



A Nurse Rostering Algorithm for a District Hospital in South Africa

MJ Treurnicht* TE Lane-Visser† L van Dyk‡ SS Friedrich§

Abstract

An acute shortage of healthcare professionals is the rule rather than the exception in South Africa. Effective scheduling of nurses is critical to ensure good quality of care, while limiting staff related healthcare costs and abiding by labour laws. South African district hospital nurses are presently scheduled through the manual production of duty rosters on a monthly basis. In this paper, two related nurse rostering problems (NRP) are formulated for a district level public hospital in Stellenbosch (South Africa). The first problem addresses the scheduling of the months that nurses are on night shift duty. The other problem addresses the scheduling of the days that nurses are working night or day shifts within a month, respectively.

A hierarchy of four levels exists among the nursing staff at Stellenbosch Hospital. Distributed over the four levels, the nursing staff totals ninety employees. Due to hospital policy, no casual nurses are employed. Fluctuations in demand are met by scheduling overtime shifts that are limited by current labour legislation. The hospital consists of seven wards, each with separate staff requirements. The NRP for Stellenbosch Hospital is solved using the genetic algorithm. The algorithm is adapted to specifically adhere to the requirements and constraints given by the formulated NRP. The algorithm outputs optimal feasible rosters for each problem and provides data required to evaluate the performance of the algorithm. Roster results are interpreted and verified using the initial nursing requirements of the hospital. The robustness of using such an algorithm for sustainable use is also discussed. The paper ultimately aims to promote the use of operations research in healthcare on a practical level in South Africa.

Key words: Genetic algorithms, hospitals, metaheuristics, scheduling

1 Introduction

One of the major challenges in most developing countries is the provision of quality healthcare for all. Similar to many other developing countries, South Africa's public sector has an acute

*Corresponding author: Stellenbosch University, South Africa, email: miekie@sun.ac.za

†Stellenbosch University, South Africa, email: tanyav@sun.ac.za

‡Stellenbosch University, South Africa, email: lvd@sun.ac.za

§Stellenbosch University, South Africa, email: 15297839@sun.ac.za

shortage of medical expertise. Proper scheduling of nursing staff is critical to effectively utilise the available scarce resources. Nurse scheduling could have a large impact on consistency of care provided, eliminating excess or waste resources and reducing health costs as well as maintaining high staff morale [2, 5].

The nurse rostering problem (NRP) and the nurse scheduling problem (NSP) have attracted much research attention due to the time consuming and often complex nature of nurse scheduling. A large variety of NRP models and solution methods have been researched. These models typically involve the development of a periodic duty roster, subject to some constraints that are mostly hospital-specific. The objectives of the rosters could also vary between maximising the shift preferences of nurses to minimizing costs [2].

The NRP could be simplified by limiting the number constraints and complexity of the objective function. Many research articles limit the complexity of the problem to emphasize a solution method [5]. Hence, in academic papers complex optimisation methods and heuristic models are researched using simplified NRP. According to Kellogg and Walczak [5] the simplification of a problem results in a situation that the study is not likely to be implemented. For successful implementation of the NRP in hospitals, a complex set of constraints are mostly required. The result is that many real-world applications of the NRP are over-constrained, and complex to solve (Ernst et al. 2004).

Meta-heuristics such as the Tabu Search method, Simulated Annealing and Genetic Algorithms have proven to be effective in finding near-optimum solutions to NRPs [2, 4]. These solution methods use random orchestrated search strategies to explore a solution space, looking for a global optimum while avoiding local optima [3]. Despite the variety of commercial rostering software available and the many publications on the topic, the majority of hospitals still rely on manual scheduling. Kellogg and Walczak [5] attribute this to a mismatch between practical applications and the type of research that is published by academia.

The aim of the paper is to investigate whether a genetic algorithm can be used to solve a real-world NRP at a public hospital in South Africa. In this paper we investigate the use of a genetic algorithm to produce nurse duty rosters for Stellenbosch Hospital, a district hospital in the Western Cape, South Africa. Two similar NRPs are formulated to represent the current rostering specifications of the hospital. Genetic algorithms are specifically programmed to solve these NRPs. The solutions are discussed with recommendations for future work.

