

Hybrid Genetic Algorithmic Approaches for Personnel Timetabling and Scheduling Problems in Healthcare

Amol C. Adamuthe
Selection Grade Lecturer,
Dept. of IT, College of
Engineering Pandharpur,
Pandharpur, Dist. Solapur,
MS, India.

Rajankumar Bichkar
Professor, Dept. of Computer
Engg, G. H. Rasoni College
of Engg. & Mgmt, Pune. MS,
India.

ABSTRACT

This paper presents a genetic algorithmic approach to the solution of the problem of personnel timetabling in laboratories in which the objective is to assign tasks to employees and nurse scheduling in medical centre where the objectives are to assign staff to particular day in planning period and minimization of personnel cost by avoiding overtime pay. The personnel scheduling and timetabling problems are multi-constrained and having huge search space which makes them NP hard. Genetic algorithmic approach is applied to both the problems. Canonical genetic algorithm demonstrates very slow convergence to optimal solution. Hence, in laboratory personnel timetabling problem a knowledge augmented operator is introduced in genetic algorithm framework. This hybridization helps to get the near-optimal solution quickly. For nurse scheduling problem, proposed hybrid genetic algorithms with partial feasible chromosome representation, initialization and operators have shown fast convergence towards optimal solution with comparatively small population size. The probability of getting near optimal solution using proposed hybrid genetic algorithm in less than 20 seconds (the average time) is more than 0.6. Timetabling and scheduling problems under consideration are quite different from each other. Hence choice of genetic operators and parameters for both the problems are different. Finding a general framework for timetabling and scheduling problems is still a challenge.

General Terms

Algorithms, Combinatorial Optimization, Scheduling Problems.

Keywords

Personnel Scheduling, Nurse Scheduling, Scheduling, Timetabling, Genetic Algorithm, Hybrid Genetic Algorithms.

1. INTRODUCTION

Personnel scheduling problem is mostly encountered in service organizations such as call centres, airport ground personnel, security agencies, hospitals, railway and bus personnel. Patrick De Causmaecker [1] has classified personnel scheduling problem in four planning categories namely, permanence centred, mobility centred, fluctuation centred and project centred planning. In [2] Ernst et al. presented review on staff scheduling and rostering with respect to applications, methods and models. The main difficulties in solving these problems are their highly constrained nature and the environmental conditions that are different for each

organization such as working hours, planning periods, existence of breaks for employees, existence of part-time employees in addition to full-time ones etc.

Research in personnel scheduling focuses on three main objectives.

- 1) Allocation of personnel to shifts, for example nurse scheduling [3, 4, 5, 6, 7], hospital personnel scheduling [8].
- 2) Assignment of tasks to personnel, for example laboratory personnel scheduling [9, 10].
- 3) Minimization of personnel cost, for example personnel scheduling on ship [11].

First objective has received more importance in recent years as compared to other two.

In literature, the terms ‘scheduling’ and ‘timetabling’ are used interchangeably. According to Wren [12]: “Scheduling is the allocation, subject to constraints, of resources to objects being placed in space-time, in such a way as to minimize the total cost of some set of the resources used”. Common examples of scheduling are driver scheduling [13, 14] which seek to minimize the total cost and job shop scheduling [15, 16, 17] which may seek to minimize the number of time periods used or some physical resources. “Timetabling is the allocation, subject to constraints, of given resources to objects being placed in space-time, in such a way as to satisfy as nearly as possible a set of desirable objectives”. Examples of timetabling are class timetabling [18, 19] and examination timetabling [20] and some forms of personnel allocation.

NP hard and NP complete problems can not be solved in reasonable time with traditional search and optimization methods. Genetic algorithms and hybrid genetic algorithms are proven to be one of the effective techniques to solve different scheduling problems [3, 6, 7, 10, 13, 18, 19, 21].

