algorithm in Nurse Scheduling

In general, the objective on solving the nurse scheduling problem is to search for a schedule that satisfies a given set of hard constraints. However, in practice, the preferences of nurses must be also satisfied to the highest degree possible. With violation of hard constraints establishes an infeasible schedule, however, violation of soft constraints is acceptable to some extent, but at the expense of the schedule quality. Soft constraints, are consequently, added to improve the quality of the schedule. The core challenge of this study is to incorporate these hard constraints into the scheduling process, so as to improve the quality of the schedule.

Pham (2007), the one who proposed BA, discussed that BA algorithm has search purposes. Moreover, BA is quite different from other search processes. Some processes utilize probabilistic approach during the neighborhood stage; however BA does not use any probability approach, instead it utilizes fitness evaluation to drive the search.

In this study, the “bees” variables are the nurses. Moreover, the “scout bees” are the ones with the highest fitness of the individual preferences shift scores, meaning these are the nurses who are considered to be “all-around”. These scout bees are the ones who acquired a very high fitness value based from the shift preferences scores. In addition, the AM, PM, Graveyard, and Off shifts are considered to be 4 patches. The patch size depends on the number of nurses per shift type.

## 5. RESULTS AND DISCUSSION

### 5.1 Data Gathering and Preparation

#### 5.1.1 4-Week Schedule Data

This schedule data was gathered from a station head nurse. The station currently has 12 contractual nurses with four (4) shifts: am, pm, g, and o. The current station has three (3) staff nurses for each shift type.

Name	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
n1	g	o	am	o	o	am	am
n2	g	o	am	am	am	g	g
n3	am	am	pm	g	g	o	o
n4	am	am	pm	pm	pm	o	o
n5	o	pm	pm	pm	g	g	o
n6	o	pm	g	g	o	am	pm
n7	am	pm	pm	pm	g	o	am
n8	o	am	am	pm	pm	g	g
n9	g	g	o	am	am	o	o
n10	pm	pm	g	g	o	pm	pm
n11	pm	g	o	am	am	pm	pm
n12	o	o	am	am	pm	pm	g

Figure 5.1: Week 1 Schedule

Name	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
n1	pm	g	g	o	am	pm	o
n2	o	am	am	am	pm	pm	pm
n3	am	am	am	o	o	o	am
n4	am	pm	pm	g	g	o	o
n5	o	pm	pm	pm	g	g	o
n6	pm	pm	g	g	o	am	pm
n7	pm	g	g	o	am	am	g
n8	o	o	o	pm	g	g	o
n9	g	g	o	g	pm	g	g
n10	g	o	o	am	am	am	pm
n11	g	o	pm	pm	o	o	am
n12	o	o	o	am	pm	pm	g

Figure 5.2: Week 2 Schedule

Name	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
n1	pm	g	g	o	pm	pm	g
n2	o	am	g	o	o	am	pm
n3	am	pm	pm	g	o	am	am
n4	pm	g	o	o	am	g	pm
n5	g	am	am	am	o	o	am
n6	pm	pm	o	am	g	g	o
n7	g	o	o	am	am	pm	pm
n8	o	am	am	pm	pm	pm	g
n9	o	pm	pm	pm	pm	g	o
n10	g	o	am	pm	g	g	o
n11	am	pm	g	g	o	o	am
n12	g	o	pm	g	g	o	pm

Figure 5.3: Week 3 Schedule

Name	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
n1	g	o	o	am	am	pm	pm
n2	pm	g	o	o	am	am	g
n3	am	pm	pm	o	pm	pm	g
n4	g	g	o	o	am	am	pm
n5	am	pm	pm	g	g	o	o
n6	am	am	pm	pm	g	o	am
n7	g	o	pm	o	o	o	am
n8	o	am	am	am	pm	g	g
n9	o	am	am	pm	pm	g	o
n10	pm	pm	pm	pm	g	g	o
n11	pm	pm	g	g	o	o	am
n12	pm	pm	g	g	o	am	pm

Figure 5.4: Week 4 Schedule

When the proponents analyzed the given data, they found out that a number of hard constraint violations were noticed, resulting to overstaffing, understaffing, and 3-offs (less than 40 hours workload).

Name	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	<i>am</i>	<i>pm</i>	<i>g</i>	<i>o</i>
n1	g	o	am	o	o	am	am	3	0	1	3
n2	g	o	am	am	am	g	g	3	0	3	1
n3	am	am	pm	g	g	o	o	2	1	2	2
n4	am	am	pm	pm	pm	o	o	2	3	0	2
n5	o	pm	pm	pm	g	g	o	0	3	2	2
n6	o	pm	g	g	o	am	pm	1	2	2	2
n7	am	pm	pm	pm	g	o	am	2	3	1	1
n8	o	am	am	pm	pm	g	g	1	2	2	1
n9	g	g	o	am	am	o	o	2	0	2	3
n10	pm	pm	g	g	o	pm	pm	0	4	2	1
n11	pm	g	o	am	am	pm	pm	2	3	1	1
n12	o	o	am	am	pm	pm	g	2	2	1	2
<i>am</i>	3	3	4	4	3	2	2				
<i>pm</i>	2	4	4	4	3	3	3				
<i>g</i>	3	2	2	3	3	3	3				
<i>o</i>	4	3	2	1	3	4	4				

