# An Evolutionary Intelligent Algorithm Approach for the Doctor Scheduling Problem

Abir Alharbi and Kholood AlQahtani

Mathematics Department, King Saud University, Riyadh, Saudi Arabia

e-mail:

abir@ksu.edu.sa , 432201096@student.ksu.edu.sa

***Abstract-*** **In this paper, we present an automated intelligent nature inspired algorithm solution to the scheduling problem for doctors in Pediatric Department of Prince Sultan Military Medical City in Riyadh Saudi Arabia. The genetic algorithm approach is utilized, which is an adaptive heuristic search algorithm based on the evolutionary ideas of natural selection and on genetics. A cost bit matrix of which each cell indicates any violation of the required constraints to make an acceptable doctor schedule, and this novel cost bit matrix is used as an objective function for the genetic algorithm. The experimental results showed that our genetic algorithm approach generates an automated intelligent doctor schedule faster and with less violated constrains than the traditional manual methods and other hill climbing search techniques.**

***Keywords- Doctor Scheduling Problem; Genetic Algorithms; Cost bit matrix; Hard constrains; Soft constrains.***

## I. INTRODUCTION

Time Scheduling assigns an appropriate number of workers to the specified jobs during each day of work. Many fields require scheduling such as job, production, railway, personnel scheduling and many more. Personnel scheduling plays an important task in institutions, it balances the workload of the personnel, departments, and effects the productivity of the whole institution. Personnel scheduling involves assigning of personnel to time slots and possible different locations known as time scheduling. It is made to maximize the work done in the institution and guarantee the increasing productivity or (to increase productivity). In hospitals, personnel scheduling is applied to assign doctors or nurses to their respective departments and making sure all shifts are covered.

A hospital providing a round-the-clock service divides its daily work into consecutive shifts, and a shift is a period of time in which a group of employees are in-service. A doctor is assigned to a set of shifts which satisfies several constraints that may be set up by staffing requirements, rules by the administration, and labor contract clauses. In a doctor scheduling problem (DSP) each doctor is assigned to the set of shifts and rest days in a timetable called a doctor roster.

The objective of DSP is to satisfy doctors' requests as much as possible while fulfilling the employers' requirements. In DSP there are many constraints and there can be several different instances with separate set of constraints. In this study, we focus on the cyclic Doctor scheduling problem described through the following three components and constrains:

(l) The personal preference of each doctor to work on particular days and shifts,
(2) The minimal coverage constraints of the minimal required number of doctors per shift and per day,
(3) The case-specific constraints specified by personal time requirements, specific workplace conditions, etc.

DSP consists of creating weekly or monthly schedules for a number of doctors by assigning one out of a number of possible shift patterns to each doctor. These schedules must satisfy working contracts and meet the requirements for the number of doctors of different grades for each shift, while being seen to be fair by the staff concerned. In a hospital, there are various kinds of employers like doctors, nurses, attendants, etc. and they must be assigned to shifts to do their work. In this study, we concentrated mainly on the doctor shifts.

DSP was proven to be NP-hard even with only a subset of real world constraints [4]. DSP can be modeled as a partial constraints satisfaction problem. Numerous studies have been done in the recent years, meta-heuristics such as Tabu search [23], genetic algorithm (GA) [3], [1], [18], constraint logic programming [2], and simulated annealing [13] have been proven to be effective in obtaining good solutions for doctor or nurse scheduling problem. The idea behind using genetic algorithm to solve DSP is to breed the fittest solutions in a specific generation. GA works with a population of “individuals”, each representing a possible solution to a given problem. Each individual is assigned a “fitness score” according to how good a solution is to the problem. Individuals with higher fitness scores are given opportunities to “reproduce” by “cross breeding” with other individuals in the population. Genetic Algorithms (GAs) are powerful general-purpose optimization tools which model the principles of evolution [15]. They are often capable of finding globally optimal solutions even in the most complex of search spaces [11]. They operate on a population of coded solutions which are selected according to their quality then used as the basis for a new generation of solutions found by combining (crossover) and (mutating) current individuals. The strength of GAs comes from the fact that the technique is robust and can deal successfully with a wide range of problem areas, including those which are difficult for other methods to solve. In this paper, we apply a GA approach with a cost bit matrix that penalizes the solution of the DSP if it violates constrains, and eventually finds the optimum doctor's schedule.

In Section II, we will present some of best practices that are available in the literature on solutions of DSP. In Section III, we will briefly introduce the GA and its parameter settings, also our DSP setup and its objective function, which is designed from the customized cost bit matrix. Moreover, in Section IV, the GA results are discussed and compared to other methods. Finally, conclusions and future work are discussed in Section V.

## II. LITRATURE REVIEW

In the literature, many research works are done on DSP or the similar Nurse scheduling problem (NSP). Miller et al. [16], and Warner et al. [23], used linear programming to formulate the Nursing Schedule Problem as the selection of a timetable that minimizes an objective function, which balanced the trade-off between staffing coverage and nurses' preferences. Abdannadher et al. [2] applied a Constraint Logic Program (CLP) framework, and Li et al. [14] employed Bayesian optimization algorithm. Aickelin et al. [3], applied an indirect genetic algorithm to NSP. Existing research work has been proposing diverse models and methodologies to improve nurse and doctor scheduling problems. Most of the current proposed solutions either make use of random based optimization algorithms such as Silver et al. [19], where a lot of research work related to Medical Informatics has done, and presented deep discussions about the role of data warehouse management system to handle hospital and nurse management information. Moreover, multidimensional analysis techniques under different parameters were used to extract the required data and information.