In this paper, we are presenting a steady state genetic algorithm and genetic algorithm with repair operator to problem of assignment of tasks to personnel in particular, the laboratory personnel timetabling. The problem instance under consideration

is that of Philip and Post [9] with small variation. Our paper [10] has shown significant improvement in the quality of obtained solution. The second problem we are tackling is that of staff scheduling in hospital also called as nurse scheduling. The constraints and data considered are those used at the Shawnee Mission Medical Center. Main objective is allocation of nurse to days in given planning period and second objective is minimization of personnel cost by avoiding overtime pay. Dean [22] has attempted this nurse scheduling problem using genetic algorithmic approach with string and matrix chromosome representation. We have tackled the problem by standard genetic algorithm and hybrid genetic algorithm. Standard genetic algorithm uses 2D chromosome representation with standard selection, crossover and mutation operators. The proposed hybrid genetic algorithm incorporates problem specific knowledge in chromosome representation, initialization and operators. The suggested chromosome representation and operators avoid violations of some of the constraints.

The rest of the paper is organized as follows: In section 2 genetic algorithms is briefly described. Section 3 is about problem statements of personnel timetabling and scheduling problems in healthcare industry. Section 4 gives the implementation details that include genetic algorithms, chromosome representation and operators. In Section 5 we give a detailed description of the problem instances and results. Finally, in Section 6 we outline the conclusions of our study.

2. GENETIC ALGORITHMS

Genetic Algorithms (GAs) are randomized yet structured search and optimization algorithms based on the evolutionary ideas of natural selection and genetics [23]. Genetic algorithms exploit historical information to direct the search into the region of better solutions within the search space. They do not require problem specific information for their working. Hence genetic algorithms are used to solve highly constrained, combinatorial optimization problems having huge search space.

GAs simulates the survival of the fittest among individuals over consecutive generations for solving a problem. Each generation consists of a population of individuals. Each individual represents a point in a search space and a possible solution. The individuals in the population are then made to go through a process of evaluation.

GAs are based on the following foundations:

1. Individuals in a population compete for resources.
2. Individuals with above average population fitness produce more offspring than those individuals that are below average population fitness. This indicates exponential speed up in the search process.
3. Genes from “good” individuals propagate throughout the population so that two good parents will sometimes produce offspring that are better than either parent.
4. Thus, each successive generation will become more suited to their environment.

After an initial population is randomly generated, the algorithm evolves through three operators:

1. Selection which equates the survival of the fittest.
2. Crossover which represents mating between individuals.
3. Mutation which introduces random modification.

Strengths of Genetic Algorithm

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

3. PROBLEM STATEMENTS

This section describes two scheduling problems required in health care industry.

3.1 Laboratory Personnel Timetabling Problem

The timetabling problem we are presenting here is tackled by Philip and Post [9] and Adamuthe and Bichkar [10]. It is an assignment problem in which the required numbers of tasks are known in advance and the employees are to be assigned to tasks by satisfying constraints. The solution of this problem requires the following characteristics of personnel and tasks to be taken into consideration.

- 1) Characteristics of Personnel :
 - Skill set
 - Work regulations (full time or part time employee), and
 - Availability on planning period,
- 2) Characteristics of Tasks :
 - Category (half day, day task or week task),
 - Priority,
 - Requirement on planning period,
 - Dependency with other tasks, and
 - Rotation of tasks among personnel.

3.1.1 Hard Constraints

Hard constraints are those that must be satisfied. Violation of these constraints (also called as conflicts) will cause the solution to be infeasible.

1. Coverage constraint: Every task must be allotted the required number of personnel.
2. Constraints by work regulations: Number of work hours assigned to personnel must satisfy his/her work regulations
3. Skill set constraint: Task should not be assigned to an employee who is not skilled for it.
4. Constraints defined by task types: Tasks are categorized into three types namely, half-day task, day task and week task. If the task is week task then it must be

assigned to same person for the entire week. Whereas, day task is given to same person in both slots (morning and evening) and half day task is to be assigned to different personnel during morning and afternoon slots of a day.

3.1.2 Soft Constraints

Soft constraints are those that are desirable in order to produce a good quality timetable but violations are allowed to satisfy hard constraints. In this problem, the task assignment should be according to the skill set of employees. Here history cost (that is number of hours worked by an employee on a task) is considered as skill set.

3.2 Nurse Scheduling in Medical Centre

This section describes the hard and soft constraints for nurse scheduling problem described by Dean [22] for Shawnee Mission Medical Center.

