

## “Staff Scheduling in Health Care Systems”

Mudra S. Gondane<sup>1</sup>, Prof. D. R. Zanwar<sup>2</sup>  
<sup>1,2</sup>(Industrial Engineering, S.R.C.O.E.M. Nagpur, India)

---

**Abstract:** Staff scheduling is found as a crucial part of staff management. Every scheduling problem is different from each other when considering the various constraints, which leads to the staff scheduling problem becoming more complex. This paper deals with Genetic Algorithm (GA) approach to solve a specific Staff Scheduling Problem. The schedules are planned for 4 weeks with 30 numbers of nurses in the ward and each working day consists of 5 different shift types. Penalty is presented to indicate the level of importance for the constraints involved that are hard constraints and soft constraints. For solving the problem of scheduling, GUI is being introduced for the user friendly approach through MATLAB Version 7.10.

**Keywords:** - Genetic Algorithm, GUI (Graphical User Interface), MAT Lab, NSP (Nurse Scheduling Problem), Staff scheduling.

---

### I. Introduction

#### 1.1. Nature of Problem:-

Personnel scheduling, or Staff Scheduling or Rostering, is the process of constructing work timetables for its staff so that an organisation can satisfy the demand for its goods or services. Hence Scheduling problem is one of optimization problem, and it relates to some other similar problems such as timetabling, sequencing and rostering. Health services pose a challenge for the automatic rostering systems. While in other areas we may accept lower quality rosters, the demands on the quality of nurse rosters are uncompromising. Hospitals simply cannot afford stressed, tired or overworked nurses and of course, the more content the nurses are, the better the healthcare is. Thus, Nurse Scheduling Problem (NSP) is a complex optimization problem of allocating nurses to duty rosters in hospitals, and it is increasingly becoming a focal point in the world because of cost pressures on hospitals. The complexity of this problem is confirmed by the fact that nurse rostering falls under NP hard problems. NP hard problems cannot be effectively solved by deterministic methods therefore the most used approach involves heuristics.

#### 1.2. Previous Work:-

Creation of a roster is indeed a complex task. Thus, The Nurse Scheduling Problem (NSP) is a complex optimization problem of allocating nurses to duty rosters in hospitals. Many hospitals are facing the dilemma that they have to provide high-quality services with a relative lower cost, so they identify the resources including nurses for eliminating the sources of waste [8]. In every different case, the essence of the NSP remains the same, which is to assign every nurse in the ward to a suitable set of shifts during the given period. However, the objectives of the NSP range from minimizing costs to maximizing the satisfaction of the personnel involved. Since NSP is becoming more complex, a range of methods are presented to solve the problems. The methods often based on mathematical programming, artificial intelligence and a variety of heuristics such as Genetic Algorithms (GA), Ant Colony Optimization (ACO) simulated annealing, tabu search and other forms of local search [1].

#### 1.3. Purpose:-

The purpose of this paper is to solve a specific nurse scheduling problem using Genetic Algorithm (GA) approach. This paper focuses on solving a specific scheduling problem for nurses using Genetic Algorithm. Additionally, there are some hard constraints and soft constraints for solving this scheduling problem. Hard constraints must be satisfied, and soft constraints are not necessarily to be satisfied. Penalties are provided to represent for the level of importance for each constraint. Penalty is incurred due the violations of soft constraints and the fitness function is the sum of total penalty. The objective of this problem is to minimize the total penalty which means to maximize the requirements of nurses.

#### 1.4. Contribution:-

This paper focuses on solving a specific scheduling problem for nurses in hospitals using Genetic Algorithm. The schedule is planned for four weeks where each working day consists of five different shift types in working days which are Morning, Afternoon, Evening and Night. Also, free day represents for a day off.

There are thirty nurses in the ward. According to the demand schedules that are provided by hospitals, the optimized schedule will be designed using genetic algorithm in this paper. In our case the nurse scheduling represents the generic problem, which covers tasks from long term planning to short term on-site duty assignments.

In this paper we generate an initial population randomly including the result obtained by using GA and will go through crossover and mutation to find near optimal solution. Result will show the Schedule for Staff for the given number of shifts in a planning period by considering constraints. We represent the model by using GUI in MATLAB.

## II. Literature Review

This chapter will review the relevant literature and background on nurse scheduling problems. A number of researchers have used genetic algorithms to solve the staff-scheduling problem. Ernst, Jiang, and Krishnamoorthy [1] classified sixteen different scheduling models which are based on the rostering processes or determination of staff requirements such as task based demand, shift based demand, days off scheduling, shift scheduling, and task assignment etc. The authors also addressed fifteen different application areas of the scheduling problems including airlines, buses, nurse scheduling, venue management, financial services, and manufacturing etc. Finally, 28 different categories of methods are identified to solve the personnel scheduling problems like artificial intelligence, constraint logic programming, constructive heuristic, simple local search, simulated annealing, tabu search, genetic algorithms etc.

Amol C. Adamuthe and Rajankumar Bichkar [2], suggested a solution for nurse scheduling in medical centre where the objectives are to assign staff to particular day in planning period and minimization of personnel cost by avoiding overtime pay. For nurse scheduling problem, they proposed hybrid genetic algorithms with partial feasible chromosome representation, initialization and operators have shown fast convergence towards optimal solution with comparatively small population size. The probability of getting near optimal solution using proposed hybrid genetic algorithm in less than 20 seconds (the average time) is more than 0.6.

John S. Dean [3], implemented two genetic algorithm staff-scheduling solutions for scheduling nurses at a hospital. One solution uses a traditional bit-string chromosome structure to represent each schedule. The other solution uses a two-dimensional array chromosome structure to represent each schedule. Experimental results show that his two-dimensional array staff-scheduling implementation performs better than his bit-string staff-scheduling implementation.

Uwe Aickelin and Kathryn A. Dowsland [4][7], described a Genetic Algorithms approach to a manpower-scheduling problem arising at a major UK hospital. Computational experiments based on 52 weeks of live data are used to evaluate three different decoders with varying levels of intelligence, and four well-known crossover operators. The results reveal that the proposed algorithm is able to find high quality solutions and is both faster and more flexible than a recently published Tabu Search approach.