An approach that has been implemented successfully in many of the American hospitals, is based on two data mining techniques called; patient rule introduction method (PRMI) and weighted items sets (WLS). They were used to analyze large quantities of data in Villiers et al. [22], where data mining techniques for solving clinical data warehouse functionality was applied, and they proposed a flexible clinical data mining system (CDMS) using SAS statistical software. In addition, they conducted research in two stages. In first stage, controlled environment was provided for CDMS access based systems and transformed it into analytical clinical data, and in the later stage, operations were tested. Cheng et al. [7] describes the design and implementation of a constraint-based nurse rostering system using a redundant modeling approach, to reduce search time. An effective way to increase constraint propagation through cooperation among different models for the same problem, was also proposed. Kundu et al. [13] described the use of GA for solving nursing schedule problem. They used two different models, a Simulated Annealing approach, and a Simulated Annealing combined with GA approach. They compared nurse performance at various levels, and have considered soft and hard constraints.

Juhos et al. [12] described a novel representation and ordering model that was aided by an evolutionary algorithm, to solve the graph k-coloring problem setup of NSP. Its strength lies in reducing the number of neighbors that need to be checked for validity. An empirical comparison was made with two other algorithms on a popular selection of problem instances and on a suite of instances in the phase transition. The new representation in combination with a heuristic mutation operator showed promising results. Culberson and Gent [9] denned the ‘frozen development’ of coloring random graphs and identified two nodes in a graph as frozen if they were the same color in all legal colorings. This was analogous to studies of the development of a backbone or spine in SAT (the Stainability problem). In other works, the graph coloring techniques and greedy algorithm were used for NSP, such as by Ratnayaka et. al. [17], where they used an enhanced greedy optimization algorithm with data warehousing for an automated nurse scheduling system, and by Gideon [10], where a thesis on a nurse scheduling methods and solutions were presented. In reference [18] a GA is used to solve nurse scheduling problem for private hospitals in the Philippines. This study considered preferences of nurses in terms of work schedule while meeting the objective of the hospital management of minimizing total salary cost, and used different operators for the genetic algorithm in solving the NSP, they claim that their method could save 12% staffing expenses monthly as compared to existing manual schedules.

GAs are used to solve other scheduling problems such as timetables problem in schools [8], where they compared a GA-based approach with various versions of Simulated Annealing and Tabu search, and in the experiments GAs produced better timetables than Simulated Annealing, but slightly worse timetables than Tabu search. The railway scheduling problem was also considered in [21] which implies the optimization of trains on a railway line that is occupied (or not) by other trains with fixed timetables. The timetable for the new trains is obtained with GA, which includes a guided process to build the initial population. The proposed GA is tested on real instances obtained from the Spanish Manager of Railway Infrastructure (ADIF). The results point out that GA is an appropriate method to explore the search space of this complex problems and can lead to good solutions in a short amount of time.

In this paper, we apply a GA with a cost bit matrix that penalizes the solution of the DSP if the constrains are violated, and hence finds an automated schedule that optimizes the doctors' roasters and satisfies all the constraints. Our novel cost bit matrix is specially designed for the constrains requested by the hospital and it is used as the objective function in the evolutionary genetic algorithm, this new automated technique has not been reported before in the literature on the DSP.

## III. Genetic Algorithms

Genetic Algorithms (GAs) are adaptive heuristic search algorithms based on the evolutionary ideas of natural selection and genetics [20]. The basic concepts of the GA were designed to simulate those processes in natural evolution system, and survival of the fittest [15]. GA are a powerful tool to solve optimization problems with multiple variables [11]. GA were applied to several scheduling problems [1], [6], [21]. GA are a search algorithm to simulate the process of natural selection. GA start with the set of potential solutions called population and evolves toward more optimal solutions. The solutions are evaluated by a fitness function. The fitness value represents the quality measure of a solution so that the algorithm can use it to select ones with better genetic material for producing new solutions and further generations. The selection chooses superior solutions in every generation and assures that inferior solutions extinct. The crossover operator chooses two solutions from current population and generates a new solution based on their genetic material. Selection and crossover operators will expand good features of superior individuals through the entire population. They will also direct the search process towards a local optimum [20]. The mutation operator changes the value of some genes in a solution and help to search other parts of problem space. The main disadvantage of GA is the requirement for a large computation time.

### A. *Doctor Scheduling Problem*

Hospital care units must provide twenty-four-hour coverage at levels to match patient demand while adhering to organizational policies designed to protect the health and welfare of patients and staff. The already difficult scheduling problem is further compounded by a shortage of doctors. The schedule must determine the day-to-day shift assignments of each doctor for a specified period in a way that satisfies the given requirements as much as possible, taking into account the wishes of doctors as closely as possible. In most studies of DSP, the restrictions are categorized as Hard Constraints and Soft Constraints. Hard constraints are the constraints, which must be satisfied to get feasible solution for use in practice, and Soft constraints are the constraints, which are used to evaluate the quality of the solution. So, soft constraints are not compulsory but are desired to be satisfied as much as possible. Hard constraints should always be satisfied in any working schedule so that there will be no breaches. Any schedule that does not satisfy all the hard constraints cannot be a feasible one. Possible examples include restrictions on the number of doctors for each shift; the maximum number of shifts in a week, a month, etc. On the contrary, soft constraints can be violated but as minimal as possible. In other words, the soft constraints are expected to be satisfied, but violation does not make it an infeasible solution. The doctors are divided into three categories: consultants (C), senior (S) and junior doctors (J) working in the same department. In this project we are concerned with scheduling shifts for junior doctors in the pediatric department wards for one month only. A major problem with any scheduling problem is the allocation of resources in an effective way, and not violating the constraints, which affects the quality of the solution. We confined the constraints as follows:

(a) Hard constraints

(i) There are constraints on the number of doctors for each working shift per day. The number of doctors for morning (m), evening (e), and night shift (n) should be between the minimum and maximum values.

(ii) There are constraints for the working patterns. Morning after night shift, evening after night, morning after evening shift and three consecutive night shifts are restricted combination of working patterns.

(b) Soft constraints

There are constraints for the total number of off-days (o), night, morning and evening shifts during the certain period of days for each doctor.