3.2.1 Hard Constraints

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

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

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

HC3: Each employee works at least three days per week (=39 hours).

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

HC5: Each employee works every third weekend.

3.2.2 Soft Constraints

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

Cost Minimization Constraints:

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

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

Personal Demands:

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

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

General:

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

3.2.3 Constraints depends on prior schedules

Hard constraints HC4, HC5 and soft constraints SC3, SC4 require values from prior schedule (See Table V). Array priorConsecutiveDays (pcd) gives details of the number of consecutive days worked by an employee prior to the start of the current schedule which is essential for constraints HC4, SC3 and SC4. The array contains values 0, 1, 2 or 3.

Array numOfWeekendsAgo (nwa) stores the number of weekends ago the employee worked prior to the start of the current schedule which is required for HC5. It contains values 0, 1 or 2, where 1 indicates that employee had worked 1 weekend ago. The elements are assigned values cyclically- 0, 1, 2, 0, 1, 2, 0 and so on.

3.2.4 Constraints not under consideration

1. Higher skill nurses substitute lower skill level nurses.
2. Employee shift was fixed for the duration of his or her employment at the hospital.

4. GENETIC ALGORITHMIC APPROACHES FOR PERSONNEL TIMETABLING AND SCHEDULING

One of the popular implementation of genetic algorithm is GALib [24], a C++ library of Genetic Algorithm Components developed by Matthew Wall at Massachusetts Institute of Technology. The GALib source code is available at no cost for non-profit purposes. It is having following features:

- 1) Supports most major operating systems and compilers.
- 2) Four different types of genetic algorithms are available- simple GA, steady-state GA, incremental GA and deme GA.
- 3) Contains many chromosome representations and selection, crossover and mutation operators.
- 4) Customization of genetic algorithms, chromosomes and operators is possible.

A simple genetic algorithm uses overlapping populations and optional elitism. The steady state GA uses overlapping populations with the population overlap as an user specified parameter. In incremental GA, only one or two children are created in each generation. The deme GA uses multiple populations in parallel using steady state algorithm. This section describes our implementation that uses steady state genetic algorithm, the chromosome representation, objective function and the genetic operators employed.

4.1 Steady State Genetic Algorithm

Steady state genetic algorithm is used to solve the problems described in section 3. The steady-state genetic algorithm uses overlapping populations. In each generation, the original population size is maintained by replacing a portion of the population by the newly generated individuals.

4.2 Chromosome Representation

In this work we have used direct 2D representation for both the problems. In the first representation as shown in Figure 1, a row represents a personnel and a column represents one day of planning period. For laboratory personnel timetabling problem,

the cells contain the task value. Depth of the search tree is multiple of number of employees (P) and planning period (D). For each assignment alternatives are available are equal to number of tasks (T). This search tree contains $(P*D)! * T^{(P*D)}$ nodes. For nurse scheduling problem, the cells contain value one or zero. Value 1 indicates staff is assigned for respective day and zero indicates off. Depth of the search tree is multiple of number of employees (P) and planning period (D). For each assignment two alternatives are available. This search tree contains $(P*D)! * 2^{(P*D)}$ nodes. One of the problems associated with this representation is that it may generate infeasible solutions. We have tackled this by imposing penalty and knowledge augmented repair operator on such infeasible solutions.

Employee	Day of Planning Period		
	Day 1	...	Day D
P_1		...	
P_2		...	
:	:	:	:
P_N		...	

Fig 1: Chromosome representation for Steady State GA.

Figure 2 shows the solution representation used in hybrid genetic algorithm to solve nurse scheduling problem. In this representation, a row represents a personnel and a column represents number of working days for each employee during planning period (W) which is same for each employee. The cells contain number of day. For example staff P_1 is working on day 4 (d_4). As per this solution representation depth of the search tree is multiple of number of employees (P) and number of working days for each employee during planning period (W). For each assignment D alternatives are available. This search tree contains $(P*W)! * D^{(P*W)}$ nodes.

