Original Article

Multi Objective Flexible Employee Scheduling for Pharmaceutical Industries Using Hybrid Genetic Algorithm

Mahalakshmi[1], Srinivasa G[2], Nidhi S Vaishnaw[3]

[1]Assistant Professor, Department of Mathematics, REVA University, Bangalore, Karnataka, India, Email: mahalakshmi4131@gmail.com

[2]Professor, Department of Mathematics, New Horizon College of Engineering, Bangalore, Karnataka, India , Email: gsrinivasa1974@gmail.com

[3]Associate Professor, CMR Institute of Technology, Karnataka, India , Email: nidhi.v@cmrit.ac.in

Abstract

In this paper a hybrid genetic algorithmis used to study the employee scheduling problem in pharmaceutical industries. Scheduling is done based on the aspects of both management and employees. The management aspect is to reduce the total cost of the employee by minimizing the makes span. The employee aspects are a preferred number of shifts, shift types, days off, etc. In this scheduling an extra care is taken in not to assign a night shift on the day beforethe day off, because once a night shift assigned to the employee almost present in the next morning. A mathematical model is formed based on flexible working of the employees. In this model, 6 hour shifts isallotted to the employees, which reduces the total cost by 10.135% and the makes span by 4.145% compared to the other models.

Keywords: Employee Scheduling, Hybrid Genetic Algorithm.

DOI: 10.47750/pnr.2022.13.S04.122

INTRODUCTION

In the present trend we often use computers and machineries to perform the jobs in a fraction of a second. These computers and machines should be operated by man power [1]. As of now there is no substitution for man intelligence power. So, manpower is an important tool in software companies, hardware companies and industrial areas. Therefore, using their knowledge, intelligence, creativity, personality, character and adapting capacity ofthe environment is very important [2]. Scheduling deals with arrangement, controlling and optimization of work and workloads in a manufacturing system. Scheduling is the main key for all organizations, which prepare a calendar for any activity to functioning the organization. Employee scheduling is one of the important process in the manufacturing system. The days off assignment problem in the employee scheduling was provided by Burns and Carter [3]. They defined a set of lower bounds and in their result; an employee should work for six consecutive days of the week. Tour assignment problem is proposed by [4]. In their paper an employee is availing two consecutive days offin a week. The break assignment is proposed in [5].Later [6] and [7] proposed a total model consisting of shift scheduling, days off assignment and break assignment. Doctor scheduling, nurse scheduling and employee scheduling using genetic algorithm are available respectively in [8], [9] and [10]. But there is very less literature available in shift scheduling using a hybrid genetic algorithm. Therefore, in this paper, we are applying the Hybrid genetic algorithm for Employee scheduling.

A genetic algorithm is a tool for solving optimization problems which are developed based on the principal of natural selection. In 1970 John Holland introduced the genetic algorithm for the first time. And he introduced basic principles of genetic algorithm [11]. Thereafter many more literatures are available in [12], [13] and [14] and few reports of the genetic algorithm are available in [15], [16], [17] and [18]. Genetic algorithm has a wide range of application due its independent of error free surface. Non differentiable, multimodal, non-continuous, and also NP- Complete problems are solved using genetic algorithm [19]. The multimodal problems are solved with the relative ease using genetic algorithm in [20] and [21]. This is also used in solving nonlinear identification problems [22]. It is also used in the secure communication systems [23] and [24].The above literature survey shows that there is no work on Employee scheduling usinga hybrid genetic algorithm. Therefore, in this paper, we are adapting hybrid genetic algorithm(HGA) for employee scheduling.

A GA hybridized with a local search procedure is called a hybrid genetic algorithm (HGA).

A basic HGA procedure has the following steps.

(1) Define an objective/fitness function, and set the GA operators (such as population size, parent/offspring ratio, selection method, number of crossovers, and mutation rate).

(2) Randomly generate the initial population as the current parent population.

(3) Evaluate the objective function for each individual (chromosome or solution) in the initial population.

(4) Generate an offspring population by using GA operators (such as selection/mating, crossover, and mutation).

(5) Evaluate the objective function of each individual in the offspring population.

(6) Perform a local search on each offspring, evaluating the fitness of each new location, and replace the offspring if there exists a locally improved solution.

(7) Decide which individuals to include in the next population. This step is referred to as “replacement" in that individual from the current parent population are “replaced" by a new population consisting of those individuals from the offspring and/or the parent populations.