Figure 5.5: Week 1 with Hard Constraint Violations

Name	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	<i>am</i>	<i>pm</i>	<i>g</i>	<i>o</i>
n1	pm	g	g	o	am	pm	o	1	2	2	2
n2	o	am	am	am	pm	pm	pm	3	3	0	1
n3	am	am	am	o	o	o	am	4	0	0	3
n4	am	pm	pm	g	g	o	o	1	2	2	2
n5	o	pm	pm	pm	g	g	o	0	3	2	2
n6	pm	pm	g	g	o	am	pm	1	3	2	1
n7	pm	g	g	o	am	am	g	2	1	3	1
n8	o	o	o	pm	g	g	o	0	1	2	4
n9	g	g	o	g	pm	g	g	0	1	5	1
n10	g	o	o	am	am	am	pm	3	1	1	2
n11	g	o	pm	pm	o	o	am	1	2	1	3
n12	o	o	o	am	pm	pm	g	1	2	1	3
<i>am</i>	2	2	2	3	3	3	2				
<i>pm</i>	3	3	3	3	3	3	3				
<i>g</i>	3	3	3	3	3	3	3				
<i>o</i>	4	4	4	3	3	3	4				

Figure 5.6: Week 2 with Hard Constraints Violations

Name	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	<i>am</i>	<i>pm</i>	<i>g</i>	<i>o</i>
n1	pm	g	g	o	pm	pm	g	0	3	3	1
n2	o	am	g	o	o	am	pm	2	1	1	3
n3	am	pm	pm	g	o	am	am	3	2	1	1
n4	pm	g	o	o	am	g	pm	1	2	2	2
n5	g	am	am	am	o	o	am	4	0	1	2
n6	pm	pm	o	am	g	g	o	1	2	2	2
n7	g	o	o	am	am	pm	pm	2	2	1	2
n8	o	am	am	pm	pm	pm	g	2	3	1	1
n9	o	pm	pm	pm	pm	g	o	0	4	1	2
n10	g	o	am	pm	g	g	o	1	1	3	2
n11	am	pm	g	g	o	o	am	2	1	2	2
n12	g	o	pm	g	g	o	pm	0	2	3	2
<i>am</i>	2	3	3	3	2	2	3				
<i>pm</i>	3	4	3	3	3	3	4				
<i>g</i>	4	2	3	3	3	4	2				
<i>o</i>	3	3	3	3	4	3	3				

Figure 5.7: Week 3 with Hard Constraints Violations

Name	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	<i>am</i>	<i>pm</i>	<i>g</i>	<i>o</i>
n1	g	o	o	am	am	pm	pm	2	2	1	2
n2	pm	g	o	o	am	am	g	2	1	2	2
n3	am	pm	pm	o	pm	pm	g	1	2	3	1
n4	g	g	o	o	am	am	pm	2	1	2	2
n5	am	pm	pm	g	g	o	o	1	2	2	2
n6	am	am	pm	pm	g	o	am	3	2	1	1
n7	g	o	pm	o	o	o	am	1	1	1	4
n8	o	am	am	am	pm	g	g	3	1	2	1
n9	o	am	am	pm	pm	g	o	2	2	1	2
n10	pm	pm	pm	pm	g	g	o	0	4	2	1
n11	pm	pm	g	g	o	o	am	1	2	2	2
n12	pm	pm	g	g	o	am	pm	0	2	3	2
<i>am</i>	3	3	2	2	3	3	3				
<i>pm</i>	4	5	5	3	3	2	3				
<i>g</i>	3	2	2	3	3	3	3				
<i>o</i>	2	2	3	4	3	4	3				

Figure 5.8: Week 4 with Hard Constraints Violations

## 5.1.2 Interview with Head Nurse

<i>No.</i>	<i>Question</i>	<i>Summary</i>
1	How do you make schedules? Is it by hand?	We manually make schedules. We cannot find any available scheduler system that could fit the needs.
2	Do you consider any pattern/rules in making schedules? What else do you consider while making them? Do you consider their specific roles like being the “charge nurse” and “medicating nurse”?	<p>Yes, only make sure that the total hour work of each nurse per week is 40 hours. So it is 5 working days, 2 off days. And no graveyard-am pattern. Actually it really depends on the hospital management, and the head nurse herself/himself. In this station, every am, pm and graveyard shifts have 3 staff nurses each. All other stations might have different combinations.</p> <p>No, we do not consider those roles while making the schedule. We only consider the skill levels which are the head nurses and the staff nurses only. However some hospital considers the level of qualification. It really depends on the hospital management.</p>
3	How long are schedules made?	It takes hours making schedules by hand. It is hard and time-consuming. Sometimes we recycle previous schedules so that it would be more convenient for us.
4	What problems have you encountered after making schedules?	The most common problem is understaffing. If understaffing occurs, we usually take over the empty slot. So every duty we have 2 roles, we work as a head nurse, as well as a regular staff nurse, which is too much load.
5	Do you consider nurse’s preferences?	No, since it is hard and it does more workload.
6	If there is a way to consider the individual nurse shift preferences, do you think the scheduling will improve?	Of course yes, everyone will be satisfied with their schedule.
7	Does the rotation of the stations do affect the nurse scheduling?	No, because rotating nurses is on a different scope, it is not part of the nurse scheduling.

### 5.1.3 Preferences Shift Score Survey

ID	Name	AM Score	PM Score	Graveyard Score
39	n1	1	2	3
40	n2	3	2	1
41	n3	3	1	2
42	n4	1	3	2
43	n5	1	2	3
44	n6	1	2	3
45	n7	3	2	1
46	n8	3	1	2
47	n9	1	2	3
48	n10	2	3	1
49	n11	1	3	2
50	n12	2	3	1

Figure 5.5: Nurse Preferences Shift Scores Results

In this survey, when a nurse provides a shift score of 3 for a specific shift, he/she has the greatest preference for that shift type, which means the nurse fits 100% for that shift. A shift score of 2 means the nurse fits 50% for that shift, meaning that he/she has a neutral stand for that shift type. A score of 1 is the least preferred shift. The scores of 1, 2, and 3 were used as the nurse preferences shift scores because the computation for the fitness evaluation using these values is consequently direct.