Employee	Number of Working Days for each Employee during Planning Period		
	1	...	D
P_1	d_3	...	d_4
:	:	:	:
P_N	D_8	...	d_1

Fig 2: Chromosome representation for Hybrid GA for NSP.

4.3 Objective Function

The objective score of an individual is computed by assigning penalty costs (penalty points) for the violations of constraints. These penalty costs are specified according to the importance of the constraint as shown in Table 1 and Table 2. These values ensure that hard constraints are not violated in compensation for multiple soft constraints. The objective function for individual ‘ t ’ is calculated as follows.

$$O(t) = \sum \alpha_i H_i + \sum \beta_j S_j \quad (1)$$

where H_i and S_j represents number of hard and soft constraints violated by a solution for i^{th} hard constraint and j^{th} soft constraint respectively. α_i and β_j represents penalty costs for violation of hard constraint and soft constraints respectively. Objective values are negative numbers. Steady state Genetic algorithm does not work with negative objective values. Hence we have used sigma truncation scaling to convert them into positive fitness score.

Table 1. Penalty points for constraint violations for Laboratory Personnel Timetabling

Constraints	Penalty Points
Coverage constraint	-150
Skill set constraint	-120
Constraints by work regulations	-100
Constraints defined by task types for full timers	-30
Constraints defined by task types for part timers	-10
Soft Constraint for skill set	-1

Table 2. Penalty points for constraint violations for NSP

Constraint	Penalty Points	
	Steady State GA	Hybrid GA
For each day of the entire schedule, assign a constant number of working employees.	-200	-500
For each employee, assign at least 3 working days per week (=39 hours).	-150	Avoided
For each employee, maximum number of consecutive working days = 6.	-250	-250
Each employee works every third weekend.	-230	-230
Do not assign same employee more than once in a day	Not Applicable	-400
Try to avoid more than 3 working days per employee per week because that leads to overtime pay.	-50	Avoided
Try hard to avoid more than 4 working days per employee per week because that leads to additional overtime pay.	-70	Avoided
If an individual wants to avoid working 3 days in a row	-90	-70
If an individual wants to maximize the number of grouped working days	-0.1	-2
Try to balance the number of Monday and Friday off days that coincide with off weekends.	-1	-1

4.4 Genetic Operators

Selection of suitable operators with probabilities in genetic algorithms plays very important role.

4.4.1 Laboratory Personnel Timetabling

Different selection, crossover and mutation applicable to given chromosome representation are tested. Roulette wheel selection is found to be effective than tournament selection. One point crossover with probability 0.8 gives better performance than two point crossover, multi-point crossover and uniform crossover. Swap mutation with probability 0.01 works well than flip mutation. Uniform initializer is used for population initialization.

Repair Operator: During the evolutionary process, some genetic operators may generate illegal individuals (that do not respect the problem constraints). To improve the quality of the solution and convergence speed it was necessary to introduce operator that uses problem specific knowledge. The idea is to apply repair operator which outperform the simple penalization of constraint violations. This operator is applied after standard genetic operators but before fitness calculation. This operator removes wrong task assignments in the solution. The operator first finds wrong task assignments. It then searches for a suitable task which is assigned to a part time employee on the same day and swaps the two assignments. It may be noted that, this operator does not necessarily generate a completely feasible solution. We have tested another version of this repair operator where the infeasibilities are completely removed. However, it did not give as good results as the partially feasibility maintaining repair operator.

4.4.2 Nurse Scheduling Problem

Table 3 shows genetic operators used in steady state and hybrid genetic algorithms. Steady state GA works with standard uniform initializer, even odd crossover and flip mutation. Crossover and mutation operator are tested under different probability conditions. The best probability values are also shown in Table 3.

Hybrid genetic algorithm uses problem specific information. The chromosome representation allows infeasible solution but at the same time avoids some hard and soft constraints. Instead of random initialization problem specific partially feasible initialization is used to help genetic algorithm. Proposed single point crossover is partially feasible for defined problem. Table 4 shows list of avoided constraints by proposed solution representation, initialization and single point crossover. But the proposed hybrid genetic algorithm is having one overhead that is same employee should not be assigned more than once to any day in the schedule.