(8) If a stopping criterion is satisfied, then the procedure is halted. Otherwise, go to Step 4.

Without Step 6, an HGA is just a GA. Therefore, HGAs have all the properties possessed by GAs. Like GAs, HGAs are a large family of algorithms that have the same basic structure but differ from one another with respect to several strategies such as stopping rules, operators which control the search process, and the local search meme.

Based on previous experiences, in this study, we use a continuous HGA where chromosomes are coded as continuous measurement variables. Suppose there are variables; that is, there are genes in each chromosome. We also make the following assumptions. The (parent) population size is and the offspring population size is also . The type of selection we utilize is random pairing. The blending crossover is utilized and the number of crossover points depends on the number of dimensions of a specific objective function. Random uniform mutation is utilized and the mutation rate is set around or equal to. The type of replacement over both parent and offspring populations is either ranking or tournament. For details on the setting of the GA operators; see, for example [25-29].

In this paper, we concentrate on flexible working an employee and also maximum utilization of machines. Flexible working of an employee is done by concentrating on the factors such as preferred number of shifts, requested day off and shift types in the scheduling period. In this paper, we are assigning a 6 hour shift to the Employees. The paper begins with the mathematical formulation, followed by implementation of an Artificial Immune System, computations and experiments and finally ends with result and discussion.

Mathematical Formulation

Every organization has its own fixed activities. In this research paper, we are giving attention to one fixed organization. This organization prepares employee's schedule. There are four 6hrs shift for an employee to work in a day. First is Morning Shift (MS) between 6am to12pm, second is Afternoon Shift (AS) between 12pm to 6pm, third is Evening Shift (ES) between 6pm to 12am and fourth is Night Shift (NS) between 12am to 6am. According to this totally 28 shifts per week are available. In this Paper we are concentrating on both Organization objectives as well as the employee’s objectives. The Organization Objective is to minimize the Deflection or Modification in the required number of employees required for each shift. The Employee objective is to give preferences to the number of shifts, types of shifts, requested days off and consecutive days off working patterns in the employee schedules. As per the organization, regulation an employee should adopt minimum 7 shifts and maximum 8 shifts per week. The preference of choosing shifts is given equally to all the employees. Final scheduling is done under the minimum requirement of employees for each shift. In this scheduling, we are mainly concentrating on giving two consecutive days off for an employee by avoiding on−off−on and off−on−off pattern of working. And also, a night shift is not assigned to an employee who is going to take a day off for the next day. The programming model created from a scheduling pattern as follows

Indices

n	number of employees working in organization
m	number of days in the scheduling period.
i	index of employees, $i = 1,2,\ldots\ldots.,n$
j	index of days in the scheduling period, $j = 1,2,\ldots\ldots,m$
s	shift types, $s = 0,1,2,3,4.(s = 0 \rightarrow$ day off, $s = 1 \rightarrow$ morning shift

s = 2 → afternoon shift, s = 3 → evening shift and s = 4 → night shift)

p scheduling periods, p=1,2,3and 4.(morning shift →6am to 12pm, afternoon shift → 12pm to 6 pm, evening shift →6pm to 12am and night shift → 12am to 6am).

Parameters

$$a_{ps} = \begin{cases} 1 & if\ the\ shift\ type\ s\ contains\ the\ period\ s \\ 0 & otherwise \end{cases}$$

dur_s duration of shift s.

min−shift minimum number of shifts allotted for an employee in the scheduling period

max−shift maximum number of shifts allotted for an employee in the scheduling period

pns_i preferred number of shifts of an employee i.

pne_{jp} preferred number of employees required for the period p on the day j

$\mathrm{lb}_{jp}, \mathrm{ub}_{jp}$ lower bound and upper bound on the demand of employees on the day j in the period p.

do_{ij} days off of an employee i on the day j

max−score maximum score of an employee if he allotted to his preferred shift

Decision Variables

$$x_{ijs} = \begin{cases} 1 & if\ an\ employee\ i\ is\ assigned\ to\ shift\ s\ on\ the\ day\ j \\ 0 & otherwise \end{cases}$$

pde_{jp} positive deflection from the preferred number of employees required for the day j in the period p

ds_i^+, ds_i^- positive and negative deflection from the preferred number of shifts for each employee

rdo_i negative deflection for the requested days off of an employee

$nspddo_i$ negative deflection of assigning night shift in the previous day of the day off

$\mathrm{off{-}p}_{ij}$ the pattern of off-on-off in the scheduling period

$\mathrm{on{-}p}_{ij}$ the pattern of on-off-on in the scheduling period

b_{ijs} break for an employee i on the day j in the shift s

Objective Functions

The objective of an organization management is to minimize the positive deflection from the preferred number of employees in the working period:

$$Minimize \sum_{j=1}^{m} \sum_{p=1}^{4} \mathrm{pde}_{jp}$$

(1)

The objectives related to employees are given as follows.