Tiago M. Diasa, Daniel F. Ferberb, Cid C. de Souza and Arnaldo V. Mourad [5] suggested Several heuristics, based on evolutive algorithms and local search, are used to solve the nurse scheduling problem at a large hospital. Due to several intricate and specific restrictions imposed on the schedules, the problem is a difficult one to solve by hand. In order to facilitate the use of the solver modules by the hospital staff, a user interface was also implemented.

Bloch liger [6] gave an introduction to staff scheduling problem using a hospital example. The tutorial mainly focused on how the problem can be analyzed and modelled including various kinds of constraints, objectives, and models. According to hospital type, the basic type of constraints like sequence constraints, job constraints, hard constraints, and soft constraints are all considered. Nonetheless, this tutorial did not give the solution to solve this problem. On the basis of this modelling, a number of algorithms are raised such as genetic algorithm, simulated annealing, and tabu search.

## III. The nurse scheduling problem (NSP)

The term nurse scheduling problem is often accompanied and sometimes even interchanged with the term nurse rostering. To understand their meanings we must first define the difference between the scheduling and rostering.

### 3.1 Scheduling and rostering

According to the research, the terms scheduling and rostering are defined as follows:-

- **Scheduling** is the allocation, subject to constraints, of resources to objects begins placed in space-time, in such a way as to minimise the total cost of some set of the resources used.
- **Rostering** is the placing, subject to constraints, of resources into slots in a pattern.

In our case the nurse scheduling represents the generic problem, which covers tasks from long term planning to short term on-site duty assignments [8]. In this paper we generate an initial population randomly including the result obtained by using GA and will go through crossover and mutation to find near optimal

solution. Result will show the Schedule for Staff for the given number of shifts in a planning period by considering constraints. We represent the model by using GUI in MATLAB.

### 3.2 Constraints

This section describes the hard and soft constraints for nurse scheduling problem described by Dean, Adamuthe et.al. Uwe Aickelin et.al. [5][2][6]

#### 3.2.1 Hard Constraints

Hard constraints are those that must be satisfied. Violation of these constraints (also called as conflicts) will cause the solution to be infeasible which is not accepted. Solutions which satisfy hard constraints are called feasible solutions.

HC1: Create a 4-week schedule. The hard constraint varies from defined by Dean. Dean has proposed solution by considering Sunday as start day where as in this paper we have considered Monday is start day.

HC2: Fixed number of working employees as required during each day of planning period.

HC3: Each employee works at least five days per week.

HC4: Maximum number of consecutive working days for any employee is six.

HC5: Minimum time gap required between two shifts should be of 12 hours.

#### 3.2.2 Soft Constraints

Soft constraints are those that are desirable in order to produce a good quality timetable but violations are allowed to satisfy hard constraints.

*Cost Minimization Constraints:*

SC1: Try to avoid more than 3 working days per employee per week because that leads to overtime pay.

SC2: Try hard to avoid more than 4 working days per employee per week because that leads to additional overtime pay.

*Personal Demands:*

SC3: If an individual wants to avoid working 3 days in a row try to accommodate.

SC4: If an individual wants to maximize the number of grouped working days, then try to maximize the number of grouped working days.

*General:*

SC5: Try to balance the number of Monday and Friday off days that coincide with off weekends.

In this paper, focus would be to solve the hard constraints only as making them the primary priority for nurse scheduling. Such that the main objective for solving the problem would be allocation of nurses or staff to particular slot in planning period.

### 3.3 Problem Formulation

The nurse scheduling problem can be formulated as an integer linear program as follows. [4][7]

#### Indices:

$i = 1 \dots n$  nurse index.

$j = 1 \dots m$  shift pattern index.

$k = 1 \dots 14$  day and night index (1...7 are days and 8...14 are nights).

$s = 1 \dots p$  grade index.

#### Decision variables:-

The problem can be formulated by taking the following indices/variables for integer linear program. [4][7]

$$x_{ij} = \begin{cases} 1 & \text{nurse } i \text{ works shift pattern } j \\ 0 & \text{else} \end{cases}$$

Parameters:

$n$  = number of nurses

$m$  = number of possible shift patterns

$p$  = number of grades

$$a_{jk} = \begin{cases} 1 & \text{pattern } j \text{ covers day } k \\ 0 & \text{else} \end{cases}$$

$$q_{is} = \begin{cases} 1 & \text{nurse } i \text{ is of grade } s \text{ or higher} \\ 0 & \text{else} \end{cases}$$

$p_{ij}$  = penalty cost of nurse  $i$  working shift pattern  $j$

$R_{ks}$  = demand of nurses of grade  $s$  or above on day/night  $k$

$F(i)$  = set of feasible shift patterns for nurse  $i$

Target function:

$$\sum_{i=1}^n \sum_{j=1}^m p_{ij} x_{ij} \rightarrow \min! \quad (1)$$

s.t.

$$\sum_{j \in F(i)} x_{ij} = 1 \quad \forall i \quad (2)$$

$$\sum_{j=1}^m \sum_{i=1}^n q_{is} a_{jk} x_{ij} \geq R_{ks} \quad \forall k, s \quad (3)$$

Constraints (2) ensure that every nurse works exactly one shift pattern from her feasible set.

Constraints (3) ensure that the demand for nurses is fulfilled for every grade on every day/night. Note that the definition of ‘q’ ensures that higher graded nurses can substitute those at lower grades if necessary.

This problem can be regarded as a multiple-choice set-covering problem. The sets are given by the shift pattern vectors and the objective is to minimise the cost of the sets needed to provide sufficient cover for each shift at each grade. The multiple-choice aspect derives from constraints (2), which enforce the choice of exactly one pattern (or set), from the alternatives available for each nurse.

As shown above in ILP lot of mathematical calculation is needed for solving this kind of staff scheduling problem. Hence to overcome the process of calculation we will be using a GUI in MAT Lab for getting the optimized result for NSP.

## IV. Description of Methodology

### 4.1. Background of Genetic Algorithm

Charles Darwinian evolution in 1859 is intrinsically a so bust search and optimization mechanism. Darwin’s principle “Survival of the fittest” captured the popular imagination. This principle can be used as a starting point in introducing evolutionary computation. Evolutionary computation, describes the field of investigation that concerns all evolutionary algorithms and offers practical advantages to several optimization problems. The advantages include the simplicity of the approach, its robust response to changing circumstances, and its flexibility and so on. GA were formally introduces in the US in the 1970 by John Holland at University of Michigan. This is one of the efficient optimization tool, work very well on mixed (Continuous and discrete) combinatorial problems.