In this paper, we shall present an automated scheduling solution for Pediatric Department of Prince Sultan Military Medical City (PSMMC) in Riyadh, Saudi Arabia. Monthly doctors' rosters are made manually before the end of each month, Figure 1, shows the original hospital roasters for the month of February 2016. Even though making monthly rosters manually required significant effort and time, it did not resolve all conflicts in the doctors' schedules specially in the time of emergency interventions or in times of sudden timetables' changes. In fact, sometimes making slight changes in one doctor's timetable meant creating a lot of tedious adjustments to the rest of the doctors' timetables working in the same ward. Proposing an automated software that can handle making all doctors roasters according to the constraints and at the same time be flexible enough to produce a new schedule quickly in case of needed sudden change, is very desirable by the hospital.

### B. *The Cost Function*

We must define a customized objective or cost function so that when we optimize it, it will obtain optimal time schedules for all doctors in the department. Let N be the number of doctors, and D the number of days. Then, DSP may be represented as an optimization problem to find a schedule matrix X of size NxD, so that each element of the matrix, $x_{ij}$, expresses shifts for the doctor i working on day j, and takes one of the four shift values (m, e, n, or o): m=morning, n=night e=evening, and o= off days. We calculate the fitness of each suggested schedule table by using our customized cost bit matrix through the following procedure:

(a) To evaluate the violation of hard constraint (i), we define mt, et, nt as the total number of doctors for morning, evening, and night shift on a day. If any of these numbers are not between the minimum and maximum number of doctors for each shift (mmin, mmax, emin, emax, nmin, nmax), provided by the hospital, then the cost $\boldsymbol{C_1}$ will be incremented by l.

(b) To evaluate the violation of hard constraint (ii), working patterns are examined. Any violation of the working patterns specified (such as m after n, e after n, m after e, or consecutive n, n, n) will increment the cost $\boldsymbol{C_2}$ by l.

(c) To evaluate the violation of soft constraint, we define M, E, N, O as the total number of the corresponding shifts, morning, evening and night and off-days for a doctor during a period of days. Mreq, Ereq, Nreq, and Oreq are set as the required number of mornings, evenings, nights shift and off-days for all doctors during a period of days. If any of these numbers M, E, N, O does not meet, Mreq, Ereq, Nreq, and Oreq respectively, then the cost $\boldsymbol{C_3}$ will be incremented by 1.

Hence, our cost function will consist of the three parts representing each cost: $\boldsymbol{C_l}$, $\boldsymbol{C_2}$ and $\boldsymbol{C_3}$ as follows:

$$\boldsymbol{f} = C_1 w_1 + C_2 w_2 + C_3 w_3 \tag{1}$$

where $w_I$, $w_2$ and $w_3$ are weight values set for each $\boldsymbol{C_l}$, $\boldsymbol{C_2}$ and $\boldsymbol{C_3}$, respectively emphasizing our priority in the satisfaction of constraints, and are determined by trial and error to find the best combination of weights.

Our goal is to minimize the cost function $\boldsymbol{f}$ given in (1) and hence find an optimal doctor schedule satisfying all hard and soft constrains. The simplest method to find the solution is a brute force approach (manually) evaluating all possible doctor schedules and finding the feasible one with the minimum cost among them. Manual methods guarantee finding a feasible schedule with the minimum cost, however, if the number of doctors increases, this approach is intractable. Moreover, it is not easy to adjust manually a timetable for one doctor's shift due to an emergency or sudden time schedule change, without needing to make many changes in the other involved doctors' timetables as well. This is considered as a class of NP-hard problem for which many methods have been applied and no unique solution exists [5], [24]. Hence, an automated algorithm that produces an optimal schedule for N doctors in D day in minutes and with no constraints violated is needed. Here, we propose to use genetic algorithm to design such an automated algorithm, and therefore present an optimal solution to the DSP in an abbreviated time, via our novel cost bit matrix construction of doctors' time schedules.

### *C. GA Paramreters for Selction and Crossover*

For the first week of Febraury, the Hospital divides doctors into three groups. Each group has one senior and three to four junior doctors based on the department daily requirement, which requires availability of ten junior and three senior doctors on that week, as seen in Table I . So they setup three groups: A, B, and C for doctors to cover the days of the week and the doctors shared the same daily schedule. For our approach with genetic algorithms, we did not need to use groups for the doctors instead we worked on producing an independent schedule for each junior doctor.

The initial population for the GA is made of different possible schedules for doctors' shifts, such as each member of the population is a NxD matrix, generated randomly assigning each doctor to one of the four daily shifts and assuring a day-off on each week, Table II shows a sample of a week schedule for 10 junior doctors (10x7 matrix), in the population. The costs of each member schedule in the population is calculated by the cost function given in (1). We will start with z = 60 members in the initial population set in random.

The method of selection in this study, is the roulette wheel selection method, which is the most common type of selection methods [11]. Two schedules, P1 and P2, are chosen randomly based on their cost and are used to produce an offspring. One schedule can be selected for a parent more than once. The crossover between the two chosen parents' genome is done at a single point randomly chosen with probability 0.8 to produce the new generation offspring. Additionally, the mutation rate is set to 0.01.

The remaining initial parameters are set as given by the PSMMC hospital for February 2016:

- N=24, D=29
- mmin=8, mmax=10, emin=6, emax=10, nmin=6, nmax=10
- soft constrains for each week were: Mreq=Ereq=Nreq=2, and Oreq=1.

The method was activated to reach an optimum cost according to (1) ($\boldsymbol{f}$ = 0, i.e., no violations to both soft and hard constrains on the proposed optimum schedule) using MATLAB genetic algorithm toolbox with Intel Core ™ i5-250M 2.5 Ghz CPU and 4GB. The GA goes through the following steps:

Step 1: Initialize the population randomly (with z individuals or NxD schedules) and calculate the fitness function for each member of the initial population using the cost bit matrix ($\boldsymbol{f}$) defined by in (1).

