Abstract No: 025-0701

Title: A Genetic Algorithm Approach for Nurse Scheduling in an Operating suite

Elham Torabi

Department of Operations, Business Analytics and Information Systems

534 Carl H. Lindner College of Business

University of Cincinnati

2925 Campus Green Dr. Cincinnati OH 45221

torabiem@mail.uc.edu, (513) 288-2546

Gino J. Lim

Department of Industrial Engineering

University of Houston

E211 Engineering Building 2, Houston, TX 77204

ginolim@uh.edu, (713) 743-4194

Craig M. Froehle

Department of Operations, Business Analytics and Information Systems

521 Carl H. Lindner College of Business

University of Cincinnati

2925 Campus Green Dr. Cincinnati OH 45221

craig.froehle@uc.edu , (513) 556-7174

POMS 23rd Annual Conference
Chicago, Illinois, U.S.A.
April 20 to April 23, 2011

A Genetic Algorithm Approach for Nurse Assignment Problem in an Operating suite

Abstract

In this paper we present a Genetic Algorithm approach for scheduling operating room (OR) nurses. Most studies in operating room scheduling deal with patient flow analysis and physician scheduling, limited literature has focused on scheduling OR nurses. Our objective is to minimize nurses' idle time, overtime and non-consecutive assignments during overtime hours while maximizing demand satisfaction. The major constraints are: 1) shift constraints and 2) match between nurses' skill sets and surgery requirements. Due to the large size of the problem, finding an optimal solution is extremely difficult. Therefore, a Genetic Algorithms approach is proposed to find a set of good schedules in a reasonable amount of time. The solutions were then evaluated in a simulation model with probabilistic surgery durations. The best performing solution in the stochastic environment is selected as the final schedule. We present a set of case studies to demonstrate the performance of our approach. We used sample data gathered from a large urban teaching hospital that has 31 operating rooms.

Background

Healthcare centers are facing serious resource shortages and high costs which lead to higher medical and insurance costs to patients and a negative effect on the survival rate. Recent studies show that the shortage of registered nurses may reach 500,000 by year 2025 (Texas Hospital Nurse Staffing Survey 2006). Given the fact that nurses make up the majority of hospital staff, the nurse shortage situation has long been a main concern for health care centers. Although a sufficiently long term plan for educating more nurses is needed to meet the demand for nurses in such magnitude, optimal nurse utilization may be a useful approach for health centers to overcome the shortages and to reduce the costs. The literature shows that most of the efforts have been made to optimize patient scheduling, physician scheduling and operating room scheduling while not enough attention has been paid to nurse assignment which obviously is of high importance due to the relatively large number of nurses in any health care system. Thus, the main focus of this study is how to optimally utilize nurses in an operating suite.

Literature review

Researchers have been tried different ways to improve the operating suite's performance. The objectives are mainly to reduce overtimes, idle times, costs, patient waiting times and improving utilization of the system. Cardoen et al. (Cardoen 2010) published the latest literature review on operating room planning and scheduling, covering more than 100 papers in the literature. The main focus of almost all papers is on operating room scheduling or physician block scheduling and none is considering nurse utilization and scheduling. Moreover, the systems considered were far too small in size compared to the operating suite which is under study in this research. This literature review has four parts: nurse assignment in clinics, OR scheduling using mathematical modeling, OR scheduling using simulation, scheduling using simulation-based optimization methods and applications of GA.

Tsai and Li (Tsai 2009) developed a two-stage model for nurse scheduling problem using Genetic Algorithm in both stages to find near optimal solutions. Studies like Berrada et al. (Berrada 1996), Dowsland (Dowsland 1998), Maenhout and Vanhouke (Maenhout 2007) have developed mathematical models and used heuristic approaches such as Tabu Search to improve the objectives. Kawanaka et al. (Kawanaka 2001) developed a Genetic Algorithm for solving the nurse scheduling problem with constraints. When solving a constrained problem with a Genetic Algorithm there is the possibility of generating infeasible solutions. In this case a repair procedure is needed to fix the infeasibility of solutions.

There has been significant work on improving OR schedules. Denton et al. (Denton 2007) developed a two stage stochastic optimization model for scheduling a single OR. They used surgery duration variance in order to reduce the costs of OR team waiting time, OR idle time and overtime. As the two stage stochastic problem is NP-hard, they have developed three heuristic algorithms to approximate the optimal solution. In Gupta (Gupta 2007) three common problems are modeled mathematically and solutions are suggested: Elective Surgery Capacity Allocation, Elective Surgery Booking model and Elective Surgery Sequencing.

Using simulation models to evaluate different strategies for scheduling multiple operating rooms are very common. Tyler et al. (Tyler 2003) tried to determine important factors in optimal utilization of operating rooms. Ramis et al. (Tyler 2001) simulated an ambulatory surgery center that has only two operating rooms. Sepulveda et al. (Sepulveda 1999) simulated the operating rooms process flow from the reception desk to the checkout desk. Ballard and Kuhl (Ballard

2006) modeled an operating suite with four operating rooms. Their goal was to determine the actual capacity of the OR suite. They used this model to evaluate changing in resource levels or the surgical procedures. Arnaout and Kulbashian (Arnaout 2008) modeled operating room scheduling as a parallel machine scheduling problem. Several heuristic methods are developed and compared in a simulation model to find the most appropriate method.

Simulation-based optimization is an approach that, until recently, was not widely used in healthcare. Denton et al. (Denton 2006) have utilized simulation for an outpatient endoscopy operating suite to study the impact of uncertainty on the system performance. They use an optimization model that applies Monte Carlo simulation and simulated annealing to evaluate schedules. Lamiri et al. (Lamiri 2008) consider both elective and emergency surgeries in planning operating rooms and developed a stochastic model by combining Monte Carlo simulation and Mixed Integer Programming. Persson et al. (Persson 2006) have utilized a simulation-based optimization approach to deal with the multi-objective scheduling problem in a Swedish postal service. None of these papers have mentioned the computation time for their approach, but the significant computation and computer time required by these techniques make them difficult to implement for problems encountered in practice.