To implement GA it is must to represent a solution to the problem as a genome (Chromosome). The GA then creates populations of solution and applies GA operators such as crossover and mutation to evolve optimal solution. There are three most important aspects of GA: Objective function is to be defined, Genetic representation is defined and implemented and implementation of GA operators. GA differs from conventional heuristic optimization techniques. GA works with the coding, GA uses population of solutions rather than the single solution for searching GA uses fitness function to evaluate rather than derivatives.

The evolutionary process of a GA is a highly simplified and stylized simulation of the biological version. It starts from a population of individuals randomly generated according to some probability distribution, usually uniform and updates this population in steps called generations. Each generation, multiple individuals are randomly selected from the current population based upon some application of fitness, bred using crossover, and modified through mutation to form a new population.

- **Crossover** – exchange of genetic material (substrings) denoting rules, structural components, features of a machine learning, search, or optimization problem
- **Selection** – the application of the fitness criterion to choose which individuals from a population will go on to reproduce.
- **Replication** – the propagation of individuals from one generation to the next.
- **Mutation** – the modification of chromosomes for single individuals.

### 4.2. Strengths of Genetic Algorithm

1. Genetic algorithms are intrinsically parallel. They can explore the solution space in multiple directions at once. So convergence to an optimal solution does not depend on the chosen initial solution.
2. They are well-suited to solving problems having huge search space.
3. They perform well in problems for which the fitness landscape is complex - where the function is Discontinuous, noisy, changes over time, or has many local optima.

#### 4.3. Proposed Approach for NSP based on Genetic Algorithm

As explained in previous chapter III, the complexity involved for using ILP method requires lots of calculation so Genetic Algorithmic approach was used in GUI to make it simpler and outlook oriented for presentation.

This paper will use real number coding such that the data would be presented in real number. The chromosomes of GA are a long string with 840 elements, which stores the information of all the employees. Since each nurse is assigned to one and only one shift per day, each nurse has seven elements for four weeks and each elements represents for the type of shifts for that day.

The integer numbers used for representing different types of shifts are given below:-

Free shift- integer 0

Shift 1- integer 1

Shift 2- integer 2

Shift 3- integer 3

Shift 4- integer 4

Shift 5- integer 5

As there are 30 nurses in this problem, the chromosomes would include 840 elements (30\*7\*4)

30-represents the number of nurses, 7-no.of days and 4- no. of week needed to be scheduled.

#### 4.4. Parameters used while encoding for GA

For solving specific type of problems in genetic algorithm encoding is required. Different parameters are needed for the purpose of encoding; some parameters are listed below in the table used for solving NSP,

| Parameter / Strategy         | Setting |
|------------------------------|---------|
| Elite Count                  | 1       |
| Number of Initial Population | 100     |
| The length of chromosome     | 840     |
| Crossover rate               | 0.8     |
| Mutation rate                | 0.002   |
| Max. Number of generation    | 500     |
| Stall Generation             | 100     |

Table 4.4: Parameters used for the genetic algorithm.

#### 4.5. Genetic Operators

The genetic operators in MATLAB Global Optimization toolbox are used to test this problem and each operator is listed as below:

- **Selection:** Tournament selection is used to select the best individual.
- **Elite Count:** The elitism operator is set to select 1 individual (the best one) for the next generation.
- **Crossover:** Two-point crossover method was used as crossover function.
- **Mutation:** The method which is adaptive feasible was used as mutation function, and this method is used for the problem this contains lower or upper bounds or non linear constraint.
- **Hybrid Function:** Pattern search is used as a hybrid function to increase the efficiency of selection with constraints.

#### 4.6. Data Structures and its Representation

- ❖ **Initial Population:** Most of the researchers prefer to generate the initial population randomly but this will consumes lot of time to get an optimal solution. In this we start the initial population with a randomly generated preference matrix, including values involved for generating the nurse schedule.
- ❖ **Parent Selection:** We choose using the MATLAB default value for random selection of parents.
- ❖ **Fitness value:** We use fitness function for finding near optimal solution.
- Initialize population was generated randomly among the preference matrices, for generating feasible schedule.
- The fitness was defined by objective of NSP model, and individual adaptive value was evaluated.
- The crossover was operated in the population according to probability of crossover  $P_c$ , so the offspring are generated.
- The individual was selected randomly according to probability of mutation  $P_m$ , so that the offspring are generated.
- The new individual adaptive value was calculated, parent and offspring are taken part in survival competition together.

Adjusting the termination criterion, then the optimal solution was obtained, otherwise going back to first step.

**4.7. Genetic Algorithm flow chart:**

```
[  
Generate initial population  
Evaluate population  
While stopping criteria not satisfied Repeat  
[  
Select elements from  $P_t$  to copy into  $P_{t+1}$   
Crossover elements of  $P_t$  and put into  $P_{t+1}$   
Mutation elements of  $P_t$  and put into  $P_{t+1}$   
Evaluate new population  $P_{t+1}$   
 $P_t = P_{t+1}$   
]  
]
```

**V. Designing of GUI in Mat Lab**

In working environment not only workers but also executives have no time to spend doing mathematical calculations to get a solution. Here we have used GUI (Graphical User Interface) base programming in which the two aspects are taken care one is presentation i.e. layout and accordingly the programming is done. GUI provides a set of tools for creating the files. GUIDE tools are used as follows,

**5.1. GUI Layout**



**Figure 5.1 GUI Layout**

GUIDE (Graphical User Interface Development Environment) the MATLAB, provides a set of tools for creating graphical user interfaces (GUIs). These tools simplify the process of laying out and programming of GUIs. Using the GUIDE Layout Editor, we can populate a GUI such as axes, panels, buttons, text fields, sliders etc. into the layout area. From the Layout Editor, you can seize the GUI, and modify components look and feel, align components, set tab order, view a hierarchical list of the component objects, and set GUI options.