Step 2: Select the elite members of the population, which will go through crossover and mutation based on their high fitness scores (satisfying most hospital constrains with the minimum number of violations).

Step 3: Choose parents for crossover, and with the crossover rate, generate offspring, in which the ranking mechanism is used for selection of chromosomes.

Step 4: Apply the mutation, on selected offspring, according to mutation rate.

Step 5: Select the members of the new generation consisting of: the parents in the old generation and the produced new offspring in Step 4, according to their fitness values.

Step 6: Repeat the procedure in Step 2 through Step 5 until the maximum number of generations is reached, or the objective function is optimized, and the threshold is met.

TABLE I. SAMPLE OF HOSPITAL SCHEDULE GROUPS FOR JUNIOR AND SENIOR DOCTORS

| Group | Doctors |
|---|---|
| A | J1, J5, J4 and S1 |
| B | J3, J2, J6, J10 and S2 |
| C | J7, J8, J9 and S3 |

TABLE II. SAMPLE HOSPITAL SCHEDULE FOR 10 DOCTORS AND DURATION 7 DAYS: MONDAY-SUNDAY)

| Doctor | M | T | W | TH | F | S | SU |
|---|---|---|---|---|---|---|---|
| J1 | m | e | n | o | m | m | n |
| J2 | e | m | m | n | o | m | n |
| J3 | n | n | o | m | e | e | e |
| J4 | m | m | e | e | n | o | m |
| J5 | e | o | e | n | m | n | e |
| J6 | m | e | n | o | m | m | n |
| J7 | e | m | m | n | o | m | n |
| J8 | n | n | o | m | e | n | e |
| J9 | m | m | e | e | n | o | m |
| J10 | e | o | e | n | m | n | e |

## IV. GA RESULTS

The GA started with a population size of z=60 individuals, with the size of each genome NxD matrix (24 doctors for 29 days). The algorithm terminated when the maximum number of generations was reached at 300, or when the increase in fitness of the best individual over five successive generations fell below a certain threshold, set at 2 $\times 10^{-6}$. Our fitness function $\boldsymbol{f}$ is set to the cost function:

$$f = 5C_1 + 5C_2 + C_3,$$

which penalizes schedule tables violating the constrains according to the assigned weights. The weights were chosen according to several attempts of trial and error on different weight values, and it was concluded that any values chosen for $w_1$ and $w_2$ as penalties for violating $C_1$ and $C_2$ should be equal, but $w_3$ must be set to a lesser value since it is hard to avoid violating $C_3$ [1]. The GA ran throughout generations to find the best genome in this population. The best genome is the doctor schedule table, which violates the least number of constrains. After all 300 generations (repeated 50 times), the genetic algorithm found the optimum genome; hence, it found the best doctor schedule table, which violates the least number of soft constrains. Our proposed GA applied on the cost bit matrix have given excellent results on solving this doctor schedule problem in both time efficiency and accuracy. Figure 2 shows the plots of the best fitness value reached (with only 3 soft constrains violations over 4 weeks for all 24 doctors) and the mean fitness value over the generations. Moreover, Figure 3 shows the average distance between all individuals during the GA generations. The best doctor schedule produced from the GA is given in Table VI, and it can be seen through the plot of the best individual with 695 variables in Figure 4.

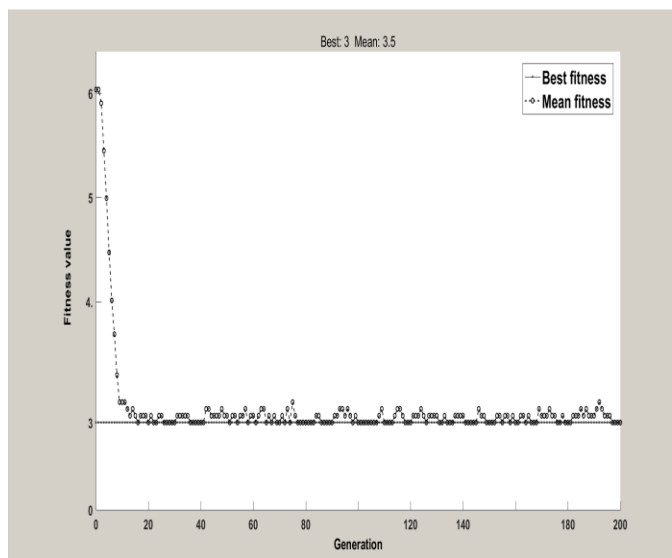


Figure 2. The best and mean fitness value over the generations for 4 week.

The proposed GA results are compared to the hospital manual roaster tables derived from Figure 1 and Table V, which shows the incident matrix for the 24 doctors in PSMMC for the month of February 2016. For comparison purposes our data was also used to find the best doctor schedule with two other methods published in the literature Kundu et al. [13] using a Simulated Annealing and Genetic

Algorithm combination method. Also, we compared our results with the hill climbing technique described by Wojciech et al. in [24].

A hill climbing (HC) algorithm, which is a fast-local search algorithm, despite being a simple search method, showed a satisfactory performance for small size instance while, it could not cope with medium and large size instances. HC is driven by a greedy procedure. It processes interventions one by one trying to assign them to teams as early as possible. Above that, it iterates over the intervention urgency classes and several possible sorting strategies. At the top level, different permutations of intervention priorities are considered. Finally, the algorithm tries to improve the current partial solution by applying a hill climbing metaheuristic and by abandoning some interventions. In simple words, it depends on the trade-off between the weights in the scoring procedure and the distribution of intervention priorities. Where Perm(x) is a certain permutation function of priorities {G1, G2, G3, G4} given in Table III. In order to analyze all possible orderings of priority classes, the whole algorithm is executed for six different permutations Perm: (G1, G2, G3, G4), (G1, G3, G2, G4), (G2, G1, G3, G4), (G2, G3, G1, G4), (G3, G1, G2, G4) and (G3, G2, G1,G4). Eventually, the final solution is the best obtained. Notice that the last priority never changes, because the scoring function is defined in such a way that it does not take into account the finish time of interventions of priority 4.