Genetic algorithms are used in a wide range of optimization problems and in many cases are modified according to the nature of the problem at hand. Genetic Algorithm can be used to obtain a good solution in a reasonable time. Cochran et al. (Cochran 2003) have introduced a multi-population Genetic Algorithm for solving multi-objective scheduling problems for parallel machines to minimize make span and total tardiness. Paul and Chanev (Paul 1998) have used Genetic Algorithm in a simulation-based optimization approach for Steelworks problem.

There are several interesting points in this literature. Improving the surgery schedules can have a considerable impact on the overall performance of the operating suite. Overtime and idle time are the most mentioned measures of performance in the literature. Kawanaka et al. (Kawanaka 2001) have developed a Genetic Algorithm to solve a constrained nurse scheduling problem. Although it is for nurse scheduling in a clinic, their method for generating feasible solutions helped us in developing GA for nurse scheduling problem in an operating suite that has many constraints sets.

System Definition

The system under study is an operating suite with 31 operating rooms each dedicated for surgeries of certain types. The rooms are equipped according to the requirements of surgeries that are performed in them.

The surgeries are categorized into 17 main medical specialties, such as Urology, Cardio, Neuro, etc. Within each medical type of surgery there is another classification: a surgery may be complex, moderate or simple. Each surgery case demands at least one of each type of nurses, one Registered Nurse (RN) and one Scrub Technician (Scrub tech). The specialty and competency level of the nurses that are assigned to a surgery case must match the surgery specialty and complexity.

The duration of cases varies from one type of surgery to another and also depends on the complexity of the case. The surgery cases are scheduled and assigned to ORs based on the availability of ORs and the estimated duration of the surgery which is determined by the surgeon. However, the surgery may take longer or shorter than the estimated duration. Comparing observed case duration data with the parametrically estimated values shows that, in general, the estimates can be far from the real value for the surgery duration. Of course, a poorly designed schedule due to inaccurate estimates will lead to underutilization or overutilization of staff and overtime of the OR suite. In this study we try to include the stochastic nature of case durations in assignment of nurses to operating rooms.

The operating suite has an average of 150 nurses, having 60 nurses available in any given day. There are two types of nurses RNs and Scrub Technicians. Some of the RNs have the skill of doing scrub but the opposite is not true. Nurses also have medical specialties. Each nurse can work on a surgery case that matches his/her specialty. Within each specialty, nurses also have different competency levels that means, based on their skills and experience, they can work on a simple, moderate or complex surgery case. The ORs start working at 6:30 a.m. and officially finishes at 11:00 p.m. There are five different shift patterns in a day, some of them are eight-hour and others are ten-hour shifts with different starting times.

Problem Statement

The nurse assignment problem in the operating suite is very difficult to solve and large problem instances often result in unreasonably long solution times. In a given day, in the operating suite, there may be up to 70 cases and 150 nurses. It can be difficult to find a good schedule in a reasonable amount of time. A schedule should specify that a given nurse is assigned to which OR at what time and what job is to be performed by the nurse.

Based upon the deterministic nurse assignment optimization model developed in Mobasher et al. (Mobasher 2011), the Nurse Assignment Model (NAM) described in this paper is a multi-objective optimization model. The objectives are minimizing demand dissatisfaction, nurse idle time and overtime. The available nurses are assigned to surgery cases that are scheduled for that day in the surgical suite. This assignment is based on the requirements of the surgery case such as specialty, complexity, number of registered nurses (RNs) and scrub technicians (Techs) that are needed for the surgery cases that are scheduled to be performed in a day. The duration of the surgery is estimated by the surgeons.

We make the following assumptions to develop the NAM. Working days are divided to 30-minute time intervals. There are eight-hour, ten-hour and twelve-hour regular shifts that include regular hours and authorized overtime hours. We assume that when a nurse is assigned to a case, he/she stays there till the case is done. A nurse with a higher competency level can also be assigned to less complex surgeries. Minimum requirements for surgeries are considered. That means only one RN and one Scrub Tech are assigned to each surgery case. This assumption is not misleading because this is the worst case and if there is any nurse remaining the nurse manager can assign extra nurses if it is needed. The decision variable is a three dimensional binary variable which has the value of one, if nurse *i* is assigned to case *c* to do role *k*. The Objective is to minimize total idle time, total overtime and demand dissatisfaction. Tables 1 and 2 show the notation used in the optimization model for sets and parameters respectively.

Table 1. Notation of sets used in Nurse Assignment Model

Notation	Description
I	Set of available nurses
K	Set of roles that are required for each surgery case (k=1 RN, k=2 Tech)
Q	Set of specialties
C	Set of cases scheduled for surgery in each day
S	Set of available shifts
P	Set of competency/complexity levels (1: Simple, 2: Moderate, 3:
H	Time intervals in a working day

Table 2. Parameters used in Nurse Assignment Model

Notation	Description
P_{ikpq}^{1}	If nurse i can do role k with competency level p in specialty q
P_{cqp}^{2}	If case c needs specialty q, complexity p and is in progress in time
P_{ch}^{3}	If case c is in progress during time interval h
P_{sh}^{4}	If shift s has time interval h as regular working hours
P_{sh}^{5}	If shifts has time interval h as authorized overtime hours
overlap	Value 1 if two cases have an overlap in time; 0 otherwise
duration	Surgery case duration
nstart/nend	Start/ end of nurse working hours
Start/end	Start/ end of a surgery case
Nendreg	Ending time of a nurse’s regular shift (not in overtime)