**5.2. GUI Programming**

GUIDE automatically generates a program file containing MATLAB functions that controls how the GUI operates. This code file provides code to initialize the GUI and contains a framework for the GUI call backs the routines that execute when a user interacts with a GUI component.

**5.3. Designing of GUI for NSP**

This window gives the main appearance for the nurse scheduling problem where different parameters were set on the left side using toggle button and the blank side gives the result whenever the button is clicked.

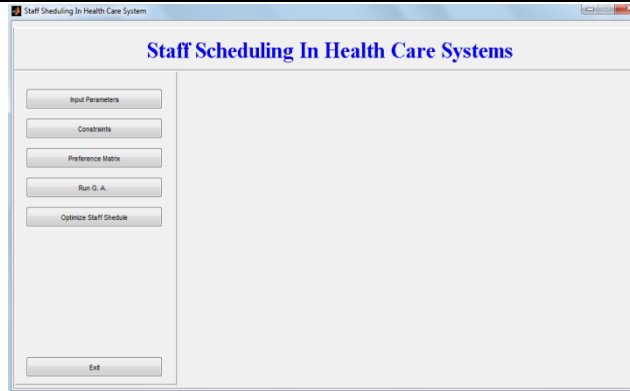


Figure 5.2 GUI Window for NSP

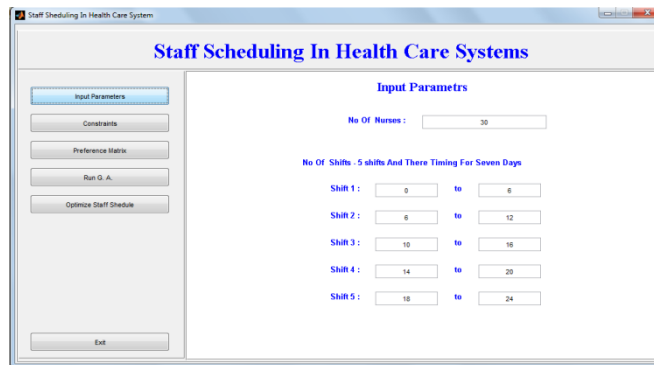


Figure 5.3 GUI windows when input parameter button is clicked



Figure 5.4 GUI windows when constraints button is clicked

Similarly, different buttons are clicked and the output is viewed in the GUI window according to the requirements of the user. GUI helps in reducing the time involved for solving mathematical expressions and gives a perfect pattern for presenting the problem along with the solution.

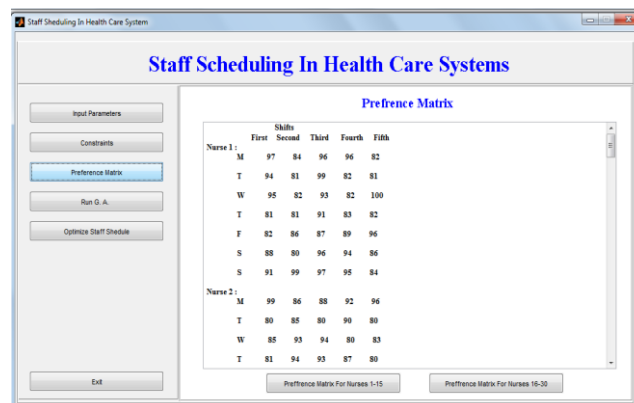


Figure 5.5 GUI windows when preference matrix button is clicked

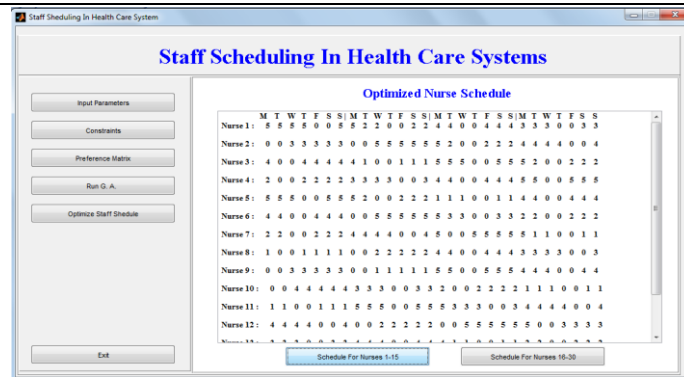


Figure 5.6 GUI windows when optimize staff schedule button is clicked

#### 5.4. Programming GUI for generating m-file

GUIDE automatically generates a program file containing MATLAB functions that controls how the GUI operates. This code file provides code to initialize the GUI and contains a framework for the GUI callbacks—the routines that execute when a user interacts with a GUI component. Use the MATLAB Editor to add code to the callbacks to perform the actions you want the GUI to perform the following steps.

**Step1:** Initially we design the GUI considering the Input and output parameters.

**Step2:** Create callback function for each parameter in m.file

**Step 3:** Give string name and Tag for each button in the Property Inspector

**Step 4:** Finally, using GA Parameters coding is done for NSP.

## VI. Case Study

### 6.1. Problem Description

The NSP data in real world is different from each hospital because of the constraints and objectives involved. This work focuses on solving the nurse scheduling problem which is provided by Uwe Aickelin and Kathryn Dowsland [7] they described the genetic algorithmic approach to manpower scheduling problem arising at major UK hospitals. Experiment is based on 52 weeks data to evaluate three different decoders along with different crossover operators and compared the result between GA and the Tabu Search algorithm.

This paper uses the GA approach to solve defined problem for given set of data. The schedules are planned for four weeks where each working day consist of five different shift types in a working day which are categorized into two Morning, one Afternoon, Evening and Night shift respectively. Also Free Day/Holiday represents for a day off as an integer 0.

The shift timings are enlisted below-

Shift 1 – 0.00 to 6.00

Shift 2 – 6.00 to 12.00

Shift 3 – 10.00 to 16.00

Shift 4 – 14.00 to 20.00

Shift 5 – 18.00 to 24.00

There are 30 nurses in the ward. Additionally, there are some hard constraints and soft constraints for this scheduling problem presented earlier.

Hard constraints are the constraints that must be satisfied, and soft constraints are the desirable constraints that are not necessarily to be satisfied. Weights are provided to represent for the level of importance for each constraint.

For example all the penalties/weight incorporated for hard constraints are different for different constraints according to the requirements for NSP.