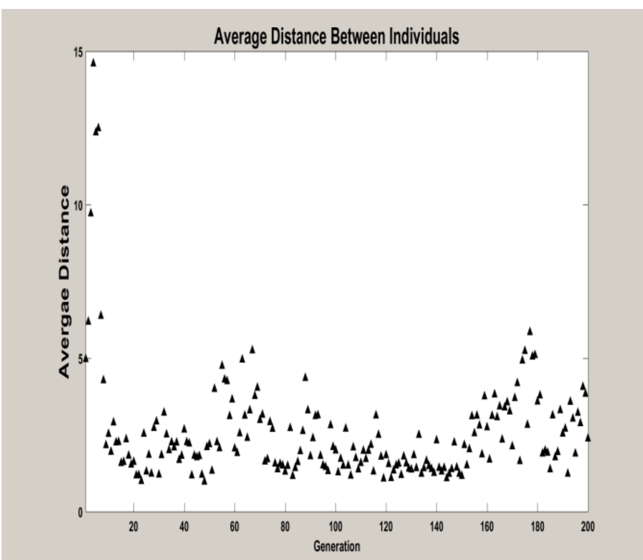


Figure 3. The average distance between the individuals competing for best schedule

Table IV shows a comparison of the performance results of the four methods. Here $\boldsymbol{f}_{\text{opt}}$ is the average optimum solution from 50 trials in each method. As we can see, all methods solved all the given problem instances and the results did not violate any of the hard constraints in all periods. Moreover, The GA approach generated schedules with optimal cost in all periods, compared to the manual schedules, the hill climbing technique and the GA with simulated annealing approach. The average execution time T over all periods of the GA is around 3.37 minutes, which is much faster than all other methods. Figure 5 plots the cost function of each method for comparison purposes over the 4 weeks. As can be seen in the plot our GA with a cost bit matrix is very effective compared to the other methods based on time and minimum constrains violation.

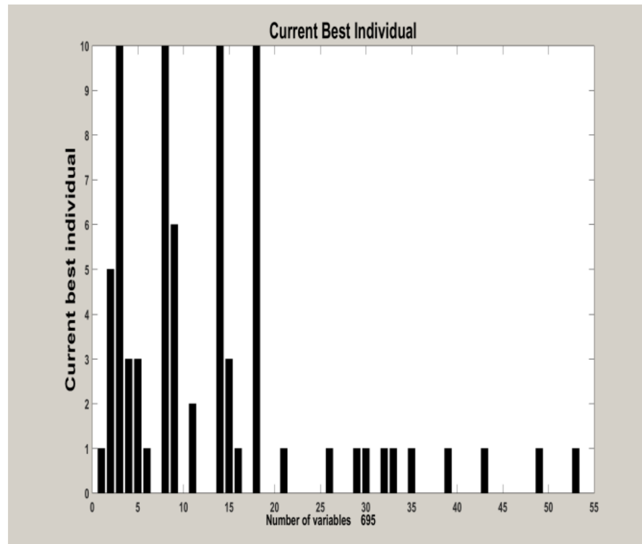


Figure 4. The Current best individual for DSP with 695 variables.

TABLE III. GROUPS AFTER APPLYING HILL CLIMBING, DISTRIBUTION OF R DOCTORS IN 6 WARDS

| **Group** | **Doctors** |
|---|---|
| G1 | $R1_3$, $R3_4$, $R4_2$, $R4_3$, $R3_1$ |
| G2 | $R4_1$, $R4_4$, $R3_3$, $R3_2$ |
| G3 | $R1_4$, $R1_2$, $R1_1$, $R2_1$, $SUB_3$ |
| G4 | $R1_5$, $R2_5$, $SUB_4$, $R2_6$, $R2_4$ |

| **A** | **B** | **C** | **D** | **E** | **F** |
|---|---|---|---|---|---|
| $R3_1$ | $R4_1$ | $SUB_1$ | $R2_4$ | $R1_5$ | $R1_1$ |
| $R4_2$ | $R3_2$ | $R2_1$ | $SUB_2$ | $R2_5$ | $R1_2$ |
| $R4_3$ | $R3_3$ | $R2_2$ | $R1_3$ | $R2_6$ | $SUB_3$ |
| $R3_4$ | $R4_4$ | $R2_3$ | $R2_7$ | $SUB_4$ | $R1_4$ |

TABLE IV. COMPARISONS OF GA AND OTHER METHODS ON PSMMC HOSPITAL DOCTOR SCHEDULE

| period | Method | $f_{opt}$ | T (min) |
|---|---|---|---|
| 1 week | GA | 2 | 2.6 |
| | Hill Climbing | 3 | 3.9 |
| | Manual | 7 | |
| | GA and simulated annealing | 3 | 3.5 |
| 2 Weeks | GA | 2 | 3.2 |
| | Hill Climbing | 6 | 4 |
| | Manual | 9 | |
| | GA and simulated annealing | 5 | 3.9 |
| 3 weeks | GA | 3 | 3.75 |
| | Hill Climbing | 5 | 4 |
| | Manual | 12 | |
| | GA and simulated annealing | 6 | 4.5 |
| 4 weeks | GA | 3 | 3.96 |
| | Hill Climbing | 5 | 5 |
| | Manual | 10 | |
| | GA and simulated annealing | 4 | 4.5 |

## V. CONCLUSION AND FUTURE WORK

In this paper, we proposed a Genetic Algorithm approach with a cost bit matrix to solve a Doctor Schedule Problem in PSMMC hospital. We designed a cost bit matrix which resulted in pruning of the search space and that was the main cause of reduction in execution time. In addition, the result of pruning increased the possibility to find feasible solution, which made our algorithm find solutions satisfying all the constraints. This automated GA approach generated a doctor roaster faster in time and better in quality with the least constraints violations than the manual method and the other considered computational methods. Although we have presented this work in terms of doctor scheduling, it should be noted that the main idea of the approach could be applied to many other scheduling problems. Our future plans include producing a software program coded by our proposed GA method with a user manual and friendly interface that can be used in hospitals to help them design schedules according to their constrains for their doctors and nurses. The software will allow the hospital management to enter the number of doctors, days, days off, night shifts, constrains, … etc., with simple inputs and less time, and hence produce an automated optimum doctor or nurse schedule in a few minutes and avoid manual schedule making, and tedious work when faced with the need for sudden modifications in these schedules.