We have two sets of constraints. Constraints (1)-(9) are hard constraints and constraints (10)-(12) are soft constraints. Hard constraints are those that must be satisfied in any feasible solution. On the other hand, soft constraints may not be completely satisfied and we want to satisfy them as much as possible. The hard constraints are as follows:

$$overlap_{c,c1} \cdot \left(\sum_k X_{ikc} + \sum_k X_{ikc1} \right) \le 1 \qquad \forall i \in I,\ c, c1 \in C \ and\ c \ne c1 \qquad (1)$$

$$\sum_c \sum_k X_{ikc} \cdot Duration_c \le 24 \qquad \forall i \in I \qquad (2)$$

$$\sum_k X_{ikc} \cdot nstart_i \le start_c \qquad \forall i \in I,\ c \in C \qquad (3)$$

$$\sum_k X_{ikc} \cdot end_c \le nend_i \qquad \forall i \in I,\ c \in C \qquad (4)$$

$$\sum_i X_{i2c} \ge 1 \qquad \forall c \in C \qquad (5)$$

$$\sum_i X_{i1c} \ge 1 \qquad \forall c \in C \qquad (6)$$

$$X_{i1c} + X_{i2c} \le 1 \qquad \forall i \in I,\ c \in C \qquad (7)$$

$$X_{ikc} \le \sum_p \sum_q \left(P_{cqp}^{2} \cdot P_{ikqp}^{1} \right) \qquad \forall i \in I,\ c \in C, k \in K, q \in Q, p \in P \qquad (8)$$

$$X_{ikc} \in \{0,1\} \qquad \forall i \in I,\ c \in C, k \in K \qquad (9)$$

Constraint set (1) indicates that if cases *c* & *c1* in set *C* have an overlap in time nurse *i* cannot be assigned to both cases. Constraint set (2) checks that total working hours of a nurse *i* is less than 12 hours for all nurses in set *I*. In this model we allow the nurses to work overtime in Constraint set (2) and we try to minimize it using a soft constraint that will be explained later. Constraint sets (3) and (4) take care of shift limitations. Nurse *i* is assigned to case *c* only if the whole surgery case is scheduled within the allowed 12 hours of her shift. Constraint set (5) assures that in any assignment nurse skill level, *p*, and specialty, *q*, which is indicated in P^{1}_{ikqp} matches with the case requirements P^{2}_{cqp}. Constraint set (6) does not allow the assignment of nurse *i* to do both jobs in *K* a surgery for all nurses and all cases. Finally, constraint sets (7) and (8) take care of minimum case requirements and ensure that there is one nurse with role *k = 1* as an RN and one nurse with role *k = 2* as a Tech assigned to each case *c*.

Our goal is to minimize the maximum of total idle time, total overtime and non-consecutive assignments. We use soft constraints to define these objectives and they are:

$$\sum_{i}\left(\sum_{k}\sum_{c} X_{ikc}\cdot\left(\sum_{s}\sum_{h} P^{3}_{ch}\cdot P^{5}_{sh}\right)\right)\leq \textit{total overtime} \qquad (10)$$

$$\sum_{i}\left(nend_{i}-nstart_{i}-\left(\sum_{k}\sum_{c} X_{ikc}\cdot\left(\sum_{s}\sum_{h} P^{3}_{ch}\cdot P^{4}_{sh}\right)\right)\right)\leq \textit{total idle time} \qquad (11)$$

$$\sum_{k}\sum_{c} X_{ikc}\cdot\left(\sum_{s}\sum_{h} P^{3}_{ch}\cdot P^{5}_{sh}\right)+NC_{i}=\left(\sum_{k} X_{ikc}\cdot P^{3}_{ch}\cdot P^{5}_{sh}\cdot end_{a}\right)-nendreg_{i} \qquad \forall \mathrm{i}\in \mathrm{I} \qquad (12)$$

Constraint (10) calculates the total overtime of nurses and is a summation of overtimes for each nurse. To calculate overtime for nurse *i* we have to sum all time interval *h* if nurse *i* is assigned to case *c* which is in progress in time interval *h* that is considered as overtime hour.

Constraint (11) is for calculating total idle time of nurse which is calculated as the total working hours of excluding the nurse's busy time. Then we add up the idle time of all nurses to get the total idle time. In constraints (12) we are trying to find the total time units in which nurses are assigned non-consecutively to overtime hours. This is denoted as *NC* for all nurses. Our objective is to minimize the summation of total idle time, total over time and non-consecutive assignments during overtime hours.

Proposed Methodology

A Genetic Algorithm is used to solve the deterministic nurse assignment problem in a reasonable time. But first, a simplified mixed integer model of the nurse assignment problem is developed and used to generate feasible solutions for the first generation of the Genetic Algorithm. In the deterministic model it is assumed that there are no call-ins and cancellations and the actual duration of cases are estimated by surgeons. By using a Genetic algorithm, a set of good solutions are found for the problem. The next step is to select one schedule from this set. In this step the solutions are evaluated using the simulation model to check which schedule works better in a stochastic environment. Figure 1 shows this methodology.

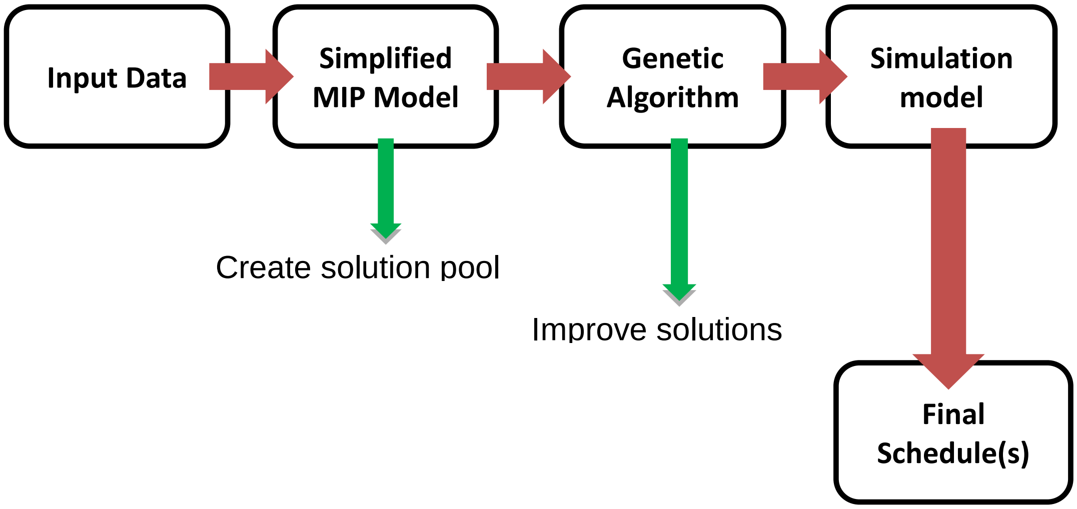

Figure 1. The methodology used for solving Nurse Assignment Problem

The Genetic Algorithm generates good feasible solutions for the problem. Also, a simulation model of the OR suite is developed using Arena simulation software using historical data obtained from the operating suite. After GA is run for the problem, the fittest solutions are run in Arena Process Analyzer under the same conditions to make a fair comparison among schedules. Finally, the solution that has a better performance in the stochastic environment will be selected as the final solution to the nurse scheduling problem.