The higher the number, the more strongly desired the constraint or request will be. Penalty is incurred due the violations of soft constraints and the fitness function is the sum of total penalty. The objective of this problem is to minimize the total penalty which means to maximize the requirements of nurses.

According to the demand schedules that are provided, the optimized schedules will be designed using genetic algorithm in this work.

### 6.2. Penalties

The cost of a solution is the sum of the penalties associated with the solution. The search Procedure seeks to maximize the fitness value of a solution, which is the inverse of its cost. Weights are provided to represent for the level of importance for each constraint. For example all the weights of hard constraints and soft constraints would be ranging from 5 to 100 according to the importance for the problem. The higher the number,



the more strongly desired the constraint or request is. In most of the cases, the penalties are provided by head nurses, but for this problem, the penalties are used according to the requirement of the problem.

Penalty 100: This number is for hard constraint and the hard constraints should not be violated. This penalty is given for the weeks (working days)

Penalty 100: Penalty for minimum weeks involved for creating schedule.

Penalty 10: The constraint is strongly desired and given for the consecutive shift types involved.

Penalty 10: The constraint is preferred for the employees/nurses involved.

Penalty 10: This constraint involves penalty for minimum employees required for work duration.

Penalty 5: Try and obey this constraint if possible but it is not essential. The constraint is preferred but not critical and is given for the shifts.

### 6.3 Objective function

Penalty is incurred due the violations of hard constraints and soft constraints, and the objective function is the sum of all the penalties. The penalty for each constraint is calculated linearly using the violation measurement factors. The violation measurement factor is the number of days or the number of nurses which violates the constraints.

The linear penalty function is calculated as: The violation measurement factor multiplied by the weight. The objective of this problem is to minimize the total penalty which means to maximize the requirements of nurses. According to the demand schedules that are provided, the optimized schedules will be designed using genetic algorithm in this paper.

### 6.3. Preference Parameter used for NSP

The objective of NSP was to minimize the cost involved and maximize the requirement of nurses. The requirement of nurses in this problem will depend on the preferences generated between the minimum and maximum limit for solving the nurse scheduling problem. As explained earlier about the objectives, the aim was to maximize the value generated from the preferences involved in NSP. For getting the maximum value for the preferences of the nurses, the preference matrix was generated in the GUI for individual nurses for the particular days for getting the preferred shift. The matrix generated for the nurses are in the manner of  $(m*n)$ .

Where,  $m$  = number of days in a week (7days)

$n$  = number of shifts for a particular day (5 shifts)

So, a matrix was generated by considering the values of  $(m*n)$  and were known as preference matrix for the nurses. The preference matrix would play a vital role for getting the optimized schedule for the nurses in health care systems. As schedule was being prepared for 4 weeks, the preference matrix for individual nurses would vary accordingly for four different weeks. By generating this kind of preference matrix for the nurses the chances for getting same shift for consecutive week should be reduced and there would be less chances for repetition. But, as explained earlier GA is a heuristic method so the solution generated is of trial and error nature. So at some point of time there may be a chance of repetition of the shift at a particular generation. GA involves the search space at random hence the values generated for preference are also random.

In order to maximize the requirement of nurses the values involved for preferences should be maximum. This can be analysed by making the summation of preference values for the entire four week schedule involved for all the nurses in hospital.

### 6.4. Result

The application of Genetic Algorithm gives a good Result most of the time. By considering 30 nurses, schedule was prepared for assigning the nurses in 5 shifts for 4 weeks. The program was runned for 4-5 times using Population Size=100, Number of Generations=500. The algorithm gets terminated after 101 generations. As the algorithm was runned for 4-5 times, the preference value varies every time of run. Hence, for 1<sup>st</sup> run the preference value was found to be 4178440, similarly for 2<sup>nd</sup>, 3<sup>rd</sup>, 4<sup>th</sup> and 5<sup>th</sup> run the value of preferences was found to be as 4204760, 4226960, 4218620 and 4197380 respectively.

Hence from the above preference values it was found that for 3<sup>rd</sup> run, the value of preference was maximum i.e. 4226960 and hence it would give the best optimal solution with maximum preference accommodated by the nurses and schedule was generated for that respective preference value.

The following table shows the optimized schedule for staff/nurses for 4 weeks using maximum preference value generated by Genetic Algorithm.