2 Formulating the NRP's for Stellenbosch Hospital

The majority of public hospitals in South Africa produce their nurse duty rosters manually. Nursing managers spend a substantial amount of time developing these rosters. Nurses are generally allowed to make requests that complicate the process even further. Nurse scheduling has always been a rather complex task. The primary reason for this is that hospitals are operational, 24 hours a day, 7 days a week. Furthermore, nursing staff needs to be scheduled in such a way that services with acceptable quality of care are always available, while simultaneously limiting the number of nurses employed and healthcare costs [2].

The nurse rostering problems are formulated specifically to model the existing practice at Stellenbosch Hospital. To deliver a 24 hour service at the hospital, shifts are defined as being

day or night shifts, each shift lasting 12 hours. It is compulsory that all hospital nurses work night shifts. To promote staff morale and to abide by labour laws, nurses work night shifts in blocks that should include at least three months. These night shift blocks are scheduled annually, allowing nurses to adapt to a night shift lifestyle.

Nursing managers produce two types of rosters. The first roster is produced annually, scheduling night or day shift blocks as well as ward allocation. This roster outputs in which months each nurse has to work night shift as well as ward allocations for each month. Monthly rosters are separated as day or night shift rosters, scheduling the days of the month in which each nurse is on duty. The monthly rosters use the output from the annual roster as input. Constraints are shared between the monthly and annual rosters, therefore if constraints should change, all the rosters should be modified.

2.1 Annual Duty Rosters

The purpose of this roster is to determine the months in which nurses have to work night shifts by considering nurse preferences as well as allocating nurses to wards. The problem is subject to some constraints that considers ward staff levels and nurse specialities. Nurses are asked to complete a preference matrix, indicating which months they would prefer to work night shifts. Another matrix contains the ward experience of each nurse. The objective function maximises the match between ward preference of nurses and the years of ward experience for all the nurses. For the purposes of this paper, the weighting of the nurse preferences and experience terms in the objective function are equal. Nevertheless, the importance of ward experience with respect to nurse experience would be different for other hospitals and could be changed according to the specifications of nursing managers. The problem is formulated as follows:

Defining variables:

$i = 1, 2, 3, \dots, 90$	Nurse index
$j = 1, 2, 3, \dots, 12$	Month index
$k = 1, 2, 3, \dots, 7$	Ward index
$l = 1, 2, 3, 4$	Speciality index

Parameters:

$B_{i,j} \triangleq$ Preference of nurse i to work during month j

$S_{i,l} \triangleq \begin{cases} 1, & \text{Nurse } i \text{ is qualified to work as speciality } l \\ 0, & \text{Else,} \end{cases}$

$P_{T_i,l} \triangleq \begin{cases} 1, & \text{A nurse following pattern } T_i \text{ works night shift during month } j \\ 0, & \text{A nurse following pattern } T_i \text{ works day shift during month } j, \end{cases}$

$W_{i,k} \triangleq$ Experience (in years) of nurse i in ward k

$C_{k,l} \triangleq$ Contraints for total number of nurses working night shift with specialty l in ward k .

Decision variables:

$T_i \triangleq \begin{cases} 1, & \text{Nurse } i \text{ is scheduled to work night shift pattern 1} \\ 2, & \text{Nurse } i \text{ is scheduled to work night shift pattern 2} \\ 3, & \text{Nurse } i \text{ is scheduled to work night shift pattern 3} \\ \dots \\ 48, & \text{Nurse } i \text{ is scheduled to work night shift pattern 48} \end{cases}$

$V_{i,j,k,l} \triangleq \begin{cases} 1, & \text{Nurse } i \text{ with speciality } l, \text{ works night shift during month } j \text{ in ward } k \\ 0, & \text{Else,} \end{cases}$

$X_{i,j,k,l} \triangleq \begin{cases} 1, & \text{Nurse } i \text{ with speciality } l, \text{ works day shift during month } j \text{ in ward } k \\ 0, & \text{Else,} \end{cases}$

Objective Function:

Maximise the total preference and experience of all nurses over the period of a year:

$$\max z = \sum_j \sum_j B_{i,j} P_{T_i,j} + \sum_i \sum_j \sum_k \sum_l W_{i,k} V_{i,j,k,l} \quad (1)$$

Subject to:

Each nurse should be assigned to one ward per month:

$$\sum_{k=1}^7 \sum_{l=1}^4 (V_{i,j,k,l} + X_{i,j,k,l}) = 1 \quad i = 1, 2, 3, \dots, 90; j = 1, 2, 3, \dots, 12 \quad (2)$$