A Genetic Algorithm for Nurse Assignment Problem

A Genetic Algorithm starts with a first population of solutions or chromosomes. The coding of chromosome representation depends on the problem that is to be solved. In conventional GA the chromosome is a string of binary codes. However, integer coding, real value coding and others have been used each for certain types of problems. In this study the solution format is a three dimensional binary matrix. The initial solution can be generated randomly or achieved from historical data. In this study the initial solutions are generated by a solution pool created in GAMS.

After generating the first population of solutions, a number of individuals are selected for reproduction. Reproduction is done under various genetic operators. Crossover is a genetic operator in which two chromosomes mate to generate two offspring. There are different crossover techniques in the literature such as one point crossover, multi point crossover, uniform crossover, etc. The number of individuals selected for crossover is known as the crossover rate. The crossover rate in the GA developed for the nurse scheduling problem in this study is 20% and has been achieved by experiments explained later. Mutation is another genetic operator that is used to maintain genetic diversity through the evolution of solutions. In fact, crossover takes the algorithm toward focusing on a certain part of the search space while better solutions may exist in another parts of the search space. Mutation helps the algorithm jump across the whole search space. The result may or may not be promising but it is necessary in order to avoid remaining in a local optimum.

When solving constrained optimization problems, the issue of feasibility arises. In GA, solutions generated by reproduction may not be feasible and need to be modified. In this study, we repair infeasible offspring as soon as they are generated. The reason is, if we want to fix the final solutions of GA, we need to check for all constraints and modify the solution. However, because of the large number of constraints, these modifications may lead to a solution that is feasible, but not good enough.

A one-point crossover rule is developed to generate solutions in which some sets of constraints are met. Then a repair procedure takes care of the remaining constraints to make sure that a feasible solution is generated. After the new solutions are generated the quality of the solutions is evaluated. This evaluation is done using a fitness function that assigns a score to each solution based on the objective(s) of the problem. If the problem has multiple objectives the fitness can be a weighted sum of the objectives. Among the solutions, those with better fitness values will be selected as the next generation for the algorithm. This procedure continues until there is no significant improvement; or after certain number of iterations.

We validate our GA code using small- and medium-size problem instances. As the decision variable in NAM has three dimensions, we have represented a solution with a three dimensional matrix. Three sets are defined: a) set of available RNs; b) set of available Techs, and; c) the overlap set. Set of available RNs for case c includes all nurses that can work as an RN and their specialty and competency match with the requirements of case c. Similarly, set of available Techs for case c includes all nurses that can work as a Tech and their specialty and competency match with the requirements of case c. The overlapping set for case c is a set of all cases that have a time overlap with case c; a nurse who is assigned to case c cannot be assigned to any case in its overlap set.

In crossover, a random integer number is generated between 1 and the number of surgery cases and the solution matrix is split into two parts. A child will inherit the first part of a parent and the other part of the second parent. In this way children satisfy constraints regarding case requirements and shift limitations and working on only one role in a case. But undesirable overtimes and assignment of nurses to multiple ORs at the same time may occur for one or more nurses. Therefore, a heuristic algorithm is developed for repairing infeasible solutions that are generated by crossover. The procedure is summarized below:

- Step 1: Check constraints 1 and 2 (simultaneous assignment and total working hours) for each nurse. If each constraint is violated for nurse *i* put $x_{ick} = 0$, where *c* corresponds to the shortest case that nurse *i* is assigned to. After both constraints are checked and fixed go to Step 2. Otherwise go to Step 3.
- Step 2: Update the sets of available RNs and Techs and go to Step 3.
- Step 3: Check case requirements for all cases; if the constraint is violated for case *c*, store it in the set of remaining cases and go to Step 4. Stop otherwise.
- Step 4: Loop Assign the available nurses to remaining cases. The priority is given to longer cases and nurses with smaller busy time. Update the sets of available RNs, available Techs and remaining cases after each assignment. After *n* iterations if no nurse is assigned to case *c*

remove it from the remaining cases (in evaluation a penalty will be assigned for demand dissatisfaction). Go to Step 5.

- Step 5: Check for total working hours of nurses. If violated for nurse *i* put $x_{ick} = 0$, where *c* corresponds to the shortest case that nurse *i* is assigned to and go to step 2. Stop otherwise.

After feasible children are generated and added to the population, all solutions are evaluated with the fitness function. The fitness function is the sum of total overtime, total idle time, and demand dissatisfaction. Demand dissatisfaction occurs when a scheduled case is not performed. If the duration of this case is *d* time units, we have failed to satisfy demand for *d* time units. Thus, the amount of demand dissatisfaction is measured as the amount of time that we failed to satisfy the demand and is equal to the duration of the case. In this way the units of all three objectives will be time and there is no need for normalizing the fitness function.

When the fitness is calculated for all solutions, solutions are sorted in ascending order of their fitness value. Then depending on the population size used in the algorithm, solutions with better fitness values are selected as the next population. If the population size is considered 100, the first hundred solutions are chosen as the next population. The algorithm is repeated until a stopping criterion is met. The measure of performance we use is the fitness value of the generated solutions and the computation time.

Experiments and Numerical results

First we made some trial runs to find a good set of algorithm parameters such as population size, number of crossovers and stopping criteria for GA. Then we try to validate our Genetic Algorithm by solving different problem instances. Finally, we evaluate best solutions generated by the GA using the simulation model developed for the system. A computer with dual-core 2.2 GHz processor and 3 GB RAM was used for computations.