| OPTIMIZED NURSE/STAFF SCHEDULE |                          |     |     |     |      |     |     |     |
|--------------------------------|--------------------------|-----|-----|-----|------|-----|-----|-----|
| Nurse 1                        |                          | Mon | Tue | Wed | Thur | Fri | Sat | Sun |
|                                | For 1 <sup>st</sup> week | 5   | 5   | 5   | 0    | 0   | 5   | 5   |
|                                | For 2 <sup>nd</sup> week | 2   | 2   | 0   | 0    | 2   | 2   | 2   |
|                                | For 3 <sup>rd</sup> week | 0   | 0   | 4   | 4    | 4   | 4   | 4   |
|                                | For 4 <sup>th</sup> week | 3   | 3   | 3   | 0    | 0   | 3   | 3   |
| Nurse 2                        | For 1 <sup>st</sup> week | 0   | 0   | 3   | 3    | 3   | 3   | 3   |
|                                | For 2 <sup>nd</sup> week | 0   | 0   | 5   | 5    | 5   | 5   | 5   |
|                                | For 3 <sup>rd</sup> week | 5   | 1   | 0   | 0    | 1   | 1   | 1   |
|                                | For 4 <sup>th</sup> week | 0   | 0   | 2   | 2    | 2   | 2   | 2   |
| Nurse 3                        | For 1 <sup>st</sup> week | 2   | 2   | 2   | 2    | 0   | 0   | 2   |
|                                | For 2 <sup>nd</sup> week | 4   | 0   | 0   | 4    | 4   | 4   | 4   |
|                                | For 3 <sup>rd</sup> week | 3   | 3   | 3   | 3    | 0   | 0   | 3   |
|                                | For 4 <sup>th</sup> week | 5   | 5   | 5   | 0    | 0   | 5   | 5   |
| Nurse 4                        | For 1 <sup>st</sup> week | 0   | 0   | 5   | 5    | 5   | 5   | 5   |
|                                | For 2 <sup>nd</sup> week | 5   | 1   | 1   | 0    | 0   | 1   | 1   |
|                                | For 3 <sup>rd</sup> week | 2   | 2   | 2   | 2    | 0   | 0   | 2   |
|                                | For 4 <sup>th</sup> week | 3   | 3   | 3   | 3    | 0   | 0   | 3   |
| Nurse 5                        | For 1 <sup>st</sup> week | 3   | 3   | 0   | 0    | 3   | 3   | 3   |
|                                | For 2 <sup>nd</sup> week | 0   | 0   | 5   | 5    | 5   | 5   | 5   |
|                                | For 3 <sup>rd</sup> week | 4   | 4   | 4   | 0    | 0   | 4   | 4   |
|                                | For 4 <sup>th</sup> week | 4   | 2   | 0   | 0    | 2   | 2   | 2   |
| Nurse 6                        | For 1 <sup>st</sup> week | 2   | 2   | 2   | 2    | 0   | 0   | 2   |
|                                | For 2 <sup>nd</sup> week | 3   | 3   | 3   | 0    | 0   | 3   | 3   |
|                                | For 3 <sup>rd</sup> week | 0   | 0   | 5   | 5    | 5   | 5   | 5   |
|                                | For 4 <sup>th</sup> week | 4   | 4   | 4   | 4    | 0   | 0   | 4   |
| Nurse 7                        | For 1 <sup>st</sup> week | 4   | 0   | 0   | 4    | 4   | 4   | 4   |
|                                | For 2 <sup>nd</sup> week | 2   | 0   | 0   | 2    | 2   | 2   | 2   |
|                                | For 3 <sup>rd</sup> week | 1   | 1   | 0   | 0    | 1   | 1   | 1   |
|                                | For 4 <sup>th</sup> week | 3   | 3   | 0   | 0    | 3   | 3   | 3   |
| Nurse 8                        | For 1 <sup>st</sup> week | 3   | 3   | 0   | 0    | 3   | 3   | 3   |
|                                | For 2 <sup>nd</sup> week | 4   | 4   | 4   | 4    | 0   | 0   | 4   |
|                                | For 3 <sup>rd</sup> week | 5   | 0   | 0   | 5    | 5   | 5   | 5   |
|                                | For 4 <sup>th</sup> week | 0   | 0   | 1   | 1    | 1   | 1   | 1   |
| Nurse 9                        | For 1 <sup>st</sup> week | 0   | 0   | 1   | 1    | 1   | 1   | 1   |
|                                | For 2 <sup>nd</sup> week | 2   | 0   | 0   | 2    | 2   | 2   | 2   |
|                                | For 3 <sup>rd</sup> week | 3   | 3   | 0   | 0    | 3   | 3   | 3   |
|                                | For 4 <sup>th</sup> week | 0   | 0   | 5   | 5    | 5   | 5   | 5   |
| OPTIMIZED NURSE SCHEDULE       |                          |     |     |     |      |     |     |     |
| Nurse 10                       |                          | Mon | Tue | Wed | Thur | Fri | Sat | Sun |
|                                | For 1 <sup>st</sup> week | 5   | 0   | 0   | 5    | 5   | 5   | 5   |
|                                | For 2 <sup>nd</sup> week | 4   | 4   | 0   | 0    | 4   | 4   | 4   |
|                                | For 3 <sup>rd</sup> week | 4   | 2   | 2   | 0    | 0   | 2   | 2   |
| For 4 <sup>th</sup> week       | 0                        | 0   | 3   | 3   | 3    | 3   | 3   |     |
| Nurse 11                       | For 1 <sup>st</sup> week | 3   | 0   | 0   | 3    | 3   | 3   | 3   |
|                                | For 2 <sup>nd</sup> week | 5   | 5   | 5   | 0    | 0   | 5   | 5   |