Table 3. Operators and Probabilities used for NSP

Operator Type	Steady State GA		Hybrid GA	
	Operator	Probability	Operator	Probability
Initialization	Uniform Initializer	-	Problem Specific	-
Crossover	Even Odd Point	0.9	Single point (Problem Specific)	0.9
Mutation	Flip	0.02	Swap (Problem Specific)	1.0

Table 4. Hybrid GA and Avoided Constraints

Step in GA	Constraints Avoided
Representation	HC3, SC1 and SC2.
Initialization	HC2 and HC3
Crossover	HC2, HC3, SC1 and SC2

5. RESULTS AND DISCUSSION

This section gives details of problem instances, results obtained by proposed genetic algorithms and comparison with existing results.

5.1 Laboratory Personnel Scheduling

We have solved the problem instance given by Philip and Post [9] with small variation. The laboratory has seven different tasks that should be assigned to personnel every day during the planning period. All the tasks are treated as week task that is same task must be allocated to a personnel during the week. The task names are AF, CT, IN, KB, PB, PR and TF. Two persons are required to carry out tasks PB and TF; whereas, all other tasks require only one person. For remaining tasks single personnel is sufficient. The laboratory has 14 employees and works in two shifts, morning and afternoon. Two types of work regulations are under consideration, full time employees who work 40 hours per week and part time employees who work 16 or 24 hours in a week. We have assumed that each employee is available on all working days and that the part time employees will work only on limited number of days. Number of days for which the personnel can be assigned to work during the planning period is shown in Table 5. Thus, employees A, E, F, J, L and M will work on 3 days per week; whereas, employees C, D, I and N will work on 2 days per week. Employees B, G, H and K are fulltime employees.

Table 5 also shows history cost that is number of hours worked by employees on each task during last 15 weeks. History cost is treated as skill set for employees. The number of hours employee worked for any task represents expertise and willingness of employee for that job. Higher the history cost, more the expertise for that task.

Table 5. History cost (skill set) data used by Philip and Post [9].

	n^\dagger	AF	CT	IN	KB	PB	PR	TF
A	3	64	88	-	64	104	64	96
B	5	-	160	-	80	136	72	124
C	2	64	-	-	32	48	36	68
D	2	-	-	-	-	80	48	104
E	3	-	-	300	-	-	-	-
F	3	48	32	-	32	44	24	44
G	5	-	-	-	92	152	104	176
H	5	88	128	-	80	128	48	104
I	2	56	-	-	32	56	32	64
J	2	-	-	300	-	-	-	-
K	5	88	120	-	40	88	40	120
L	3	56	56	-	32	64	32	64
M	3	72	-	-	32	80	32	64
N	2	56	-	-	40	84	48	68

n^\dagger represents number of days the employee can be assigned for work. $n=5$ represents a full-time employee whereas other values of n represent part-time employees. The values in this column are obtained from results given in [9].

Philip and Post [9] have solved this problem by using bipartite graph model followed by local search for further improvement in quality of solution. They have reported a good solution that has only two violations that is assignment of two or more tasks in a week to employees A and L.

The proposed genetic algorithm has been applied to the problem instance and executed several times to test the effect of various parameters. The best task schedule obtained is shown in Table 6. One of the basic improvement obtained in this solution is that every employee has been assigned only a single task throughout the week.

Table 6. Task Schedule with zero violations

	MON	TUE	WED	THU	FRI
A	KB	KB	KB		
B	CT	CT	CT	CT	CT
C		TF		TF	
D			TF	TF	
E	IN		IN		IN
F	TF		TF		TF
G	PB	PB	PB	PB	PB
H	PB	PB	PB	PB	PB
I	AF			AF	
J		IN		IN	
K	PR	PR	PR	PR	PR
L	TF	TF			TF
M		AF	AF		AF
N				KB	KB

To further compare the quality of our result with that obtained by Philip and Post, we have determined an expertise quality index calculated as shown below.

$$E = \sum_e \sum_t S_{et} \quad (2)$$

where, S_{et} is the skill score of an employee 'e' for task 't'. These values are simply obtained by assigning integer values (starting from 1) to task for which an employee has experience. For example, consider employee C who has experience on tasks KB, PR, PB, AT and TF (having 32, 36, 48, 64 and 68 hours history cost respectively). These tasks are assigned values as 1, 2, 3, 4 and 5 respectively. This index for our solution is 127 and 98 for solution given by Philip and Post. This clearly indicates the superiority of the genetic algorithmic approach.