The simplified MIP model was coded in GAMS (The General Algebraic Modeling System) and solved using CPLEX. The model was used for generating a pool of feasible solutions for our Genetic Algorithm, which was executed in Matlab.

For choosing the population size, several experiments were run for a simple example with different population sizes. Because the best fitness achieved in different runs of the algorithms with exactly the same parameters may still be different, we have used multiple runs of the GA for each problem. For each population size the algorithm was run multiple times and the average of best achieved fitness value is compared. Figure 2 shows a visual comparison of the experiments. It is observed that as the population size increases, the fitness value improves. However, the computation time dramatically increased for population sizes of 150 or higher. This suggests that there should be a tradeoff between the solution quality and the computation time when choosing an appropriate population size.

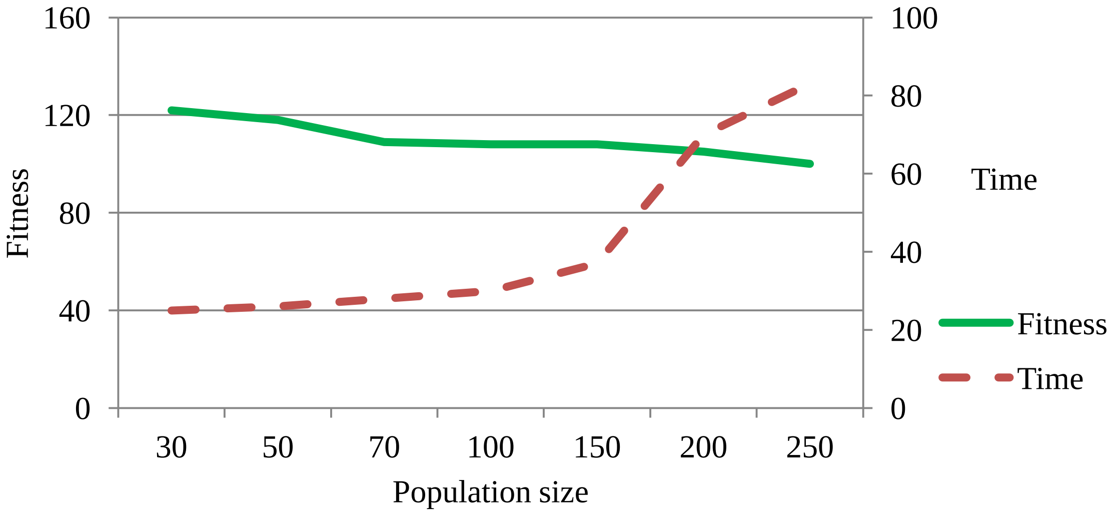

Figure 2. Effect of changing population size on fitness value for a medium size problem

Another set of experiments were run to choose the crossover rate. A higher crossover rate means generating more solutions and it leads to a higher computation time. Therefore, we consider this trade off in choosing the crossover rate. The comparison is also shown in Figure 3.

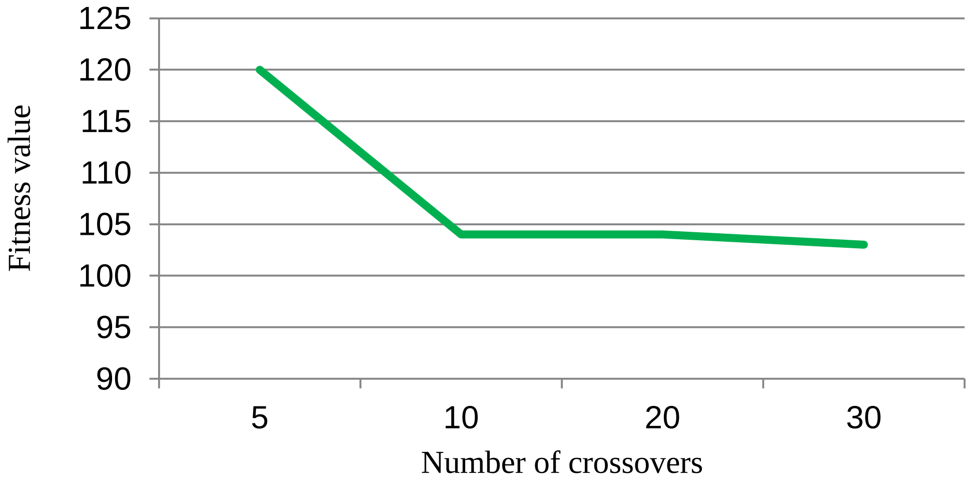

Figure 3. Fitness value vs. number of crossovers

The primary stopping criterion of Genetic Algorithm is when no significant improvement on the fitness value is observed after certain number of consecutive iterations. However, for large problems we used the second stopping criterion (the number of iterations) to save computation time.

For finding the proper number of iterations, a small problem was run for 120 iterations for several times and the quality and computation time was captured in different iterations. The results are compared in Figure 4. Although the experiments show that the algorithm converges within 30 iterations for this problem, we may not observe the same convergence for different problem instances. However, we do not recommend a large number of iterations either because of saving computation time which is one of our main goals of the proposed method. Our experiments show that the algorithm does not show much improvement after 30 iterations for small- to medium-size problems.