Minimization of positive and negative deflections of each employee to preferred shift is taken as the first objective:

$$Minimize \sum_{i=1}^{n} (ds_i^{+} + ds_i^{-})$$

(2)

Minimization of negative deflection of the requested days off for an employee is considered as second objective:

$$Minimize \sum_{i=1}^{n} rdo_i$$

(3)

Minimization of off–on–off pattern is considered as third objective:

$$Minimize \sum_{i=1}^{n} \sum_{j=1}^{m} \mathrm{off} - \mathrm{p}_{ij}$$

(4)

Minimization of on–off–on pattern is considered as fourth objective:

$$Minimize \sum_{i=1}^{n} \sum_{j=1}^{m} \mathrm{on} - \mathrm{p}_{ij}$$

(5)

Minimization of negative deflection of assigning night shift in the previous day of the day off is the final objective:

$$Minimize \sum_{i=1}^{n} nspddo_i$$

(6)

Constraints:

An employee should be assigned at least for any one of the shifts available for the day or should be taken day off:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{s=0}^{4} x_{ijs} = 1$$

(7)

The number of shifts assigned should not exceed the maximum number of shifts:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{s=1}^{4} \mathrm{dur}_s \, x_{ijs} \leq \mathrm{max} - \mathrm{shift}$$

(8)

The number of shifts assigned should be greater than or equal to the minimum number of shifts:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{s=1}^{4} \mathrm{dur}_s\, x_{ijs} \geq \ \mathrm{min-shift}$$

(9)

To avoid off–on–off pattern in the scheduled period, we consider the following constraint:

$$\sum_{i=1}^{n}\sum_{j=1}^{m-2} x_{ij0} + \sum_{i=1}^{n}\sum_{j=1}^{m-2}\sum_{s=1}^{4} x_{i(j+1)s} + \sum_{i=1}^{n}\sum_{j=1}^{m-2} x_{i(j+2)0} - \sum_{i=1}^{n}\sum_{j=1}^{m-2} \mathrm{off-p}_{ij} \leq 2$$

(10)

To avoid on–off–on pattern in the scheduled period, we consider the following constraint:

$$\sum_{i=1}^{n}\sum_{j=1}^{m-2} x_{ij0} + \sum_{i=1}^{n}\sum_{j=1}^{m-2}\sum_{s=1}^{4} x_{i(j+1)s} + \sum_{i=1}^{n}\sum_{j=1}^{m-2} x_{i(j+2)0} - \sum_{i=1}^{n}\sum_{j=1}^{m-2} \mathrm{on-p}_{ij} \leq 2$$

(11)

For each day the minimum staff level should be maintained:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{s=1}^{4} a_{ps}\, x_{ijs} \geq \ \mathrm{lb}_{jp}$$

(12)

The employee should not work more than five consecutive days in the scheduled period:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{s=1}^{4} x_{ijs} + x_{i(j+1)s} + x_{i(j+2)s} + x_{i(j+3)s} + x_{i(j+4)s} + x_{i(j+5)s} \leq 5$$

(13)

Care should be taken that the number of employees allotted for each shift should not exceed the preferred number of employees for each period of the working day:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{s=1}^{4} a_{ps}\, x_{ijs} - \ \sum_{j=1}^{m}\sum_{p=1}^{4} \mathrm{pde}_{jp} \leq \sum_{j=1}^{m}\sum_{p=1}^{4} \mathrm{pne}_{jp}$$

(14)

Night shift should not be allotted to an employee i on the previous day of the day off:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{s=1}^{4} (x_{ijs} + x_{i(j+1)s} + x_{i(j+2)s} + x_{i(j+3)s} + x_{i(j+4)s}) - \sum_{s=1}^{3} x_{i(j+5)s} \leq 5$$

(15)

An half an hour break is given for an employee in each shift:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{s=1}^{4} x_{ijs} - \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{s=1}^{4} b_{ijs} \ \leq 1$$

(16)

Since an employee should not work more than five consecutive days and also giving two consecutive days off, we have to allot all the shifts in five days of the working period. As per the one of the organization regulation, an employee should do minimum 7 shifts per week. In that 7 shifts 5 shifts may be allotted for 5 consecutive days, the remaining 2 shifts must be inserted in any two working days in the scheduled period.