|          |                          |   |   |   |   |   |   |   |
|----------|--------------------------|---|---|---|---|---|---|---|
|          | For 3 <sup>rd</sup> week | 4 | 4 | 0 | 0 | 4 | 4 | 4 |
|          | For 4 <sup>th</sup> week | 2 | 2 | 2 | 0 | 0 | 2 | 2 |
| Nurse 12 | For 1 <sup>st</sup> week | 2 | 0 | 0 | 2 | 2 | 2 | 2 |
|          | For 2 <sup>nd</sup> week | 0 | 0 | 3 | 3 | 3 | 3 | 3 |
|          | For 3 <sup>rd</sup> week | 5 | 5 | 0 | 0 | 5 | 5 | 5 |
|          | For 4 <sup>th</sup> week | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| Nurse 13 | For 1 <sup>st</sup> week | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
|          | For 2 <sup>nd</sup> week | 0 | 0 | 4 | 4 | 4 | 4 | 4 |
|          | For 3 <sup>rd</sup> week | 0 | 0 | 2 | 2 | 2 | 2 | 2 |
|          | For 4 <sup>th</sup> week | 3 | 3 | 0 | 0 | 3 | 3 | 3 |
| Nurse 14 | For 1 <sup>st</sup> week | 3 | 3 | 3 | 0 | 0 | 3 | 3 |
|          | For 2 <sup>nd</sup> week | 5 | 5 | 0 | 0 | 5 | 5 | 5 |
|          | For 3 <sup>rd</sup> week | 4 | 4 | 4 | 4 | 0 | 0 | 4 |
|          | For 4 <sup>th</sup> week | 0 | 0 | 2 | 2 | 2 | 2 | 2 |
| Nurse 15 | For 1 <sup>st</sup> week | 2 | 2 | 2 | 2 | 0 | 0 | 2 |
|          | For 2 <sup>nd</sup> week | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
|          | For 3 <sup>rd</sup> week | 5 | 5 | 5 | 5 | 0 | 0 | 5 |
|          | For 4 <sup>th</sup> week | 4 | 4 | 4 | 0 | 0 | 4 | 4 |
| Nurse 16 | For 1 <sup>st</sup> week | 4 | 4 | 4 | 4 | 0 | 0 | 4 |
|          | For 2 <sup>nd</sup> week | 3 | 3 | 3 | 3 | 0 | 0 | 3 |
|          | For 3 <sup>rd</sup> week | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
|          | For 4 <sup>th</sup> week | 2 | 2 | 2 | 2 | 0 | 0 | 2 |
| Nurse 17 | For 1 <sup>st</sup> week | 0 | 0 | 2 | 2 | 2 | 2 | 2 |
|          | For 2 <sup>nd</sup> week | 0 | 0 | 4 | 4 | 4 | 4 | 4 |
|          | For 3 <sup>rd</sup> week | 3 | 3 | 3 | 3 | 0 | 0 | 3 |
|          | For 4 <sup>th</sup> week | 3 | 1 | 0 | 0 | 1 | 1 | 1 |
| Nurse 18 | For 1 <sup>st</sup> week | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
|          | For 2 <sup>nd</sup> week | 0 | 0 | 5 | 5 | 5 | 5 | 5 |
|          | For 3 <sup>rd</sup> week | 5 | 2 | 2 | 0 | 0 | 2 | 2 |
|          | For 4 <sup>th</sup> week | 0 | 0 | 4 | 4 | 4 | 4 | 4 |
| Nurse 19 | For 1 <sup>st</sup> week | 4 | 4 | 4 | 4 | 0 | 0 | 4 |
|          | For 2 <sup>nd</sup> week | 3 | 3 | 3 | 3 | 0 | 0 | 3 |
|          | For 3 <sup>rd</sup> week | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
|          | For 4 <sup>th</sup> week | 2 | 2 | 2 | 0 | 0 | 2 | 2 |
| Nurse 20 | For 1 <sup>st</sup> week | 2 | 2 | 2 | 0 | 0 | 2 | 2 |
|          | For 2 <sup>nd</sup> week | 4 | 4 | 4 | 0 | 0 | 4 | 4 |
|          | For 3 <sup>rd</sup> week | 3 | 3 | 3 | 3 | 0 | 0 | 3 |
|          | For 4 <sup>th</sup> week | 5 | 0 | 0 | 5 | 5 | 5 | 5 |
| Nurse 21 | For 1 <sup>st</sup> week | 5 | 5 | 0 | 0 | 5 | 5 | 5 |
|          | For 2 <sup>nd</sup> week | 0 | 0 | 2 | 2 | 2 | 2 | 2 |
|          | For 3 <sup>rd</sup> week | 0 | 0 | 4 | 4 | 4 | 4 | 4 |
|          | For 4 <sup>th</sup> week | 3 | 3 | 0 | 0 | 3 | 3 | 3 |
| Nurse 22 | For 1 <sup>st</sup> week | 3 | 3 | 0 | 0 | 3 | 3 | 3 |
|          | For 2 <sup>nd</sup> week | 5 | 0 | 0 | 5 | 5 | 5 | 5 |
|          | For 3 <sup>rd</sup> week | 5 | 1 | 1 | 0 | 0 | 1 | 1 |
|          | For 4 <sup>th</sup> week | 2 | 2 | 0 | 0 | 2 | 2 | 2 |

|          |                          |   |   |   |   |   |   |   |
|----------|--------------------------|---|---|---|---|---|---|---|
| Nurse 23 | For 1 <sup>st</sup> week | 2 | 2 | 0 | 0 | 2 | 2 | 2 |
|          | For 2 <sup>nd</sup> week | 3 | 3 | 3 | 0 | 0 | 3 | 3 |
|          | For 3 <sup>rd</sup> week | 4 | 0 | 0 | 4 | 4 | 4 | 4 |
|          | For 4 <sup>th</sup> week | 5 | 5 | 0 | 0 | 5 | 5 | 5 |
| Nurse 24 | For 1 <sup>st</sup> week | 0 | 0 | 5 | 5 | 5 | 5 | 5 |
|          | For 2 <sup>nd</sup> week | 2 | 0 | 0 | 2 | 2 | 2 | 2 |
|          | For 3 <sup>rd</sup> week | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
|          | For 4 <sup>th</sup> week | 4 | 4 | 0 | 0 | 4 | 4 | 4 |
| Nurse 25 | For 1 <sup>st</sup> week | 4 | 0 | 0 | 4 | 4 | 4 | 4 |
|          | For 2 <sup>nd</sup> week | 5 | 5 | 0 | 0 | 5 | 5 | 5 |
|          | For 3 <sup>rd</sup> week | 5 | 3 | 3 | 3 | 0 | 0 | 3 |
|          | For 4 <sup>th</sup> week | 2 | 2 | 0 | 0 | 2 | 2 | 2 |
| Nurse 26 | For 1 <sup>st</sup> week | 2 | 2 | 2 | 0 | 0 | 2 | 2 |
|          | For 2 <sup>nd</sup> week | 4 | 4 | 0 | 0 | 4 | 4 | 4 |
|          | For 3 <sup>rd</sup> week | 5 | 0 | 0 | 5 | 5 | 5 | 5 |
|          | For 4 <sup>th</sup> week | 0 | 0 | 3 | 3 | 3 | 3 | 3 |
| Nurse 27 | For 1 <sup>st</sup> week | 3 | 3 | 0 | 0 | 3 | 3 | 3 |
|          | For 2 <sup>nd</sup> week | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
|          | For 3 <sup>rd</sup> week | 4 | 0 | 0 | 4 | 4 | 4 | 4 |
|          | For 4 <sup>th</sup> week | 5 | 5 | 5 | 0 | 0 | 5 | 5 |
| Nurse 28 | For 1 <sup>st</sup> week | 5 | 5 | 5 | 0 | 0 | 5 | 5 |
|          | For 2 <sup>nd</sup> week | 0 | 0 | 3 | 3 | 3 | 3 | 3 |
|          | For 3 <sup>rd</sup> week | 2 | 0 | 0 | 2 | 2 | 2 | 2 |
|          | For 4 <sup>th</sup> week | 4 | 0 | 0 | 4 | 4 | 4 | 4 |
| Nurse 29 | For 1 <sup>st</sup> week | 4 | 4 | 4 | 4 | 0 | 0 | 4 |
|          | For 2 <sup>nd</sup> week | 5 | 5 | 0 | 0 | 5 | 5 | 5 |
|          | For 3 <sup>rd</sup> week | 1 | 0 | 0 | 1 | 1 | 1 | 1 |
|          | For 4 <sup>th</sup> week | 2 | 0 | 0 | 2 | 2 | 2 | 2 |
| Nurse 30 | For 1 <sup>st</sup> week | 2 | 2 | 0 | 0 | 2 | 2 | 2 |
|          | For 2 <sup>nd</sup> week | 4 | 0 | 0 | 4 | 4 | 4 | 4 |
|          | For 3 <sup>rd</sup> week | 0 | 0 | 5 | 5 | 5 | 5 | 5 |
|          | For 4 <sup>th</sup> week | 3 | 3 | 3 | 0 | 0 | 3 | 3 |

Table 6.4. Optimized 4 week nurse schedule obtained for maximum Preference value

### 6.5. Graphical Representation

Fig. 6.5. Represents the graphical approach of Genetic Algorithm while the program was runned and got terminated. It shows the best and mean of fitness value obtained from new generation of individuals while performing crossover and mutation process for GA.