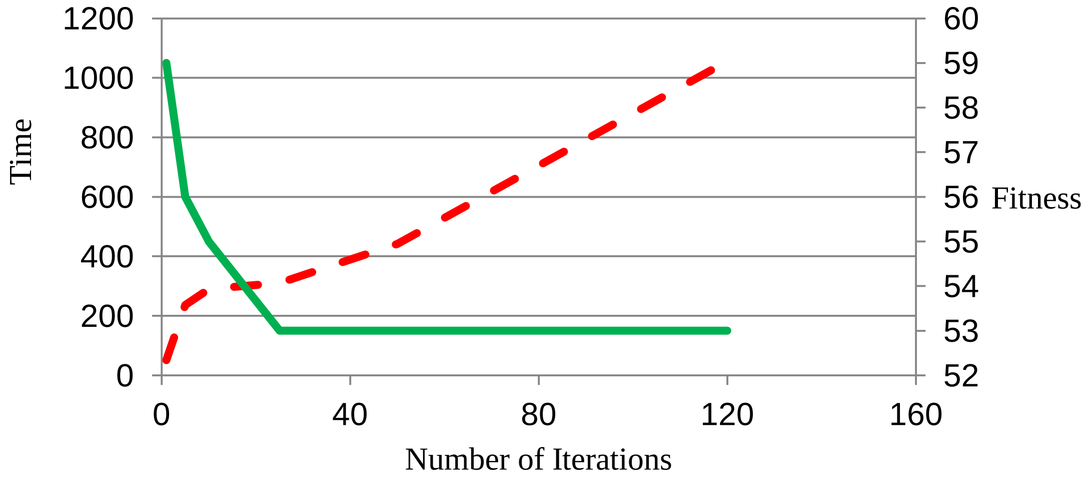

Figure 4. Improvement of fitness value with increasing the number of iterations

After identifying algorithm parameters, we validate the algorithm by solving different problem instances. Table 3 shows different problem instances that are used for this purpose. We have obtained two sets of historical data from the MD Anderson Cancer Center operating suite.

Table 3. List of problem instances

Problem	Num of nurses	Num of cases	Num of shifts	Num of specialties
1	20	14	1	1
2	28	37	2	5
3	28	37	2	5
4	35	37	3	5
5	53	115	6	17

The algorithm solves small problems in acceptable time, but the average solution gap for medium-sized problems is 30% compared with the exact approach. For small- and medium-size problems, we have the optimal solution for the Mixed Integer Program (MIP) models. For large problems, we were not able to benchmark our results of the GA because CPLEX fails to generate a feasible solution.

Figure 5 shows the result of GA for Problem 1 with 20 nurses and 14 cases. The algorithm converges to the global optimal solution within 10 minutes. We have also observed that the quality of the whole population improves with increased iterations.

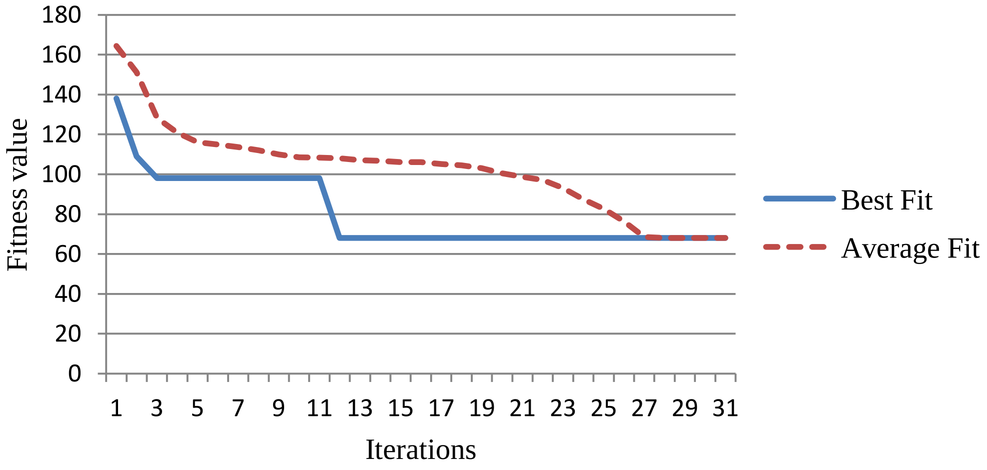

Figure 5. Result of GA for Problem 1

Problem 3 is the same as Problem 2 except that durations of some of the cases are changed. This is done for sensitivity analysis of the algorithm. The duration of shortest and longest cases were changed to a value closer to the mean of the historical date for that kind of surgery. It is observed that in this case, the performance of schedules is better in the sense that the solution quality is much better when we compare the results with the optimal solution. The average optimality gap was 20% for this problem.

Problem 4 is solved within 50 iterations with 25% to 30% gap from the optimal solution. The difference observed between two different runs is because of the random numbers generated in the crossover operation which leads to generate different solutions.

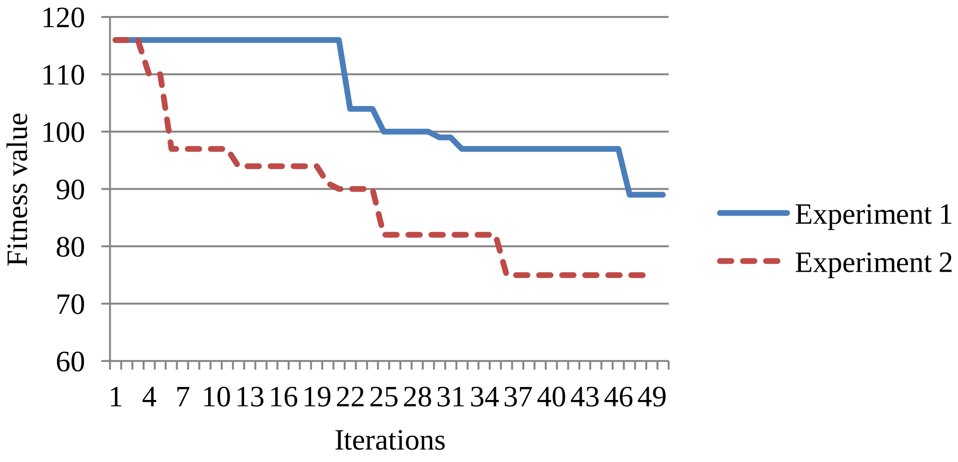

Figure 6. Results of GA for Problem 4

Figure 7 shows the result of GA for a medium size problem in different runs with different number of iterations. It is observed that the objective value has decreased over the iterations in all different runs. This figure shows that the algorithm is able to generate better solutions in different runs and the algorithm converges toward a certain point regardless of the randomness that exists in the nature of the algorithm.