Implementation Of Hybrid Genetic Algorithm

Creating An Initial Population

We randomly create the initial population of the employee chromosomes. Then we check whether primary goal is achieved or not trough minimum working requirements.

Crossover

The main intention of crossover is to combine the best properties of both the parents to yield a new better chromosome [34, 35]. The simple crossover operator is consists of randomly selected crossover point and thereafter recombines the pair of chromosome in order to form a new chromosome.

Mutation

The next stage of the crossover is the mutation process which acts on the pair of chromosomes. Mutation is the important force for revolution, although it is infrequent in nature. There are two important mutation process, namely inverse mutation and pairwise mutation.

Inverse Mutation

We generate some random positions between (1, n), where n is the number of employees. Let us select two positions randomly for mutation, let it be 1 and 5. Then inverse mutation is obtained by reversing the order of the sequence between the positions 1 and 5 as given below

Original string → [5 1 6 2 7] 9 4 8 3 10
Mutated string ← [7 2 6 1 5] 9 4 8 3 10

Pair wise Mutation

We generate some random positions between (1, n), where n is the number of employees. Let us select two positions randomly for mutation, let it be 1 and 5. Then pairwise mutation is obtained by interchanging the positions 1 and 5 as given below

Original string → [5] 1 6 2 [7] 9 4 8 3 10
Mutated string ← [7] 1 6 2 [5] 9 4 8 3 10

Evaluation of Fitness value

The fitness value is calculated using two criteria depending on the solution type.

Criteria 1: If the chromosome is under the feasible solution, then the fitness value is the objective function value.

Criteria 2: If the chromosome is under the non-feasible solution, then the fitness value is equal to 1000 iterations.

Chromosome Selection

Chromosomes are selected using Roulette.

Preserving Strategy

The chromosome with high fitness values are replaced by the new chromosomes with lower fitness value and used as initial population for the next generation.

Termination Criteria

Once we met the specified number of iterations, we stop the process immediately.

The general flow chart of a genetic algorithm is given in the figure 1.