The effect of repair operator is demonstrated in Figure 3, where the best results in 10 runs (each started with random seed) are reported. In first 5000 generations genetic algorithm and hybrid genetic algorithm shows same convergence. Figure 3 shows objective values from generation number 5000 where we can see the fast convergence of hybrid genetic algorithm over standard genetic algorithm. The best initially generated random solution has objective value in the range of -5300 to -5900. The genetic algorithm without repair shows a good convergence speed in the initial part of the run that it moves from the region with objective value in between -5300 and 5900 to -200 to -500. However after 5000 generations, there is very little improvement which is a characteristic of genetic algorithm. Finally after 60000 generations we get a solution that still has -30 value (indicating soft constraint violations). On the other hand genetic algorithm with repair operator shows a considerable improvement in the solution. A solution having objective value -10 is obtained in less than 10000 generations only. However even after executing upto 60000 generations there was no further improvement.

Proper balance between genetic algorithm and repair operator is necessary for better exploitation and exploration of the search space. Figure 4 shows performance of hybrid genetic algorithm (GA + repair) for different repair operator activation frequencies. Repair operator is applied to 10% of the population.

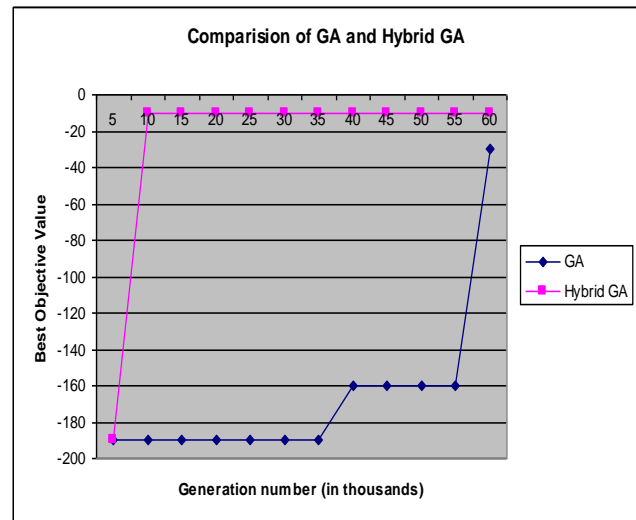


Fig 3: Effect of repair operator in GA convergence for Laboratory Personnel Timetabling.

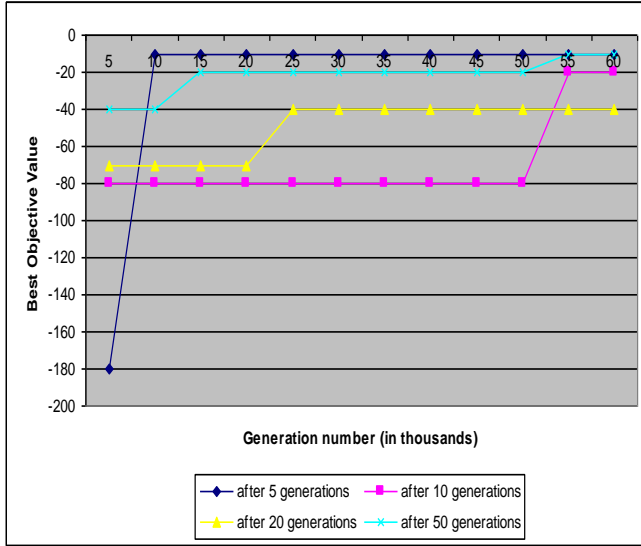


Fig 4: Performance of hybrid genetic algorithm (GA + repair) for Laboratory Personnel Timetabling with different repair activation frequencies.