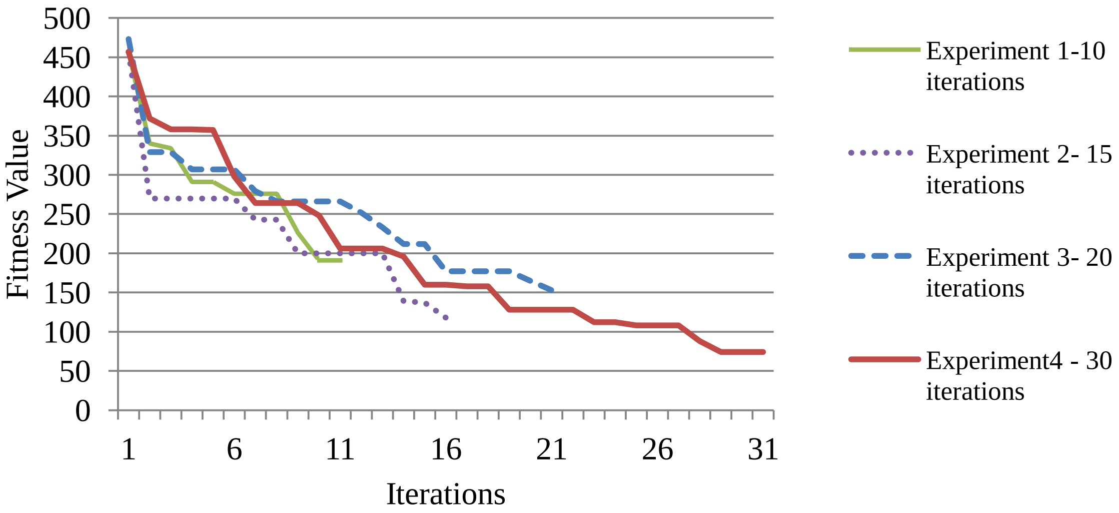

Figure 7. Comparisons among different runs for a medium-size problem

Evaluating the schedules under uncertainty

A simulation model of the OR was developed in which the duration of cases was defined by probability distributions that were extracted from historical data for each specialty. The best results of GA were run in the simulation model to be evaluated in a stochastic environment. The genetic algorithm returns a set of solutions with same or close fitness values. In this situation we need to choose one from ten or more good solutions. This decision is not easy to make. Therefore we run the top 10 solutions in the stochastic environment to have a measure for choosing the best available schedule. For this purpose the simulation model is developed and used to evaluate the solutions. The measures of performance are nurse utilization (as a representation of idle time), total overtime, and total surgeries completed. In the simulation model we do not allow non-consecutive assignments because the model is designed such that the nurse stops working whenever she is released in her overtime hours.

All problem instances were run in Arena Process Analyzer. The Process Analyzer (PAN) helps in the evaluation of different simulation model scenarios and is used for post-model development comparison of scenarios by allowing comparison of the outputs of models based on different model inputs. The advantage of using PAN is that it executes all scenarios in exactly same conditions. Otherwise, in different scenarios the random number generated for duration of a given surgery case would be different and the comparison would not be valid. The results of PAN for different problem instances are presented below. All scenarios were run for 100 replications. Average overtime, average number of surgeries completed and average utilization of all nurses are presented as measures of performance for the scenarios.

Because we have multiple measures of performance, the final decision depends on the decision-maker’s preferences on the measures of performance. Therefore, we develop a single score function that is a linear combination of all objective values in which each objective is associated with a weight corresponding to its level of importance. Identifying the values of these weights is beyond the scope of this study and needs input from nurse managers. In this problem the score function is a normalized weighted sum of nurse utilization, total overtime, and the number of surgeries completed.

Our aim is to calculate the score ρ for n given nurse schedules. Let λ_i be the normalized weight for the *i* th objective and ρ_j the score for the j^{th} schedule where *i= 1,2,3* and *j = 1,2,…,n* . If O_{ij} is the value of the *i* th objective for the j^{th} schedule, the score for the the j^{th} schedule will be

$$\rho(j) = \sum_{i=1}^{3} \lambda(i) * O(i, j)$$

Table 4 show the performance of the best performing solutions of the MIP model and the Genetic Algorithm in the simulation model for Problem 2. The purpose of this comparison is to benchmark the GA vs. the MIP model whose optimal solution was obtained for Problem 2. For this problem, GA results perform better under uncertainty. Average nurse utilization for GA result is twice the one of the MIP. Also, the average overtime is less under the GA results.

Table 4. Simulation results for MIP and GA solutions for Problem 2

Method	Average Overtime	Average Surgeries Completed	Average utilization
MIP	236.04	29.04	20.23
GA	200.47	22.91	43.30

Table 5 is the result of Problem 4 in a stochastic environment. In this problem, the number of nurses was the same as Problem 2 and we added seven more cases in shift 3. For this problem the MIP model has better overall performance than the GA. The MIP results have higher nurse utilization, but higher overtime compared to that of the GA results. This may imply that GA performs better when demand and resources are more balanced.

Table 5. Simulation results for MIP and GA for Problem 4

Method	Average Overtime	Average Surgeries	Average utilization
MIP	20.57	21.71	59.46
GA	18.25	21.59	26.39

Table 6 is the result of GA for Problem 5 (the largest problem). The scenarios have almost the same performance in demand satisfaction and nurse utilization. Therefore, Scenario 10 can be chosen if overtime and nurse utilization are of higher importance.

Table 6. Simulation results for GA for Problem 5

Scenario	Average Overtime	Average Surgeries Completed	Average utilization
1	478.43	39.71	46.44
2	593.24	38.03	43.01
3	593.24	38.03	43.01
4	763.78	38.61	48.16
5	660.05	39.31	43.37
6	763.78	38.61	48.16
7	438.09	38.54	44.88
8	660.05	39.31	43.37
9	442.37	37.69	44.81
10	420.94	38.89	48.30

Summary, Conclusions and future work

This study is an effort to solve a multi-objective nurse scheduling problem in an operating suite. Different shift patterns, surgery case requirements, and nurse skill sets make the problem very complicated. Moreover, the problem is very large, even when scheduling nurses for just one day. Existence of multiple objective functions adds to the complexity of the problem.