## 5.2 Developing Criteria

In this study, the proponents have developed criteria with four (4) components based from the interview. All these components mainly focused on the hard and soft constraints.

### 5.2.1 Criterion 1 – Shift Requirement (Hard Constraint)

To achieve a perfect criterion, all shift counts must meet the station's required number of nurses per shift type. In this study, since there are seven (7) days and four (4) shifts, the total number of shifts per week is 28. If there is a violation of understaffing or overstaffing, there is 1-point deduction.

$$\text{criterion 1 percentage} = ((x-y)/ x) * 100\%$$

$$x = \text{total number of shifts per week}$$

$$y = \text{total number of understaff and overstaff violation count}$$

Name	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	am	pm	g	o
n1	g	o	o	am	am	pm	pm	2	2	1	2
n2	pm	g	o	o	am	am	g	2	1	2	2
n3	am	pm	pm	o	pm	pm	g	1	2	3	1
n4	g	g	o	o	am	am	pm	2	1	2	2
n5	am	pm	pm	g	g	o	o	1	2	2	2
n6	am	am	pm	pm	g	o	am	3	2	1	1
n7	g	o	pm	o	o	o	am	1	1	1	4
n8	o	am	am	am	pm	g	g	3	1	2	1
n9	o	am	am	pm	pm	g	o	2	2	1	2
n10	pm	pm	pm	pm	g	g	o	0	4	2	1
n11	pm	pm	g	g	o	o	am	1	2	2	2
n12	pm	pm	g	g	o	am	pm	0	2	3	2
am	3	3	2	2	3	3	3				
pm	4	5	5	3	3	2	3				
g	3	2	2	3	3	3	3				
o	2	2	3	4	3	4	3				

Figure 5.8: Week 4 with Hard Constraints Violations

One example (Figure 5.8) of a schedule made by hand is that there are seven (7) counts of understaffing, and five (5) counts of overstaffing (found on the last four rows), resulting to 16 out of 28. Multiplying the result by 100 gets the percentage. Sample computation:  $(16 / 28) * 100\% = 57.14\%$ .

### 5.2.2 Criterion 2 – Workload Requirement (Hard Constraint)

To achieve a perfect criterion, all shift counts must meet the station's required number of hours of workload which is 40 hours, or more. In this study, each nurse has a limit of 2-offs. When a violation occurs, such as having 3-off count, there is 1-point deduction.

$$\text{criterion 2 percentage} = ((x-y) / x) * 100\%$$

$$x = \text{total number of staff nurses}$$

$$y = \text{total number of nurses with workload violation count}$$

Name	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	am	pm	g	o
n1	pm	g	g	o	am	pm	o	1	2	2	2
n2	o	am	am	am	pm	pm	pm	3	3	0	1
n3	am	am	am	o	o	o	am	4	0	0	3
n4	am	pm	pm	g	g	o	o	1	2	2	2
n5	o	pm	pm	pm	g	g	o	0	3	2	2
n6	pm	pm	g	g	o	am	pm	1	3	2	1
n7	pm	g	g	o	am	am	g	2	1	3	1
n8	o	o	o	pm	g	g	o	0	1	2	4
n9	g	g	o	g	pm	g	g	0	1	5	1
n10	g	o	o	am	am	am	pm	3	1	1	2
n11	g	o	pm	pm	o	o	am	1	2	1	3
n12	o	o	o	am	pm	pm	g	1	2	1	3
am	2	2	2	3	3	3	2				
pm	3	3	3	3	3	3	3				
g	3	3	3	3	3	3	3				
o	4	4	4	3	3	3	4				

Figure 5.6: Week 2 with Hard Constraints Violations

One example (Figure 5.6) of a schedule made by hand is that there are four (4) counts of workload violation (found on the last column), resulting to 8 out 12 nurses have not violated. Multiplying the result by 100 gets the percentage. Sample computation:  $(8 / 12) * 100\% = 66.67\%$ .

### 5.2.3 Criterion 3 – Shift Sequence (Hard Constraint)

To achieve a perfect criterion, all shift counts must meet the right shift sequence. If there is an AM-PM, PM-G, G-AM combination, a violation occurs. A 1-point deduction occurs when a violation was committed.

$$\text{criterion 3 percentage} = ((x-y) / x) * 100\%$$

$$x = \text{total number of staff nurses}$$

$$y = \text{total number of nurses with shift sequence violation count}$$

Name	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
n1	pm	g	g	o	pm	pm	g
n2	o	am	g	o	o	am	pm
n3	am	pm	pm	g	o	am	am
n4	pm	g	o	o	am	g	pm
n5	g	am	am	am	o	o	am
n6	pm	pm	o	am	g	g	o
n7	g	o	o	am	am	pm	pm
n8	o	am	am	pm	pm	pm	g
n9	o	pm	pm	pm	pm	g	o
n10	g	o	am	pm	g	g	o
n11	am	pm	g	g	o	o	am
n12	g	o	pm	g	g	o	pm

Figure 5.7: Week 3 with Hard Constraints Violations

One example (Figure 5.7) of a schedule made by hand is that there is one (1) count of violation, resulting to 11 out of 12 nurses. After getting the result, multiply it by 100 to get the percentage. Sample computation:  $(11 / 12) * 100\% = 91.67\%$ .