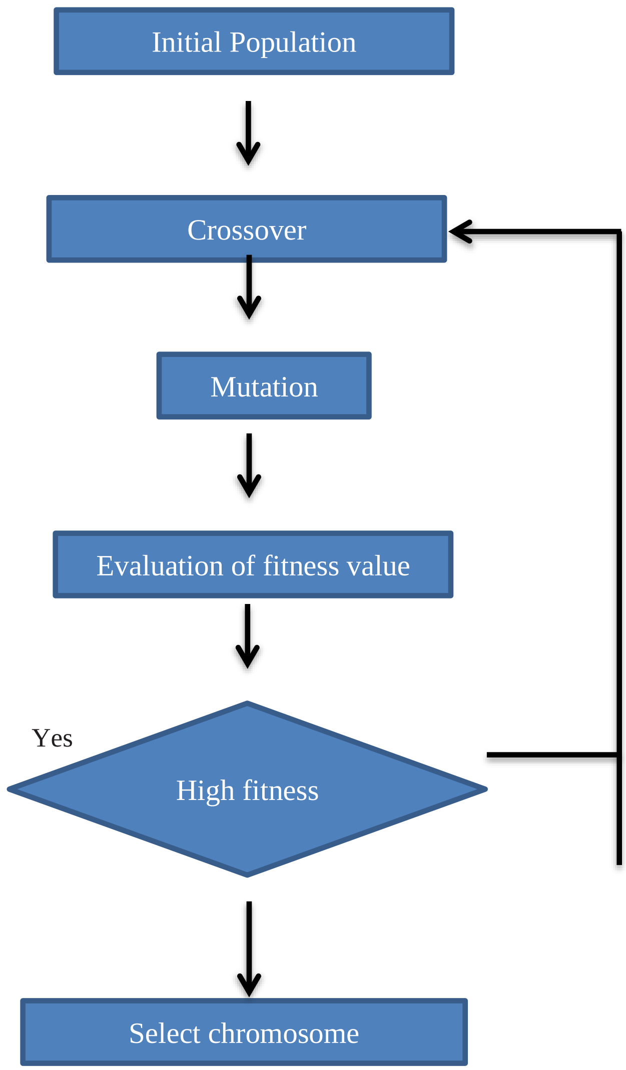

Figure 1: Flow chart of the Genetic Algorithm

Computations And Results

We are making a scheduling pattern for one week of the working period. In this problem we are concerned with a small unit of a manufacturing industry with 30 employees. There are four shifts available on each day of the working period, namely Morning Shift (6am to 12pm), Afternoon Shift (12pm to 6pm), Evening Shift (6pm to 12am) and Night Shift (12am to 6am). As one of the regulation of management each employee should do a minimum of 7 shifts and maximum of 8 shifts per week. The minimum number of employees required (lb_{jp}) for the week days (i.e., Monday to Friday) and weekend days (i.e., Saturday and Sunday) is given as follows. In the week days each 8 employees are needed for morning, afternoon, evening shifts and 7 employees are needed for night shifts. In the Weekend days each 8 employees are needed for morning, afternoon, evening shifts and 8 employees are needed for night shifts. The maximum number of employees (ub_{jp}) is determined as (lb_{jp}+3). The preferred number of employees (pne_{jp}) is determined as (lb_{jp}+1). Since in the scheduling pattern has two consecutive days off, the employee has to choose any one of the Son & Mon, Mon & Tue, Tue & Wed, Wed & Thu, Thu & Fri, Fri & Sat and Sat & Sun pattern. The preferred number of shifts, shift types and days off for each employee are given in Table 1 and the number of employees required for each shift is given in the Table 2.

Table 1: Preferred number of shifts, shift types and day off of the employees.

Employees	Preferences of the employees		
	Number of shifts	Shift types	Day's off
1	7	MS/AS	Son & Mon
2	7	MS/AS	Son & Mon
3	7	MS/AS	Son & Mon
4	7	MS/AS	Son & Mon
5	7	MS/AS	Mon & Tue
6	7	MS/AS	Mon & Tue
7	8	MS/AS	Mon & Tue
8	7	MS/AS	Mon & Tue
9	8	MS/AS	Tue & Wed
10	7	MS/AS	Tue & Wed
11	7	AS/ES	Tue & Wed
12	7	AS/ES	Tue & Wed
13	7	AS/ES	Wed & Thu
14	7	AS/ES	Wed & Thu
15	7	AS/ES	Wed & Thu
16	8	AS/ES	Wed & Thu
17	7	AS/ES	Thu & Fri
18	7	AS/ES	Thu & Fri
19	7	AS/ES	Thu & Fri
20	7	AS/ES	Thu & Fri
21	7	ES/NS	Fri & Sat
22	8	ES/NS	Fri & Sat
23	7	ES/NS	Fri & Sat
24	7	ES/NS	Fri & Sat
25	7	ES/NS	Sat & Sun
26	7	ES/NS	Sat & Sun
27	7	ES/NS	Sat & Sun
28	7	ES/NS	Sat & Sun
29	7	ES/NS	Sat & Sun
30	7	ES/NS	Sat & Sun

Table 2: Number Employees required for each shift

Days	# of Employees required			
	Morning Shift	Afternoon Shift	Evening Shift	Night Shift
Mon	8	8	8	7
Tue	7	8	8	6
Wed	8	8	8	8
Thu	7	7	8	8
Fri	8	8	7	6
Sat	7	7	8	6
Sun	8	8	8	8

The proposed algorithm is coded in the MATLAB R2013a (7.14.0.739), 64-bit (win64). All tests were executed on an Intel core i3 processor under a Microsoft Windows 7 (64-bit) operating system. The resulting employee schedule is given in the

Table 3 and the employee score for the obtained scheduling pattern is given in the Table 4.