The objectives are to minimize total overtime, total idle time, and total non-consecutive assignment of nurses during overtime hours while the surgery demands are met. Considering the above facts, finding an optimal solution for the multi- objective nurse scheduling problem in the operating suite at MDACC is very difficult, often requiring untenably long computation times.

A methodology is proposed to find a good schedule in a reasonable time in which we use a simplified MIP model to create a solution pool. Then we try to find a better solution by using this solution pool as the first generation of a Genetic Algorithm that is specially designed for the Nurse Assignment Problem. This Genetic Algorithm is different from conventional GA because we have constraints in the problem and also we are dealing with a three-dimensional search space. Therefore, a new crossover rule and a repair procedure are considered in the GA for NAP.

Different problem instances are populated based on two sets of actual data from the operating suite. Several experiments are done to determine appropriate algorithm parameters such as stopping criterion, crossover rate and population size. Then, the algorithm was validated by solving different problem instances.

After GA stops, we have a set of schedules among which we should choose the one with a better fitness value. A simulation model of the operating suite is utilized to measure the performance of the resulting schedules under uncertainty. The probabilistic element in this simulation model is the surgery durations that follow certain probability distribution. The schedules evaluated in the simulation model are compared according to the average nurse utilization, total overtime and number of surgeries completed. Based on our preference over the objective functions a schedule can be selected to be used for that day in the operating suite.

Results show that for small problems the algorithm reaches to the optimal solution. For medium scale problems the algorithm terminates with a 30% gap in 25 to 40 minutes depending on the problem. The small and medium scale problems can also be solved optimally using MIP. However, the MIP model did not return a feasible solution for large scale problems in a few days. The Genetic Algorithm finds a good feasible solution for large problems more quickly than the MIP model; even when encumbered by being run in Matlab and processed on a low-power PC (2.2GHz dual-core machine with 3GB of RAM), the solution was found within 9 hours. While this may be acceptable if planning a few or more days in advance, much shorter solution times could likely be achieved if the GA was directly coded and run on a faster computer.

For further improvement of the scheduling tool, we will consider integration of nurse and case scheduling. As the schedule of cases in a day directly impacts nurse scheduling problem, an integrated nurse and case scheduling approach will be developed. In this approach, first, the case schedule would be improved, and, based on those results, an optimal nurse schedule will be generated. This can help recognize case scheduling policies that best utilize system resources.

References

Arnaout, J. M., S. Kulbashian. 2008. Maximizng the utilization of operating rooms with stochastic times using simualtion. Proceedings of the Winter Simulation Conference.

Ballard, S. M., M.E., Kuhl, 2006. The use of simulation to determine maximum capacity in the surgical suite operating room. Proceedings of the Winter Simulation Conference.

Berrada, I., J. A. Ferland, P. Michelon. 1996. A multi-objective approach to nurse scheduling problem with both hard and soft constraints. Socio-Economy Planning, 30(3); 183-193.

Cardeon, B., E. Demeulemeester, J. Beliaen. 2010. Operating room planning and scheduling: a literature review. European Journal of Operational Research, 201(3); 921-932.

Cochran, K. J., S. M. Horng, J. W. Fowler. 2003. A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines. Computers and Operations Research, 30; 1087-1102.

Denton, B., A.S. Rahman, H. Nelson, A. C. Bailey. 2006. Simulation of a multiple operating room surgical suite. Winter Simulation Conference.

Denton, B., J. Viapiano, A. Vogel. 2007. Optimization of surgery sequencing and scheduling decisions under uncertianty. Healthcare Management Science, 10(1); 13-24.

Dowsland, K. A. 1998. Nurse scheduling with tabu search and strategic oscillation. European Journal of Operational Research, 106; 393-407.

Gupta, D. 2007. Surgical suites operations management. Production and Operations Management, 16(6): 689–700.

Kawanaka, H., K. Yamamoto, T. Yoshikawa, T. Shinogi , S. Tsuruoka. 2001. Genetic Algorithms with the constraints for nurse scheduling problem. Proceedings of the 2001 Congress on Evolutionary Computation, 2; 1123-1130.

Lamiri, M., X. Xiaolan, A. Dolgui, F. Grimaud. 2008. A stochastic model for operating room planning with elective and emergency deman for surgery. Journal of operational Research, 185(3); 1026-1037.

Maenhout, B., M. Vanhoucke. 2007. An electromagnetic meta-heuristic for the nurse scheduling problem. Journal of Heuristics, 13; 359-385.

Mobasher, A., G. J. Lim, J. F. Bard, V. Jordan. 2011. Daily scheduling of nurses in operating suites. IIE Transactions on Healthcare Systems Engineering. 1, 232–246.

Paul, R. J., T. S. Chanev. 1998. Simulation optimization using a genetic algorithm. Simulation Practic and Theory. 6; 601-611.

Persson, A., H. Grimm, A. Ng, T. Lesama, J. Ekberg, S. Falk, P. Stablum. 2006. Simulation-based multi-objective optimization of a real world scheduling problem. Proceedings of the Winter Simulation Conference.

Ramis, F.J., J.L. Palma, F.F. Baesler. 2001. The use of simulation for process improvement at an ambulatory surgery center. Proceedings of the Winter Simulation Conference.

Sepúlveda, J. A., W. J. Thompson, F. F. Baesler, M. I. Alvarez, L. E. Cahoon. 1999. The use of simulation for process improvement in a cancer treatment center. Proceedings of the Winter Simulation Conference.

Texas Department of State Health Services and the Statewide Health Coordinating Council. Texas Hospital Nurse Staffing Survey: 2006.

Tsai, C., S. H. A. Li. 2009. A two-stage modeling with genetic algorithms for the nurse scheduling problem. Expert Systems with Applications, 36; 9506-12.

Tyler, D. , C. A. Pasquariello, C. H. Chen 2003. Determining optimum operating room utilization. Anesthesiology, 96; 114-21.