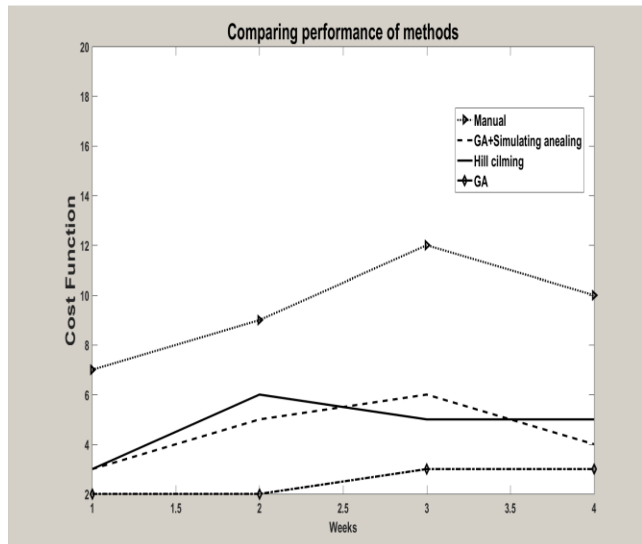


Figure 5. Comparison of performance ($f_{opt}$) of the four methods on PSMMC hospital doctor schedule

## ACKNOWLEDGMENT

This research project was supported by a grant from the "Research Centre of the Female Scientific and Medical Colleges", Deanship of Scientific Research, King Saud University.

## REFERENCES

[1] A. Alharbi and K. AlQahtani, "A Genetic Algorithm solution for the Doctor Scheduling Problem", ADVCOMP 5/COMARA: Computational Mathematics in Real-life Applications The Tenth International Conference on Advanced Engineering Computing and Applications in Sciences, October 9 - 13, (2016)-Venice, Italy.

[2] S. Abdennadher and H. Schienker, "Nurse Scheduling using Constraint Logic Programming", Proc. AAAI '99/IAAI '99, (1999), pp. 838-843.

[3] U. Aickelin and K.A. Dowsland, "An indirect Genetic Algorithm for a Nurse-Scheduling", Computers & Operations Research, vol. 31, no. 5, (2004) April, pp. 761-778.

[4] U. Aickelin and K.A. Dowsland, "Exploiting Problem structure roistering problem", Journal of Scheduling, vol. 3, (2000), pp. 139-153.

[5] G. Baskaran, A. Bargiela and R. Qu, “Hierarchical Method for Nurse Rostering Based on Granular Pre-Processing of Constraints,” The 23rd EUROPEAN Conference on Modelling and Simulation, Madrid, 9-12 June 2009, pp. 855-861.

[6] A. Brezulianu, F. Monica, and F. Lucian, “A Genetic Algorithm Approach for a Constrained Employee Scheduling Problem as Applied to Employees at Mall Type Shops”, IJAST, vol. 14, (2010), pp. 1-14.

[7] B. M. W Cheng, J. H. M Lee, and J. C. K Wu,” A constraint-based Nurse Rostering system using a Redundant modeling Approach”, IEEE Transactions on Information Technology in Biomedicine, (1997), pp. 44 - 54.

[8] A. Colorni, M. Dorigo, and V. Maniezzo, “A Genetic Algorithm To Solve The Timetable Problem”, computational optimization and applications Journal (1994),

[9] J. Culberson and Gent, “Frozen development in graph colouring”, Theoretical Computer Science. (2001), 265, 1-2, pp.227-264

[10] A. Gideon, “A Nurse Scheduling Using Graph Coloring”, Master thesis submitted, Mathematics Department, Kwame Nkrumah University, Ghana, (2013).

[11] D. E. Goldberg, “Genetic Algorithms in Search, Optimization and Machine Learning”, Addison Wesley (1989) .

[12] J. Istvan, A. Toth, and J. I. van Hemert, “Binary Merge Model Representation of the Graph Colouring Problem”, Evolutionary Computation in Combinatorial Optimization (2004), pp. 124-130.

[13] S. Kundu, M. Mahato, B. Mahanty, and S. Acharyya, “Comparative Performance of Simulated Annealing and Genetic Algorithm in Solving Nurse Scheduling Problem”, Proc. Int’l Multi Conference of Engineers and Computer Scientists 2008, (2008) January, pp. 1-5.

[14] J. Li and U. Aickelin, “A Bayesian Optimization Algorithm for the Nurse Scheduling”, Evolutionary Computation, (2003). In: CEC ’03.

[15] Z. Michalewicz, “Genetic Algorithms+Data Structures= Evolution Programs”, 3rd edition, Springer-Verlag, (1996).

[16] H. E. Miller, W. P. Pierskalla, and G. J. Rath, “Nurse Scheduling using Mathematical Programming”, Operations Research, vol. 24, no. 5, (1976), pp. 857-870.

[17] R. Ratnayaka, Z. Wang, S. Anamalamudi and S. Cheng, "Enhanced Greedy Optimization Algorithm with Data Warehousing for Automated Nurse Scheduling System," E-Health Telecommunication Systems and Networks, Vol. 1 No. 4, (2012), pp. 43-48.