5.2 Nurse Scheduling Problem

The problem instance used for testing involves 14 nurses out of which 6 are needed each day during planning period. Out of 14 nurses, 5 nurses want to maximize grouped working days and 5 nurses want to avoid 3 consecutive working days. The proposed steady state and hybrid genetic algorithms have been applied to the problem instance and executed several times to tune the various parameters of genetic algorithm such as population size, probability of crossover and mutation. The sample task schedule obtained is shown in Table 7. Genetic algorithms are tested with different seed values.

Table 8 shows comparison of proposed genetic algorithms and Dean's implementation [22] with respect to population size, average number of generations and average time required to obtain a good quality solution on 20 runs. Dean has suggested a population size 3003 for the problem instance to ensure that entire search space is available for genetic algorithm. Proposed genetic algorithms have shown significant improvement with small population size, 100 and 50 for steady state and hybrid genetic algorithms respectively. The average time required to get the near-optimal solution is 80 and 20 seconds for steady state and hybrid algorithms respectively. During experimentation we found that proposed genetic algorithm is dependent on the random start position. Hybrid genetic algorithm is based on partial feasible initialization. Results are obtained on 20 different random partially feasible initializations. 60% of the random partial feasible initializations results in near optimal solutions with less than average time.

Table 8. Comparison of Proposed Hybrid GA with [22]

Parameter	Value		
	Dean J. [22]	Steady State GA	Hybrid GA
Population Size	3003	100	50
Percentage of Population Overlapping	Elitism	20	20
Average time required (in seconds)	150	80	20

6. CONCLUSION

Results show that genetic algorithm is a powerful search technique to solve such multi-constrained timetabling and scheduling problems. The genetic algorithm with standard selection, crossover and mutation performs well for laboratory personnel timetabling and nurse scheduling problem. The main obstacle in performance of genetic algorithm is premature convergence. To avoid these problems in laboratory personnel scheduling we have introduced a repair operator which helps genetic algorithm to speed up by making intelligent local moves. The proposed GA has given a significant improvement over the results presented in [8]. Chromosome representation, initialization and operators proposed in Hybrid GA for nurse scheduling problem partially avoid hard and soft constraint violations. Proposed approach shows significant improvement with respect to solution quality and speed of convergence as compared to the approach used in [22]. The proposed partially feasible chromosome representation and operators may be applicable to large problems. We are in the process of running test on more complex problem instances.

7. ACKNOWLEDGMENTS

Amol Adamuthe thanks to Mr. John S. Dean, Information and Computer Science Department, Park University for providing his source code for nurse scheduling problem which is used for comparison with proposed genetic algorithm.

8. REFERENCES

- [1] Causmaecker, P. D., Demeester, P., Berghe, G. V., and Verbeke, B. 2004. Analysis of Real-world Personnel Scheduling Problems. In Proc. of 5th Practice and Theory of Automated Timetabling.
- [2] Ernst, A. T., Jiang, H., Krishnamoorthy, M., and Sier, D. 2004. Staff Scheduling and Rostering: A review applications, methods and models. European Journal of Operational Research. 153, 3-27.
- [3] Burke, E. K., Cowling, P., Causmaecker, P. D., and Berghe, G.V. 2001. A Memetic Approach to the Nurse Rostering Problem. J. Applied Intelligence. 15, 199-214.
- [4] Aickelin, U., and White, P. 2004. Building Better Nurse Scheduling Algorithms. J. Annals of Operations Research. 128, 159-177.