### 5.2.4 Criterion 4 – Fitness Function (Soft Constraint)

The Fitness Function is an indicator on how much a program satisfies the soft constraint, nurses' preferences. In this study, the proponents designed the function based on the nurses' shift scores. In every patch, this is evaluated by given schedule over total highest fitness that each patch could acquire.

$$\text{criterion 4 percentage} = ((x / y) * 100\%)$$

$$x = \text{total preference shift type score}$$

$$y = \text{total highest preference shift type score}$$

*overall criterion 4 percentage = (a + b + c) / 3*

*a = criterion 4 percentage for AM Shift*  
*b = criterion 4 percentage for PM Shift*  
*c = criterion 4 percentage for Graveyard Shift*

**STEP 1: RANDOM INITIALIZATION**

AM Shift: n5 n6 n4  
 PM Shift: n2 n12 n3  
 Graveyard Shift: n7 n1 n11  
 Off: n11 n10 n8

**STEP 2: FITNESS EVALUATION**

AM Shift: 3/9 66.666666666667%  
 PM Shift: 6/9 83.333333333333%  
 Graveyard Shift: 6/9 83.333333333333%

**OVERALL: 77.777777777778%**

Figure 5.6: Sample Fitness Function Computation After Initialization Run

For example, in the AM shift, n5 has a one (1) am preference shift score, n6 has a one (1) am preference shift score, and n4 has a one (1) am shift score. These three (3) scores are accumulated and divided by total highest score the shift scores could get which in this case is 9 since there are 3 nurses for the AM shift. After that, the result is multiplied by 100 to get the fitness evaluation for that shift. Repeat the process for the PM and Graveyard Shifts.

Sample computation:

*AM Shift: (1+1+1) / 9 \* 100% = 66.67%*  
*PM Shift: (2+3+1) / 9 \* 100% = 83.33%*  
*Graveyard Shift: (1+3+2) / 9 \* 100% = 83.33%*

*Overall Criterion 4: (66.67% + 83.33% + 83.33%) / 3 = 77.77%*

The fitness function only considers the AM, PM, and Graveyard shifts (patches) because the fitness evaluation is used for the nurse's preferences, which does not include the Off shift (patch). Moreover, when the highest result from the fitness function has been provided, it is considered to be optimal having no violation counts of understaffing, overstaffing, and no less than 40 hours of work per week or having more than 2 offs.

### 5.3 Implementing BA Using an Automated Nurse Scheduling Program

---

**ALGORITHM 1:** Pseudo-code for the Bees Algorithm
 

---

1. *Initialize the population with a set number of scout bees distributed in the search space randomly*
2. *Evaluate the fitness of the population*
3. *While ( the stopping criterion is not met )*
4. *Select the highest points of fitness (by neighborhood search) to be visited by the bees*
5. *Send bees to the selected points of fitness (more bees to the best points) and evaluate*
6. *Select the bee with the highest fitness from each patch*
7. *Assign remaining bees to the search space and evaluate*
8. *End while*

The proponents modified the initialization of the algorithm. Based from the review of related literature, one weakness of BA is the randomization. To make it more effective, the proponents applied a part of the framework from the Mutingi and Mhbowa's PSO study (2014) which is satisfying the hard constraints in the initialization of the population. The bees (nurses) are randomly placed on four (4) initialized patches, which are the AM patch (shift), PM patch (shift), Graveyard patch (shift), and Off patch (shift) with a patch size of three (3), or the number of bees (nurses) per shift. During the initialization, the bees (nurse), with the highest preferences shift score assigned on each AM, PM and Graveyard patch, are considered as the "scout bees".

After initialization, the enhanced random schedule goes under a fitness evaluation, wherein the nurse's preferences or the soft constraints are considered. If the current fitness evaluation does not meet the criteria set, it continues a loop until a best schedule with the highest fitness evaluation is attained and that there are no violations for the Hard Constraints specified.

There are two cases for the algorithm. The first case is for the first day of the week and the second case is for the remaining days of the week. For the first day, the algorithm will first check the AM patch. It will now select a bee that has a score of one on the AM patch. That selected bee will now find its best flower patch or the flower patch which that selected bee has a score of three. It will now go to its selected patch and recruit a bee that has a score of one on the selected patch, and they will swap with the selected bee. If the first condition was not met, the selected bee will now recruit bees that have a score of two on the AM patch from the selected patch, and the other remaining flower patch. If either of the conditions meet, the recruited bee will now swap with the selected bee, and the selected bee will now become a scout bee. The algorithm will now go to the PM patch, and will now select a bee that has a score of one on the PM patch. That selected bee will now find its best flower patch, or the flower patch which that selected bee has a score of three. It is noted that the AM patch is optimized so that it will not compare with the same patch anymore. It will now go to its selected patch and recruit a bee that has a score of one on the selected patch, and they will swap with the selected bee. If the first condition is not met, the selected bee will now recruit a bee from the Graveyard patch that has a score of one or two on the PM patch. If either of the conditions meets, the recruited bee will now swap with the selected bee, and the selected bee will now become a scout bee. The algorithm will now go to the Graveyard patch, wherein it will now select a bee with a score of one on that patch. It will now compare the PM patch and since the AM patch is optimized, it will not compare to its own patch anymore. It

will now recruit a bee that has a score of two or three from the PM patch. Since Graveyard bees that have a score of one were compared from the PM patch, it will not compare with that patch anymore. If the conditions meet, the recruited bee will now swap with the selected bee, and the selected bee will now become a scout bee.

The second case will be the second day to the seventh day. Before it will go through the process, it will first encounter a Graveyard-AM shift control process. In this control process, the two lists will be compared, the list from the previous day and the list from the recent day. It will detect a Graveyard before that has an AM shift now. If the condition has been met, the AM shift will automatically be swapped with the Off shift. It will also check the Off shift if it is a Graveyard shift before. For instance, AM will be swapped to Off, and Off will be swapped with AM, if the Off patch that is swapped to the AM patch was a Graveyard patch before, so that the Graveyard-AM shift sequence will be avoided. After the process, it will now run the algorithm.