Each nurse is only assigned to their qualified speciality per month:

$$\sum_{k=1}^7 (V_{i,j,k,l} + X_{i,j,k,l}) = S_{i,l} \quad i = 1, 2, 3, \dots, 90; j = 1, 2, 3, \dots, 12; l = 1, 2, 3, 4 \quad (3)$$

Number of night shift nurses according to speciality in wards per month

$$\sum_{i=1}^{90} V_{i,j,k,l} \leq C_{k,l} \quad j = 1, 2, 3, \dots, 12; k = 1, 2, 3, \dots, 7; l = 1, 2, 3, 4 \quad (4)$$

All nurses are scheduled according to shift patterns (expressed in terms of night shifts months)

$$\sum_{k=1}^7 \sum_{l=1}^4 V_{i,j,k,l} = P_{T_i,j} \quad i = 1, 2, 3 \dots, 90; j = 1, 2, 3, \dots, 12. \quad (5)$$

2.2 Monthly Duty Rosters

The monthly rostering problem determines the shift patterns for the months in which nurses are on duty. The monthly rostering problem is similar to the night shift rostering problem. Since wards are assigned for each month on an annual basis, the assignment of wards is not included in the monthly rosters. The monthly rostering problem inputs the results from the annual roster for specific constraints, such as the speciality of nurses scheduled for day or night shift and their assigned wards. The objectives of the monthly rostering solutions are to maximise the sum of the preference for the entire nursing staff. The problem is subject to a set of constraints that are specific to each of the seven wards and four specialities. For example, in the maternity ward, exactly two sisters have to be on duty for each day of the month.

Defining variables:

$i = 1, 2, 3, \dots, 90$	Nurse index
$d = 1, 2, 3, \dots, 12$	Day index
$k = 1, 2, 3, \dots, 7$	Ward index
$l = 1, 2, 3, 4$	Speciality index

Parameters:

$D_{i,d}$	$\hat{=}$ Preference of nurse i to work during day d
$S_{i,l}$	$\hat{=}$ $\begin{cases} 1, & \text{Nurse } i \text{ is qualified to work as speciality } l \\ 0, & \text{Else,} \end{cases}$
$N_{i,k}$	$\hat{=}$ $\begin{cases} 1, & \text{Nurse } i \text{ is scheduled to work in ward } k \text{ for this month} \\ 0, & \text{Else,} \end{cases}$
$Q_{R_i,k}$	$\hat{=}$ $\begin{cases} 1, & \text{A nurse following pattern } R_i \text{ works night shift during day } d \\ 0, & \text{A nurse following pattern } R_i \text{ works day shift during day } d, \end{cases}$
$K_{k,l}$	$\hat{=}$ Constraints for total number of nurses with specialty l in ward k .

Decision variables:

$$R_i \triangleq \begin{cases} 1, & \text{Nurse } i \text{ is scheduled to work pattern 1} \\ 0, & \text{Nurse } i \text{ is scheduled to work pattern 0} \end{cases}$$

$$V_{i,d,k,l} \triangleq \begin{cases} 1, & \text{Nurse } i \text{ with speciality } l, \text{ works during day } l \text{ in ward } k \\ 0, & \text{Else,} \end{cases}$$

Objective Function:

Maximise the total preference of all nurses over the period of a month

$$\max z = \sum_i^{90} \sum_d^{31} D_{i,d} Q_{R_i,d} \quad (6)$$

Subject to:

Each nurse is assigned according to the wards scheduled for each day in the monthly roster

$$\sum_{l=1}^4 V_{i,d,k,l} = N_{i,k} Q_{R_i,d} \quad i = 1, 2, 3, \dots, 90; d = 1, 2, 3, \dots, 31; k = 1, 2, 3, \dots, 7 \quad (7)$$

Each nurse is only assigned to their qualified speciality for each day

$$\sum_{k=1}^7 V_{i,d,k,l} = S_{i,l} Q_{R_i,d} \quad i = 1, 2, 3, \dots, 90; d = 1, 2, 3, \dots, 31; l = 1, 2, 3, 4 \quad (8)$$

Number of night shift nurses according to speciality in wards per day

$$\sum_{i=1}^{90} V_{i,d,k,l} \leq K_{k,l} \quad i = 1, 2, 3, \dots, 90; k = 1, 2, 3, \dots, 7; l = 1, 2, 3, 4 \quad (9)$$