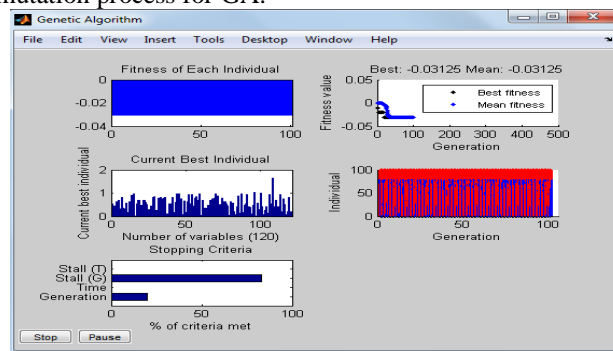


Figure 6.5 Representation of Genetic algorithm (Individual Fitness and Best value)

## 6.6. Conclusion

This paper focuses on solving a nurse scheduling problem in health care systems. This paper uses Genetic Algorithm in MAT Lab using GUI to solve this type of problem considering all the hard constraints and soft constraints as mentioned earlier for getting the best optimal solution in form of optimized schedule for staff. As explained earlier the schedule for 4 weeks was provided by considering the maximum preference value “4226960” as the best solution for GA. The feasibility of the method of Genetic Algorithm was tested by applying it to this type of nurse scheduling problem and considering that all the constraints are satisfied. In this, the data was made of static nature without making any changes in the data and output was obtained as per the requirement for staff scheduling.

Accordingly after running the software the output was shown in the prescribed format. As an advantage of genetic algorithm, it can be easily modified for different problems and can handle wider solution space. Also it can perform very well for large-scale and can also be employed for a wide variety of optimization problems. If GA was need to be compared with the work that other researchers have done. While considering about limitations of GA, as explained earlier it is the heuristic method and are based on randomness so results obtained are of trial and error nature so to get accuracy always in results would be difficult.

The present work was found different as preference matrix was considered for the present work which was earlier not considered by the others. The preference section was not taken into account for the research areas for staff scheduling in hospitals or in some other work domain. As in this work the gap between the researches was filled and the preference parameter was taken into account for further coding of genetic algorithm. Such that the scheduling of nurses in hospitals was obtain by using the preference matrix which played a vital role for staff scheduling.

## 6.7. Future Work

The future work for this study would be as mentioned previously, many other algorithms like Ant Colony Optimization (ACO), simulated annealing, tabu search and other forms of local search are appropriate for solving this kind of problems. This problem can be used to test other algorithms and compare with Genetic Algorithms. Also same methodology can be applied for other staff scheduling problems like assignment, transportation, job shop scheduling problem etc.

## References

- [1] A.T. Ernst, H. Jiang, M. Krishnamoorthy and D. Sier, *Staff scheduling and rostering: A review of applications, methods and models* European Journal of Operational Research 153 (2004)3–27
- [2] Amol C. Adamuthe and Rajankumar Bichkar, *Hybrid Genetic Algorithmic Approaches for Personnel Timetabling and Scheduling Problems in Healthcare* International Conference on Technology Systems and Management (ICTSM) 2011 Proceedings published by International Journal of Computer Applications® (IJCA)
- [3] John S. Dean, *Staff Scheduling by a Genetic Algorithm with a Two Dimensional Chromosome* Structure Park University, Information and Computer Science Department.
- [4] Uwe Aickelin, Kathryn A. Dowsland *An Indirect Genetic Algorithm for a Nurse Scheduling Problem* Computers & Operations Research, 31(5), pp 761-778, 2003
- [5] Tiago M. Dias, Daniel F. Ferber, Cid C. de Souza and Arnaldo V. Moura, *Constructing nurse schedules at large hospital* International Transaction in Operational Research 10 (2003) 245–265
- [6] Ivo Blochliher, *Modelling staff scheduling problems* A tutorial European Journal of Operational Research 158 (2004) 533–542
- [7] Uwe Aickelin, Kathryn A. Dowsland *Exploiting problem structure in a genetic algorithm approach to a nurse rostering problem.* Journal of Scheduling, 3 (3): 139-153, 2000.
- [8] Edmund K. Burke et al. *The State of The Art of Nurse Rostering* Journal of Scheduling 7: 441–499, 2004. ©2004 Kluwer Academic Publishers. Printed in the Netherlands.
- [9] William H. Hsu, *Genetic Algorithms* Department of Computing and Information Sciences Kansas State University.
- [10] S.N.Sivanandadam and S.N. Deepa, *Introduction to Genetic Algorithms.*
- [11] B. Cheang , H.Li, A. Lim, B. Rodrigues, *Nurse Rostering problems—a bibliographic survey* European Journal of Operational Research 151 (2003) 447–460
- [12] Monica Fira, Adrian Brezulanu, and Lucian Fira, *A genetic algorithm approach for a constrained employee scheduling problem as applied to employees at mall type shops* International Journal of Advanced Science and Technology Vol. 14, January, 2010

### Books:

- S. N. Sivanandam and S.N Deepa, *Introduction to Genetic Algorithm* (Publication Springer Berlin Heidelberg New York)

### Thesis:

- Mengtao Ji, *A Scheduling Optimization Problem Applied To Nurses Using Genetic Algorithm* Gerald Schwartz school of business and information systems, St. Francis Xavier University Antigonish, Nova Scotia March 30. 2012