[18] R. A. Namoco, and R. G. Salazar, “Solving the Nurse Scheduling Problem of Private Hospitals in the Philippines using Various Operators for Genetic Algorithm”, Indian Journal of Science and Technology, (2016), Vol 9(47), pp. 2-17.

[19] M. Silver, T. Sakuta , H. C. Su, S. B. Dolins, and M. J Oshea, “Case Study: How to Apply Data Mining Techniques in a Healthcare Data Warehouse,” Journal of Healthcare Information Information Management, (2001) Vol. 15, No. 2, pp. 155-164.

[20] S. N. Sivanandam, S. N. Deepa, “Introduction to Genetic Algorithms”, Springer, (2007).

[21] P. Tormos, A. Lova, F. Barber, L. Ingolotti, M. Abril, and M.A. Salido, “A Genetic Algorithm for Railway Scheduling Problems”, In: Xhafa F., Abraham A. (eds) Metaheuristics for Scheduling in Industrial and Manufacturing Applications. Studies in Computational Intelligence, (2008) vol 128. Springer, Berlin, Heidelberg.

[22] P. Villiers, “Clinical Data Warehouse Functionality,” SAS Institute Inc., New Caledonia, (1998).

[23] D. M. Warner and J. Prawda, “A Mathematical Programming Model for Scheduling Nursing Personnel in a Hospital”, Management Science, vol. 19 (4-Part-1), (1972) December, pp. 411-422.

[24] J. Wojciech and W. Szymon, “Efficient Greedy Algorithm with Hill Climbing for Technicians and Interventions Scheduling Problem”, Roadef chalange, France, (2007).

| Department of Paediatrics | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Paediatric ICU Team | | | | | | | | | | |
| Division Mobile: 0504585767 | | Feb 2016 | | | | | | | | |
| | | 1-6 PICU On- call Team A 08:00-08:00 | | | 3-1 PICU service Team B 08:00 – 16:00 | | | PRRT & Transportation 08:00-08:00 Team C | |
| Date | Day | Consultant | Fellow / Registrar | Resident | Consultant | Fellow | Registrar/Resident | Consultant | Fellow /Registrar |
| 1 | Mon | Chehab | Rizwan | ELGAWHARAH | Mohaimeed | Rizwan | NOUR S | Mohaimeed | Rizwan |
| 2 | Tues | Chehab | Warwar | QASSIM | Mohaimeed | Rizwan | NOUR S | Mohaimeed | Rizwan |
| 3 | Wed | Chehab | Yacoub | M. ASIRI | Mohaimeed | Rizwan | NOUR S | Mohaimeed | Rizwan |
| 4 | Thu | Mohaimeed | Inayat | GRACE | Mohaimeed | Inayat | GRACE | Mohaimeed | Inayat |
| 5 | Fri | Mohaimeed | Rizwan | TAGHREED | Mohaimeed | Rizwan | TAGHREED | Mohaimeed | Rizwan |
| 6 | Sat | Mohaimeed | Ahmed | ELGAWHARAH | Mohaimeed | Ahmed | ELGAWHARAH | Mohaimeed | Ahmed |
| 7 | Sun | Thabet | Warwar | AMAL | Bafaqih | Warwar | MALIK | Bafaqih | Warwar |
| 8 | Mon | Thabet | Yacoub | NADA ALHARBI | Bafaqih | Warwar | MALIK | Bafaqih | Warwar |
| 9 | Tues | Thabet | Inayat | RAED | Bafaqih | Warwar | MALIK | Bafaqih | Warwar |
| 10 | Wed | Thabet | Rizwan | BODOUR | Bafaqih | Warwar | MALIK | Bafaqih | Warwar |
| 11 | Thu | Bafaqih | Yaser | MUJAHID | Bafaqih | Yaser | MUJAHID | Bafaqih | Yaser |
| 12 | Fri | Bafaqih | Warwar | NADA | Bafaqih | Warwar | NADA | Bafaqih | Warwar |
| 13 | Sat | Bafaqih | Yacoub | ESRAA M | Bafaqih | Yacoub | ESRAA M | Bafaqih | Yacoub |
| 14 | Sun | Mohaimeed | Inayat | EBTISAM | Chehab | Inayat | HAMDAN | Chehab | Inayat |
| 15 | Mon | Mohaimeed | Rizwan | TAGHREED | Chehab | Inayat | HAMDAN | Chehab | Inayat |
| 16 | Tues | Mohaimeed | Yaser | NADA | Chehab | Inayat | HAMDAN | Chehab | Inayat |
| 17 | Wed | Mohaimeed | Yacoub | QASSIM | Chehab | Inayat | HAMDAN | Chehab | Inayat |
| 18 | Thu | Thabet | Ahmed | ESRAA M | Thabet | Ahmed | ESRAA M | Thabet | Ahmed |
| 19 | Fri | Thabet | Inayat | MUJAHID | Thabet | Inayat | MUJAHID | Thabet | Inayat |
| 20 | Sat | Thabet | Warwar | SARAH F | Thabet | Warwar | SARAH F | Thabet | Warwar |
| 21 | Sun | Bafaqih | Rizwan | RAED | Thabet | Yaser | Abdullah S | Thabet | Yaser |
| 22 | Mon | Bafaqih | Ahmed | TAGHREED | Thabet | Yaser | Abdullah S | Thabet | Yaser |
| 23 | Tues | Bafaqih | Inayat | AMAL | Thabet | Yaser | Abdullah S | Thabet | Yaser |
| 24 | Wed | Bafaqih | Warwar | QASSIM | Thabet | Yaser | Abdullah S | Thabet | Yaser |
| 25 | Thu | Chehab | Rizwan | NADA ALHARBI | Chehab | Rizwan | FATIMAH | Chehab | Rizwan |
| 26 | Fri | Chehab | Yacoub | EBTISAM | Chehab | Yacoub | HALA | Chehab | Yacoub |
| 27 | Sat | Chehab | Yaser | MUJAHID | Chehab | Yaser | NADA ALHARBI | Chehab | Yaser |
| 28 | Sun | Chehab | Inayat | QASSIM | Mohaimeed | Ahmed | Yosef | Mohaimeed | Ahmed |
| 29 | Mon | Chehab | Warwar | M. ASIRI | Mohaimeed | Ahmed | Yosef | Mohaimeed | Ahmed |