All nurses are scheduled according to specific shift patterns

$$\sum_{k=1}^6 \sum_{l=1}^4 V_{i,d,k,l} = Q_{R_i,d} \quad i = 1, 2, 3, \dots, 90; j = 1, 2, 3, \dots, 31 \quad (10)$$

2.3 Limitations to the NRPs

The formulated NRPs do not address fluctuating demand, leave, overtime shifts or hiring temporary nurses. Unit managers assess the demand for nurses continuously and assign overtime shifts accordingly. Nurses are requested to apply for leave a month in advance. It is against Stellenbosch Hospitals policy to hire additional temporary nurses. Overtime is scheduled during the month to compensate for demand fluctuations and nurses leave requirements. These aspects could be added to the problem, however, at this stage of developing a rostering solution, it is desired that the unit managers still have manual control over detailed shift changes and overtime shifts.

3 Solving the NRP using Genetic Algorithms

Two genetic algorithms were programmed for the respective annual and monthly NRPs. Microsoft Excel was chosen as a computing platform together with Visual Basic as programming language. The reasoning behind using Microsoft Excel as user interface is that administrative staff and unit managers at the hospitals are already trained to use Microsoft Office. Unit managers are currently using Microsoft Excel to manually produce duty rosters. They would therefore easily adapt to using an algorithm to produce a duty roster that can be modified manually using the well known spreadsheet functionality of MS Excel.

Optimization problems require both variety and progression. Natural phenomena, therefore, provide valuable principles for algorithms. Genetic algorithms mimic the biological theory of evolution where plants and animal species breed to form new offspring with unique features. With the birth of a new offspring a new generation is formed. The concept of evolution is that new generations possess different characteristics than the previous generation with the capability of improved performance in the new environment. The survival of the fittest principle is applied and thereby as time progresses the new generations become stronger through abandoning weaker individuals. Mutations that occur randomly reduce the possibility of inbreeding and therefore the chances that the population is trapped at a local optimum [4].

Unfortunately, solving highly constrained problems with genetic algorithms is rather complex. A reason for this is that the generation of new offspring through crossover does not allow for constraint consistency. Hence, the birth of a new offspring through crossover does not necessarily allow for a new feasible solution. The new offspring has to meet with the requirements of the constraints before it can be considered a possible solution. If infeasible solutions dominate the solution space, it is unlikely that the genetic algorithm will succeed in finding a good feasible solution [1].

Infeasible offspring can be avoided by using penalty functions or repairing infeasible solutions. Penalty functions steer the search away from infeasible solutions, thereby avoiding the problem. Unfortunately, there are no guarantees that a solution can be found through avoiding this complication with penalty functions. Repairing infeasible solutions are also not the perfect solution. Through repairing, the characteristics of the offspring are often changed in such a way that a weaker individual is formed, resulting in a time consuming process of generating many weak offspring that are abandoned and does not contribute in creating a stronger population [1].

3.1 Initiation

The initial population consists of a number of feasible trial solutions. Each individual solution in the population is created randomly and repaired to meet the set of constraints. An individual solution consists of genes that determine the strength of the individual. In the NRP a gene could be the shift type chosen for a corresponding nurse. A basic feasible solution would therefore consist of a number of genes, indicating the chosen shift types for a number of nurses. The fitness (value of the objective function) is calculated for each feasible member of the initial population.

3.2 Iterations

The iteration phase of the algorithm randomly selects parents to compete in a tournament. Two parents with relative high fitness are chosen and paired with each other in a multiple crossover process. Crossover points are chosen at random, and therefore if two parents were to pair more than once, different offspring will be created. Offspring inherit a combination of both parents genes, a different combination of genes from the same parents would therefore result in a different child. Mutation occurs on a random basis. The operator of the algorithm can specify a mutation probability. This probability determines the mutation rate. If the mutation possibility is met, the new offspring is generated with new genes that do not necessarily belong to any specific parent.

New offspring is subject to the set of constraints. If a new offspring is created that does not fit criteria and are therefore infeasible, the solution (offspring) is repaired in the same way that the individuals of the original population were repaired. The fitness of each offspring is calculated. If the offsprings fitness is better than the weakest member of the population, any random old member of the population is replaced with the offspring. This process of creating new offspring is continued until the termination criterion is met.