## 5.4 Testing and Evaluation

After completing the automated system, the proponents proceed on a 50-run testing by using the data gathered anonymously from one (1) station. Figure 5.7 has the following details: run number (1<sup>st</sup> column), the criterions 1-3 result for Weeks 1 to 4 (2<sup>nd</sup> – 4<sup>th</sup> columns), criterion 4 result for Week 1 (5<sup>th</sup> column), criterion 4 result for Week 2 (6<sup>th</sup> column), criterion 4 result for Week 3 (7<sup>th</sup> column), and criterion 4 result for Week 4 (8<sup>th</sup> column). The 9<sup>th</sup> column provides the overall criterion result for Week 1. The 10<sup>th</sup> column provides the overall criterion result for Week 2. The 11<sup>th</sup> column provides the overall criterion result for Week 3. The 12<sup>th</sup> column provides the overall criterion result for Week 4. The last column provides the overall result for Weeks 1-4.

Run	Cri1_w1-w	Cri2_w1-w	Cri3_w1-w	Cri4_w1	Cri4_w2	Cri4_w3	Cri4_w4					OVERALL
1	100	100	100	94.18	94.71	94.71	94.71	98.545	98.6775	98.6775	98.6775	98.64438
2	100	100	100	94.71	95.77	95.24	95.77	98.6775	98.9425	98.81	98.9425	98.84313
3	100	100	100	94.71	91.01	96.3	96.83	98.6775	97.7525	99.075	99.2075	98.67813
4	100	100	100	97.89	94.71	95.77	96.83	99.4725	98.6775	98.9425	99.2075	99.075
5	100	100	100	94.71	95.77	95.24	96.3	98.6775	98.9425	98.81	99.075	98.87625
6	100	100	100	93.12	97.36	95.2	92.59	98.28	99.34	98.8	98.1475	98.64188
7	100	100	100	93.65	94.18	94.18	94.18	98.4125	98.545	98.545	98.545	98.51188
8	100	100	100	96.3	96.3	95.24	94.18	99.075	99.075	98.81	98.545	98.87625
9	100	100	100	94.71	95.77	96.3	97.36	98.6775	98.9425	99.075	99.34	99.00875
10	100	100	100	96.83	94.18	92.59	93.12	99.2075	98.545	98.1475	98.28	98.545
11	100	100	100	94.18	95.24	93.12	93.12	98.545	98.81	98.28	98.28	98.47875
12	100	100	100	96.3	95.24	94.71	94.71	99.075	98.81	98.6775	98.6775	98.81
13	100	100	100	96.3	94.71	95.24	96.3	99.075	98.6775	98.81	99.075	98.90938
14	100	100	100	94.18	95.77	95.77	96.3	98.545	98.9425	98.9425	99.075	98.87625
15	100	100	100	95.24	94.18	96.3	94.71	98.81	98.545	99.075	98.6775	98.77688
16	100	100	100	98.41	95.77	94.18	95.77	99.6025	98.9425	98.545	98.9425	99.00813
17	100	100	100	97.88	93.12	95.24	96.83	99.47	98.28	98.81	99.2075	98.94188
18	100	100	100	92.59	93.12	92.59	94.71	98.1475	98.28	98.1475	98.6775	98.31313
19	100	100	100	96.3	97.88	95.24	95.24	99.075	99.47	98.81	98.81	99.04125
20	100	100	100	94.71	93.65	95.77	96.83	98.6775	98.4125	98.9425	99.2075	98.81
21	100	100	100	94.18	96.83	92.59	96.3	98.545	99.2075	98.1475	99.075	98.74375
22	100	100	100	94.71	95.77	97.88	94.18	98.6775	98.9425	99.47	98.545	98.90875
23	100	100	100	92.59	94.71	92.59	97.35	98.1475	98.6775	98.1475	99.3375	98.5775
24	100	100	100	94.71	96.3	95.77	95.24	98.6775	99.075	98.9425	98.81	98.87625
25	100	100	100	97.36	92.59	94.18	94.71	99.34	98.1475	98.545	98.6775	98.6775
26	100	100	100	95.24	96.83	94.71	94.71	98.81	99.2075	98.6775	98.6775	98.84313
27	100	100	100	96.3	95.24	95.24	96.83	99.075	98.81	98.81	99.2075	98.97563
28	100	100	100	93.12	95.24	95.24	94.71	98.28	98.81	98.81	98.6775	98.64438
29	100	100	100	98.41	94.71	94.71	93.65	99.6025	98.6775	98.6775	98.4125	98.8425
30	100	100	100	95.77	94.18	95.24	94.71	98.9425	98.545	98.81	98.6775	98.74375
31	100	100	100	94.71	96.3	92.59	93.65	98.6775	99.075	98.1475	98.4125	98.57813
32	100	100	100	94.71	96.3	95.77	94.71	98.6775	99.075	98.9425	98.6775	98.84313
33	100	100	100	96.3	94.18	93.12	94.18	99.075	98.545	98.28	98.545	98.61125

34	100	100	100	95.24	93.12	97.89	97.35	98.81	98.28	99.4725	99.3375	98.975
35	100	100	100	93.65	93.65	94.18	94.18	98.4125	98.4125	98.545	98.545	98.47875
36	100	100	100	91.54	92.06	94.18	94.18	97.885	98.015	98.545	98.545	98.2475
37	100	100	100	94.18	92.06	92.06	96.3	98.545	98.015	98.015	99.075	98.4125
38	100	100	100	94.71	93.12	95.24	94.71	98.6775	98.28	98.81	98.6775	98.61125
39	100	100	100	95.24	96.3	93.65	93.65	98.81	99.075	98.4125	98.4125	98.6775
40	100	100	100	93.65	94.18	95.24	96.83	98.4125	98.545	98.81	99.2075	98.74375
41	100	100	100	95.77	93.12	92.59	92.06	98.9425	98.28	98.1475	98.015	98.34625
42	100	100	100	95.77	94.71	96.3	92.06	98.9425	98.6775	99.075	98.015	98.6775
43	100	100	100	94.71	92.59	92.59	93.65	98.6775	98.1475	98.1475	98.4125	98.34625
44	100	100	100	91.54	93.12	96.3	95.77	97.885	98.28	99.075	98.9425	98.54563
45	100	100	100	94.71	95.24	94.71	93.12	98.6775	98.81	98.6775	98.28	98.61125
46	100	100	100	94.71	94.18	94.71	94.71	98.6775	98.545	98.6775	98.6775	98.64438
47	100	100	100	97.88	93.65	95.77	95.24	99.47	98.4125	98.9425	98.81	98.90875
48	100	100	100	94.71	95.24	95.24	93.65	98.6775	98.81	98.81	98.4125	98.6775
49	100	100	100	95.24	94.71	95.24	92.59	98.81	98.6775	98.81	98.1475	98.61125
50	100	100	100	95.24	97.36	92.59	91.01	98.81	99.34	98.1475	97.7525	98.5125