- [5] Burke, E. K., Curtois, T., Post, G., Qu, R., and Veltman, B. 2005. A Hybrid Heuristic Ordering and Variable Neighbourhood Search for the Nurse Rostering Problem. Technical Report. Nottingham University.
- [6] Ozcan, E. 2005. Memetic Algorithms for Nurse Rostering. In Proc. of 20th International Symposium on Computer and Information Sciences, 482-492.
- [7] Maenhout, B., and Vanhoucke, M. 2006. A Comparison and Hybridization of Crossover Operators for the Nurse Scheduling Problem. Working Papers of Faculty of Economics and Business Administration, Ghent University.
- [8] White, C. A., and White, G. M. 2002. Scheduling Doctors for Clinical Training Unit Rounds Using Tabu Optimization. In Proc. of 4th Practice and Theory of Automated Timetabling, 120-128.
- [9] Franses, P., and Post, G. 2002. Personnel Scheduling in Laboratories. In Proc. of 4th Practice and Theory of Automated Timetabling, 113-119.
- [10] Adamuthe, A. C., and Bichkar, R. S. 2011. Genetic algorithmic approach for personnel timetabling. In Proc. of International Conference on Technology Systems and Management 2011, 69-76, Springer- Verlag.
- [11] Schmidt, M. 1999. Solving Real-Life Time-Tabling Problems. In Proc. of 11th International Symposium ISMIS '99, 648-656.
- [12] Wren, A. 1996. Scheduling, Timetabling and Rostering - A Special Relationship? In Proc. of 1st Practice and Theory of Automated Timetabling, 46-75.
- [13] Li, J., and Kwan, R.S.K. 2003. A Fuzzy Genetic Algorithm for Driver Scheduling. J. European Journal of Operational Research. 147, 334-344.
- [14] Li, J., Kwan, R. S. K. 2005. A Self-Adjusting Algorithm for Driver Scheduling. J. Journal of Heuristics. 11, 351-367.
- [15] Miyashita, T. 2003. An Application of Immune Algorithms for Job-Shop Scheduling Problems. In Proc. of the 5th IEEE International Symposium on Assembly and Task Planning.
- [16] Jensen, M.T. 2003. Generating Robust and Flexible Job Shop Schedules Using Genetic Algorithms. J. IEEE Transactions on Evolutionary Computation. 7, 275-288.
- [17] Ombuki, B. M., and Ventresca, M. 2004. Local Search Genetic Algorithms for the Job Shop Scheduling Problem. J. Applied Intelligence. 21, 99-109.
- [18] Wilke, P., Grobner, M., and Oster, N. 2002. A Hybrid Genetic Algorithm for School Timetabling. -.
- [19] Karova, M. 2004. Solving Timetabling Problems Using Genetic Algorithms. Proc. of 27th Int'l Spring Seminar on Electronic Technology.
- [20] Qu, R., and Burke, E. K. 2005. Hybrid Variable Neighborhood HyperHeuristics for Exam Timetabling Problems. In Proc. of 6th Metaheuristics International Conference.
- [21] Adamuthe, A. C., and Bichkar, R. S. 2011. Minimizing Job completion time in grid scheduling with resource and timing constraints. In Proc. of International Conference and Workshop on Emerging Trends in Technology 2011, 338-343.
- [22] Dean, J. 2008. Staff Scheduling by a Genetic Algorithm with a Two-Dimensional Chromosome Structure. In Proc. of Practice and Theory of Automated Timetabling 2008.
- [23] Goldberg, D. E. 1989. Genetic Algorithms in Search, Optimization, and Machine Learning, Reading, Mass. Addison-Wesley.
- [24] Wall, M. 1996. GALib: A C++ Library of Genetic Algorithm Components. Massachusetts Institute of Technology.

Table 7. Solution for Nurse Scheduling Problem

Employee	pcd	nwa	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7							
			M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
1	1	0		1	1	1								1	1	1	1	1	1								1	1	1	
2	0	1	1	1	1					1	1	1									1	1	1	1	1	1				
3	0	2				1		1	1	1	1		1						1	1	1				1	1	1		1	
4	1	0			1		1	1				1	1		1		1	1			1	1			1	1		1		
5	0	1	1	1			1						1		1	1			1			1	1				1	1	1	
6	0	2	1	1	1									1	1	1	1	1	1									1	1	1
7	1	0	1	1			1			1	1		1				1	1		1				1	1			1		
8	0	1					1	1	1	1	1	1									1	1	1	1	1	1				
9	0	2				1		1	1			1		1	1				1		1	1				1	1	1		
10	2	0	1	1	1									1	1	1	1	1		1				1	1		1			
11	0	1					1	1	1	1	1	1								1	1	1					1	1	1	
12	0	2	1		1	1					1		1	1			1	1		1			1	1	1		1	1		
13	3	0				1	1		1	1			1	1					1		1	1			1	1		1	1	
14	0	1				1		1	1			1		1		1			1		1	1			1	1		1	1	