3.3 Termination rule

The iteration phase is a loop that continues until a termination condition is encountered. There are a number of different termination rules that can be used such as a fixed CPU time, a fixed number of iterations or a fixed number of consecutive iterations without improvement. If the algorithm is executed for an adequate time period, the entire population will have the same fitness. If no new offspring are created during a number of iterations, it can be assumed that the algorithm has found a near optimum solution. Therefore, after reaching the termination criterion the solution space would have converged to the final solution.

4 Results

At first the genetic algorithms were programmed to randomly search for a feasible solution, without repairing or using penalty functions. As literature suggested, the algorithm was unable to find any feasible solutions in an acceptable time period. The reason for this is most likely the diversity in shift patterns that could be assigned. If no structure is provided, the algorithm will randomly search for a combination of shift patterns among the nurses that satisfies the constraints. Unfortunately, the possibility of finding a combination of 48 patterns among 90 nurses that fits the constraints is very small.

By repairing individuals to create a population of feasible solutions, infeasible solutions were removed from the solution space. It should be noted that it is possible that the repairing method could influence the impact of the crossover method. This is due to the fact that an offspring is repaired after crossover. It is therefore likely that the offspring after repairing might not be similar to either parent. The good characteristics might not survive the repair process. It is therefore likely that through repairing the offspring, the good characteristics might get lost and the algorithm might stop at a local optimum.

The genetic algorithms were successful in finding feasible rosters for both annual and monthly rosters. Using genetic algorithms to solve two related NRPs we were able to accomplish the following:

- An annual roster that allocates each nurse to night shift blocks and wards for each month of the year
- A monthly roster that specifies the days that each nurse are on duty

The genetic algorithms output for the annual is in the form of shift pattern numbers for all the nurses, which are translated to a roster showing the months in which the nurse are on night or day shift duty. The monthly rosters output is similar to the annual roster, with the exception that wards are not allocated on a monthly basis. The monthly roster will indicate the days of the month that the nurses are on duty. If the annual roster indicates that the nurses are on night shift duty, the days that the nurses are on duty according to the monthly roster will apply to night shifts. The same principle applies to day shifts. Thus the two rosters are dependent and should not be used independently.

5 Conclusions and Recommendation

The final feasible rosters are good solutions, but are most likely not the optimum solutions. Nevertheless, the aim of this study was to provide a good alternative to producing manual rosters at Stellenbosch Hospital. The algorithms were able to find a variety of different rosters that would fit the constraints of the hospital and choose the best rosters for nursing staff preference and ward experience.

The next step is to test and validate the method for Stellenbosch Hospital. If necessary, constraints such as leave, overtime and demand fluctuations could be added. Further alterations will be done at the request of the hospital staff. If the hospital staff would like to replace manual scheduling with the algorithm, implementation should include adequate training and support.

Results from the algorithms should be validated during implementation to ensure that all the nurses are treated fairly and hospital demands are met. If possible results in terms of nurse preferences should not be made public to the general nursing staff. It is inevitable that some nurses will have higher preference ratings than others. It should therefore be avoided that nurses compare their preferences with each other. This will reduce the jealousy effect of introducing the preference method.

The genetic algorithm was programmed to be problem specific. This enabled the algorithm to search within a structure as opposed to absolute random searches. Therefore if this algorithm is considered for another hospital similar to Stellenbosch Hospitals, alterations are necessary.

Acknowledgements

The authors wish to thank the staff of Stellenbosch Hospital for their enthusiasm and support in this project.

Bibliography

- [1] Aickelin, U. and Dowsland, K. (2004). An indirect genetic algorithm for a nurse scheduling problem. *Computers & Operations Research*, 31:761–778.
- [2] Cheang, B., Li, H., Lim, A., and Rodrigue, B. (2003). Nurse rostering problems - a bibliographic survey. *European Journal of Operational Research*, 151:447–460.
- [3] Glover, F. and Kochenberger, G., editors (2003). *Handbook of Metaheuristics*. Kluwer Academic Publishers, Boston.
- [4] Hillier, F. and Lieberman, G. (2005). *Introduction to Operations Research*. McGraw-Hill, Singapore.
- [5] Kellogg, D. and Walczak, S. (2007). Nurse scheduling: From academia to implementation or not? *Interfaces*, 37:355–369.