Figure 5.7: 50-run Testing Results

Computational results show that the proposed algorithm achieves better results when automated since violations have been easily rejected, because of the framework used in this study. The 4<sup>th</sup> run of the system was able to obtain the optimal 1-month schedule with a 99.075% overall criterion percentage.

Name	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	am	pm	g	o
n1	g	g	g	g	g	o	o	0	0	5	2
n2	am	am	am	am	o	am	am	6	0	0	1
n3	am	am	am	am	am	o	o	5	0	0	2
n4	pm	o	pm	pm	pm	o	pm	0	5	0	2
n5	o	g	g	o	g	g	g	0	0	5	2
n6	g	o	o	g	g	g	g	0	0	5	2
n7	pm	o	am	o	am	am	am	4	1	0	2
n8	am	am	g	o	o	am	am	4	0	1	2
n9	g	g	o	g	o	g	g	0	0	5	2
n10	o	pm	pm	pm	am	pm	pm	1	5	0	1
n11	pm	pm	pm	pm	pm	pm	o	0	6	0	1
n12	o	pm	o	am	pm	pm	pm	1	4	0	2
am	3	3	3	3	3	3	3				
pm	3	3	3	3	3	3	3				
g	3	3	3	3	3	3	3				
o	3	3	3	3	3	3	3				

EVALUATION	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
am	100.00	100.00	100.00	88.89	88.89	100.00	100.00
pm	88.89	100.00	100.00	100.00	100.00	100.00	100.00
g	100.00	100.00	88.89	100.00	100.00	100.00	100.00
OVERALL	96.30	100.00	96.30	96.30	96.30	100.00	100.00

Figure 5.8: Optimal Week 1 Schedule (Run 4)

Name	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	am	pm	g	o
n1	g	g	g	g	g	o	o	0	0	5	2
n2	am	am	am	am	am	am	o	6	0	0	1
n3	am	o	am	am	o	g	am	4	0	1	2
n4	pm	pm	o	pm	o	pm	pm	0	5	0	2
n5	g	g	o	o	g	g	g	0	0	5	2
n6	pm	g	g	g	pm	g	o	0	2	4	1
n7	o	am	am	am	am	am	am	6	0	0	1
n8	o	am	g	g	am	o	am	3	0	2	2
n9	g	o	pm	o	g	pm	g	0	2	3	2
n10	am	pm	o	pm	pm	o	pm	1	4	0	2
n11	pm	pm	pm	o	o	pm	g	0	4	1	2
n12	o	o	pm	pm	pm	am	pm	1	4	0	2
am	3	3	3	3	3	3	3				
pm	3	3	3	3	3	3	3				
g	3	3	3	3	3	3	3				
o	3	3	3	3	3	3	3				

EVALUATION	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
am	88.89	100.00	100.00	100.00	100.00	88.89	100.00
pm	88.89	100.00	88.89	100.00	88.89	88.89	100.00
g	100.00	100.00	88.89	88.89	100.00	88.89	88.89
OVERALL	92.59	100.00	92.59	96.30	96.30	88.89	96.30

Figure 5.9: Optimal Week 2 Schedule (Run 4)

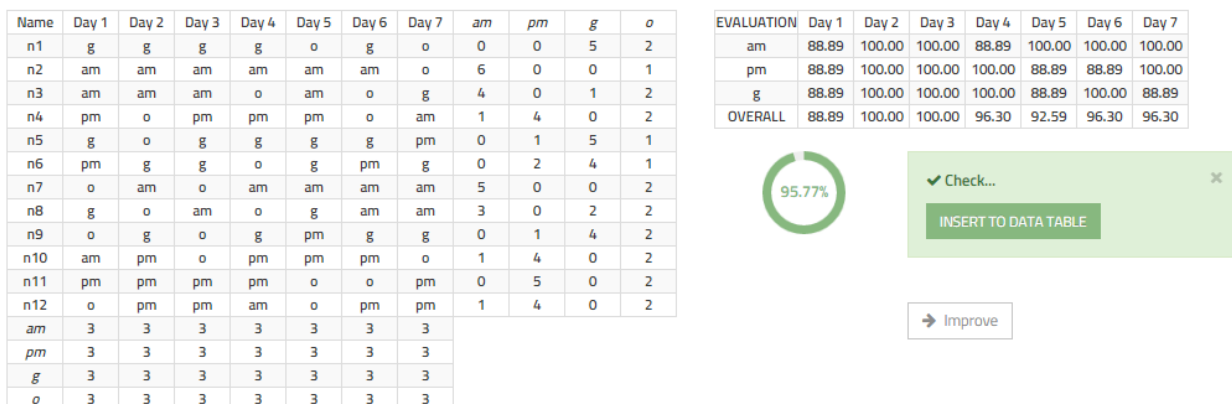


Figure 5.10: Optimal Week 3 Schedule (Run 4)