Figure 1. Example of a Hospital Manual Doctor schedule PSMMC hospital

TABLE V. THE HOSPITAL INCIDENT MATRIX FOR DOCTORS WORKING IN SAME GROUP AND IN 6 WARD FOR N=24.

| | $R3_1$ | $R4_1$ | $R1_1$ | $R1_5$ | $R2_4$ | $SUB_1$ | $R3_2$ | $R4_2$ | $R1_2$ | $R2_1$ | $R2_5$ | $SUB_2$ | $R3_3$ | $R4_3$ | $R1_3$ | $R2_2$ | $R2_6$ | $SUB_3$ | $R3_4$ | $R4_4$ | $R1_4$ | $R2_3$ | $R2_7$ | $SUB_4$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $R3_1$ | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $R4_1$ | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $R1_1$ | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $R1_5$ | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $R2_4$ | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $SUB_1$ | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $R3_2$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $R4_2$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $R1_2$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $R2_1$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $R2_5$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $SUB_2$ | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $R3_3$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $R4_3$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $R1_3$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $R2_2$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $R2_6$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| $SUB_3$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| $R3_4$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| $R4_4$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| $R1_4$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| $R2_3$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| $R2_7$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| $SUB_4$ | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |

TABLE VI. THE GENETIC ALGORITHM BEST DOCTOR SCHEDULE N=24, D=29 FOR PSMMC HOSPITAL DOCTOR SCHEDULE

| Days/ Junior doctors | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| J1 | m | e | n | o | m | m | n | m | e | n | o | m | m | n | m | e | n | o | m | m | n | m | e | n | o | m | m | n | m |
| J2 | e | m | m | n | o | m | n | e | m | m | n | o | m | n | e | m | m | n | o | m | n | e | m | m | n | o | m | n | e |
| J 3 | n | n | o | m | e | e | e | n | n | o | m | e | e | e | n | n | o | m | e | e | e | n | n | o | m | e | e | e | n |
| J 4 | m | m | e | e | n | o | m | m | m | e | e | n | o | m | m | m | e | e | n | o | m | m | m | e | e | n | o | m | m |
| J 5 | e | e | e | n | m | n | o | e | e | e | n | m | n | o | e | e | e | n | m | n | o | e | e | e | n | m | n | o | e |
| J 6 | m | e | n | o | m | m | n | m | e | n | o | m | m | n | m | e | n | o | m | m | n | m | e | n | o | m | m | n | m |
| J 7 | e | m | m | n | o | m | n | e | m | m | n | o | m | n | e | m | m | n | o | m | n | e | m | m | n | o | m | n | e |
| J 8 | n | n | o | m | e | e | e | n | n | o | m | e | e | e | n | n | o | m | e | e | e | n | n | o | m | e | e | e | n |
| J 9 | m | m | e | e | n | o | m | m | m | e | e | n | o | m | m | m | e | e | n | o | m | m | m | e | e | n | o | m | m |
| J 10 | e | e | e | n | m | n | o | e | e | e | n | m | n | o | e | e | e | n | m | n | o | e | e | e | n | m | n | o | e |
| J 11 | m | e | n | o | m | m | n | m | e | n | o | m | m | n | m | e | n | o | m | m | n | m | e | n | o | m | m | n | m |
| J 12 | e | m | m | n | o | m | n | e | m | m | n | o | m | n | e | m | m | n | o | m | n | e | m | m | n | o | m | n | e |
| J 13 | n | n | o | m | e | e | e | n | n | o | m | e | e | e | n | n | o | m | e | e | e | n | n | o | m | e | e | e | n |
| J 14 | m | m | e | e | n | o | m | m | m | e | e | n | o | m | m | m | e | e | n | o | m | m | m | e | e | n | o | m | m |
| J 15 | e | e | e | n | m | n | o | e | e | e | n | m | n | o | e | e | e | n | m | n | o | e | e | e | n | m | n | o | e |
| J 16 | m | e | n | o | m | m | n | m | e | n | o | m | m | n | m | e | n | o | m | m | n | m | e | n | o | m | m | n | m |
| J 17 | e | m | m | n | o | m | n | e | m | m | n | o | m | n | e | m | m | n | o | m | n | e | m | m | n | o | m | n | e |
| J 18 | n | n | o | m | e | e | e | n | n | o | m | e | e | e | n | n | o | m | e | e | e | n | n | o | m | e | e | e | n |
| J 19 | m | m | e | e | n | o | m | m | m | e | e | n | o | m | m | m | e | e | n | o | m | m | m | e | e | n | o | m | m |
| J 20 | e | e | e | n | m | n | o | e | e | e | n | m | n | o | e | e | e | n | m | n | o | e | e | e | n | m | n | o | e |
| J 21 | m | e | n | o | m | m | n | m | e | n | o | m | m | n | m | e | n | o | m | m | n | m | e | n | o | m | m | n | m |
| J 22 | e | m | m | n | o | m | n | e | m | m | n | o | m | n | e | m | m | n | o | m | n | e | m | m | n | o | m | n | e |
| J 23 | n | n | o | m | e | e | e | n | n | o | m | e | e | e | n | n | o | m | e | e | e | n | n | o | m | e | e | e | n |
| J 24 | m | m | e | e | n | o | m | n | m | e | e | n | o | m | n | m | e | e | n | o | m | m | m | e | e | n | o | m | m |