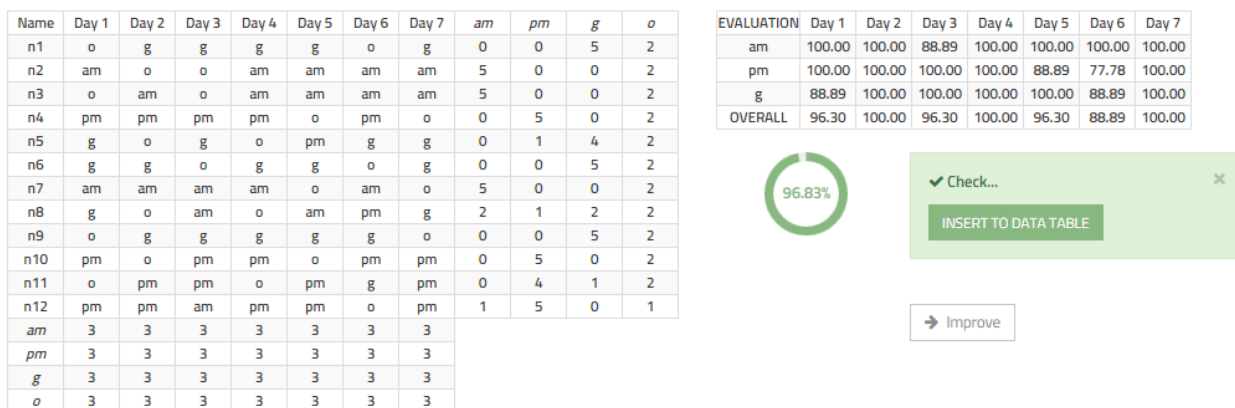


Figure 5.11: Optimal Week 4 Schedule (Run 4)

Based on the testing results, generated schedules with the algorithm were able to reach optimal solutions, in the range of 98.71%, and with no violations at all. In addition, all generated schedules by the said system were also evaluated and approved by a head nurse. Thus, BA is considered to be an efficient and effective algorithm in solving NSP.

## 6. CONCLUSIONS AND RECOMMENDATIONS

### 6.1 Conclusions

In this study, BA with enhanced random initialization for the hard constraints and fitness evaluation for the soft constraints is proposed to solve the nurse scheduling problem, producing near-optimal solutions. All objectives were accomplished. In addition, using automated schedulers are still uncommon practice; however this study is a good jump-start. Also, the study can lead to improvements in hospital resource efficiency, staff satisfaction and administrative workload. With this, it could offer more powerful decision scheduler systems which better capture the requirements and needs of the hospital workplace. This proposed approach offers beneficial contribution to researchers, as well as practitioners in the health service industry.

#### 6.1.1 Contributions to BA Theory

Utilization of BA in the study is considered to be a contribution to the healthcare management science and operations management community. It primarily offers an effective approach to solving nurse scheduling problems in the presence of management goals and preferences. The algorithm has the ability to deliver good solutions within a reasonable time. Moreover, the algorithm forms a platform for additional decision support for decision makers in a specific field.

#### 6.1.2 Contributions to Nursing Practice

The proposed utilization of BA in this field uses an interactive approach that provides good solutions, rather than a single “optimal solution”. This is considered to be more acceptable to most practicing decision makers in the field. Therefore, the user can utilize further information from nurses and the management to make adjustments to the solution process. Moreover, as part of the scientific community, everything should consider certain issues that are significant to modern hospital administrators in order to raise the uptake of nurse scheduling research in the real world. The Bees Algorithm is considered to be an effective and efficient algorithm for solving NSP.

### 6.2 Recommendations

First and foremost, the proponents recommend to future Computer Science student researchers to consider this study as a Systems Analysis and Development (SAD) project for system improvement and enhancement to cater the whole nursing department, since the implementation only caters one (1) station with three (3) AM staff nurses, three (3) PM staff nurses, three (3) Graveyard staff nurses and three (3) Off staff nurses.

Second, the proponents recommend to future researchers to add other constraints like the Congeniality factor or the compatibility among staff nurses.

Lastly, the proponents also recommend to future researchers to consider comparing BA to other existing Swarm Intelligence approaches used to also solve NSP, particularly PSO and ACO. PSO has already been applied, but the study with ACO only provided a model.

## REFERENCES

- Ablazo, A.C., Bonga, E.J.R., Viscayno, V.C.B. 2012. Forming student groups using Bees Algorithm. Ateneo De Davao University. School Of Arts And Sciences, Davao City.
- Ahmed, H., Glasgow, J. 2012. Swarm intelligence: concepts, models and applications. School of Computing Queen's University Kingston, Ontario, Canada. Retrieved February 3, 2015 from <http://research.cs.queensu.ca/TechReports/Reports/2012-585.pdf>.
- Aickelin, U., Dowsland, K.A. 2004. An indirect genetic algorithm for a nurse scheduling problem. *Computers & Operations Research* (arXiv) 31 (5): 761–778. doi:10.1016/s0305-0548(03)00034-0. Retrieved February 3, 2015 from <http://ima.ac.uk/papers/aickelin2004c.pdf>.
- Bellanti, F., Carello, G., Della Croce, F., Tadei, R. 2004. A greedy-based neighborhood search approach to a nurse rostering problem, *European Journal of Operational Research*, Vol. 153, No. 1, 2004, 28-40. Retrieved January 3, 2015 from [http://www.researchgate.net/profile/Roberto\\_Tadei/publication/220289726\\_A\\_greedy-based\\_neighborhood\\_search\\_approach\\_to\\_a\\_nurse\\_rostering\\_problem/links/09e4150c5c5c097a08000000.pdf](http://www.researchgate.net/profile/Roberto_Tadei/publication/220289726_A_greedy-based_neighborhood_search_approach_to_a_nurse_rostering_problem/links/09e4150c5c5c097a08000000.pdf).
- Burke, E. K., Causmaecker, P. D., Berghe, G. V., and Landeghem, H. V. 2004. The state of the art of nurse rostering. *J. Journal of Scheduling*, 7, 441–499. Retrieved January 3, 2015 from <https://lirias.kuleuven.be/bitstream/123456789/123829/1/JOS.11>.
- Burke, E.K., De Causmaecker, P., Petrovic, S., Vanden Berghe, G. 2002. A multi criteria meta-heuristic approach to nurse rostering, *Proceedings of Congress on Evolutionary Computation, CEC2002*, Honolulu, IEEE Press, 2002, 1197-1202.
- Dias, T.M., Ferber, D.F., De Souza, C.C., Moura A.V. 2003. Constructing nurse schedules at large hospitals. *International Transactions in Operational Research*. 10 (2003) 245–265.
- Ernst, A.T., Jiang, H., Krishnamoorthy, M., Sier, D. 2004. Staff scheduling and rostering: A review of applications, methods and models, *European Journal of Operational Research*, Vol. 153, No. 1, 2004, 3-27.
- Goldberg, D.E. 1989. *Genetic algorithms in search, optimization, and machine learning*. Reading, MA Addison Wesley.
- Mastrocinque, E., Yuce, B.; Lambiase, A., Packianather, M.S. 2013. A multi-objective optimisation for supply chain network using the bees algorithm. *Int. J. Eng. Bus. Manage.* 2013, 5, 1-11.
- Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E. 1953. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics* 21 (6): 1087.
- Muslija, N., Gaertner, J., Slany, W. 2002. Efficient generation of rotating workforce schedules, E.K. Burke, W. Erben (Eds.): *Proceedings of the 3rd International Conference on the Practice and Theory of Automated Timetabling*, ISBN 3-00-003866-3, 2000, 314-332.
- Mutingi, M., Mbohwa, C. 2014. Healthcare staff scheduling in a fuzzy environment: A fuzzy genetic algorithm approach, *Proceedings of the 2014 International Conference on Industrial Engineering and Operations Management*, Bali, Indonesia, January 7 – 9, 2014, pp. 3038-3047.
- Petrovic, S., Beddoe, G., Vanden Berghe, G. 2002. Storing and adapting repair experiences in personnel rostering, E.K. Burke, P. De Causmaecker (Eds.), *Proceedings of the 4th International Conference on Practice and Theory of Automated Timetabling*, Gent, 2002, 185-186.
- Pham, D., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., Zaidi, M. 2006. The bees algorithm - a novel tool for complex optimisation problems. Retrieved February 3, 2015 from <https://svn-d1.mpi-inf.mpg.de/AG1/MultiCoreLab/papers/Pham06%20-%20The%20Bee%20Algorithm.pdf>.
- Pham, D.T., Afify, A.A., Koc, E. 2007. Manufacturing cell formation using the bees algorithm. In *Proceedings of the Innovative Production Machines and Systems Virtual Conference*, Cardiff, UK, 2013 July 2007.
- Pham, D.T., Ghanbarzadeh, A. 2007. Multi-objective optimisation using the bees algorithm. In *Proceedings of the IPROMS 2007 Conference*, Cardiff, UK, 2013 July 2007.
- Pham, D.T., Koc, E., Lee, J.Y., Phruksanant, J. 2007. Using the bees algorithm to schedule jobs for a machine. In *Proceedings of the 8th International Conference on Laser Metrology, CMM and Machine Tool Performance*, Cardiff, UK, 2-13 July 2007; pp. 430-439.
- Pham, D.T., Otri, S., Afify, A., Mahmuddin, M., Al-Jabbouli, H. 2007. Data clustering using the bees algorithm. In *Proceedings of the 40th CIRP International Manufacturing Systems Seminar*, Liverpool, UK, 30 May-1 June 2007.
- Pham, D.T., Soroka, A.J., Ghanbarzadeh, A., Koc, E., Otri, S., Packianather, M. 2006. Optimising neural networks for identification of wood defects using the bees algorithm. In *Proceedings of the 2006 IEEE International Conference on Industrial Informatics*, Singapore, 16-18 August 2006; pp. 1346-1351.
- Pham, D.T., Soroka, A.J., Koc, E., Ghanbarzadeh, A., Otri, S. 2007. Some applications of the bees algorithm in engineering design and manufacture. In *Proceedings of International Conference On Manufacturing Automation (ICMA 2007)*, Singapore, 28-30 May 2007.
- Saji, Y., EssaidRiffi, M., Ahiod, B. 2013. Multi-objective ant colony optimization algorithm to solve a nurse scheduling problem, *International Journal of Advanced Research in Computer Science and Software Engineering*, August - 2013, pp. 311-320.
- Silvestro, R., Silvestro, C. 2000. An evaluation of nurse rostering practices in the National Health Service, *Journal of Advanced Nursing*, Vol. 32, No. 3, 2000, 525-535.
- Suman, B., Kumar, P. 2006. A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the Operational Research Society* (2006) 57, 1143–1160.
- Webster, S., Jog, P.D., Gupta, A., 1998. Genetic algorithm for scheduling job families on a single machine with arbitrary earliness/tardiness penalties and an unrestricted common due date. *International Journal of Production Research* 36, 2543–2551.
- Wren, A. 1996. Scheduling, timetabling and rostering - a special relationship? In *Proc. of 1st Practice and Theory of Automated*

Timetabling, 46-75.

Yuce, B. 2012. Novel computational technique for determining depth using the bees algorithm and blind image deconvolution. Ph.D Thesis, Cardiff University, Cardiff, UK, 2012.