Table 3: Obtained employee scheduling pattern

Employees	Working Days						
	Mon	Tue	Wed	Thu	Fri	Sat	Sun
1		MS&AS	MS&AS	MS	NS	ES	
2		MS	MS&AS	MS&AS	NS	ES	
3		MS	MS	MS&AS	MS&AS	ES	
4		MS	MS	AS	MS&AS	MS&AS	
5			MS&AS	NS	AS	MS	MS&AS
6			MS	AS	MS&AS	MS	MS&AS
7			AS	MS	MS&AS	MS&AS	MS
8			MS	AS	MS	MS&AS	MS&AS
9				MS&AS	MS	MS	MS&AS
10	MS&AS			MS	MS	MS&AS	MS
11	AS&ES			NS	NS	AS&ES	MS
12	AS&ES			NS	NS	ES	AS&ES
13	AS&ES	AS&ES			NS	NS	NS
14	MS	AS&ES			AS&ES	NS	ES
15	MS	MS			AS&ES	AS&ES	AS
16	MS&AS	MS			NS	AS&ES	NS
17	AS&ES	AS	AS			NS	AS&ES
18	AS&ES	AS&ES	AS			NS	NS
19	MS	AS&ES	AS&ES			MS	MS
20	MS	MS	AS&ES			ES	AS&ES
21	ES&NS	ES	MS	MS			ES&NS
22	MS	AS	ES&NS	ES			ES&NS
23	AS	AS	ES&NS	ES			ES&NS
24	AS	ES	ES&NS	ES			ES&NS
25	NS	ES&NS	ES	ES&NS	ES		
26	ES&NS	ES&NS	NS	AS	AS		
27	ES&NS	NS	NS	ES&NS	ES		
28	NS	ES&NS	NS	ES&NS	ES		
29	NS	NS	ES&NS	ES&NS	ES		
30	NS	NS	ES&NS	ES&NS	ES		

Table 4: Preference score of the employees

Employees	Preferences scores of the employees			
	Number of shifts	Shift types	Day's off	Not assigning night shift in previous day of the day off
1	10	7.2	10	10
2	10	7.2	10	10
3	10	8.6	10	10
4	10	10	10	10
5	10	8.6	10	10
6	10	10	10	10
7	8.7	10	10	10
8	10	10	10	10
9	8.7	10	10	10

10	8.7	10	10	10
11	10	7.2	10	10
12	10	7.2	10	10
13	10	7.2	10	10
14	10	7.2	10	10
15	10	7.2	10	10
16	7.8	7.2	10	10
17	10	7.2	10	10
18	10	5.8	10	10
19	10	5.8	10	10
20	10	7.2	10	10
21	10	7.2	10	10
22	8.7	7.2	10	10
23	10	7.2	10	10
24	10	8.6	10	10
25	10	10	10	10
26	10	7.2	10	10
27	10	10	10	10
28	10	10	10	10
29	10	10	10	10
30	10	10	10	10

Conclusion

In this scheduling an extra care is taken for not to assign Night Shift (12am to 6am) on the day before the day off because once night shift assigned, the employee almost present in the next morning. Also, we consider the following example to validate our model consisting of 6 hour shifts.

A. Minimizing the Makes Span

Assuming that we got an order, which will take approximately 1000 hours to complete. In the normal 8 hour shifts an employee will be availing 1 hour break (Including two short breaks and one lunch break) and therefore he will be working only for 7 hours per shift. So to complete the order it will take approximately 143 shifts of 8 hours i.e. 143×8=1144 hours. But if we allot 6 hour shift it will be contained only one half an hour break and an employee will be working for 5½ hours per shift. To complete the same order, it will need nearly 182 shifts of 6 hours i.e. 182×6=1092 hours. It will reduce 52 hours of the total completion time compared to 8 hours shift.

B. Minimization of Cost

Suppose the same order should be delivered within a week, then we must assign the proper number of employees required to complete and deliver the order. In 8 hour shift each employee will be working only for 7 hours (remaining one hour is the break). Since each employee will be working only 5 shifts per week, then total working hour of each employee becomes 5×7=35 hours per week. Therefore, we need a minimum of 29 employees to deliver the order in a week. But in the case of 6 hour shift each employee will be working 5½ hours (remaining half an hour is the break). Since each employee will be working minimum 7 shifts per week, then the total working hour of each employee becomes 7×5½ = 38½ per week. Therefore, we need a minimum of 26 employees to deliver the order in a week. This will reduce 3 employee's salary in the total cost comparing with a 8 hour shift. The proposed model reduces the total cost by 10.135% and the makes span by 4.145% compared to the other models (8 hour shifts). Comparison between proposed model of 6 hour shifts and the existing 8 hour shifts is given in the figure 2.

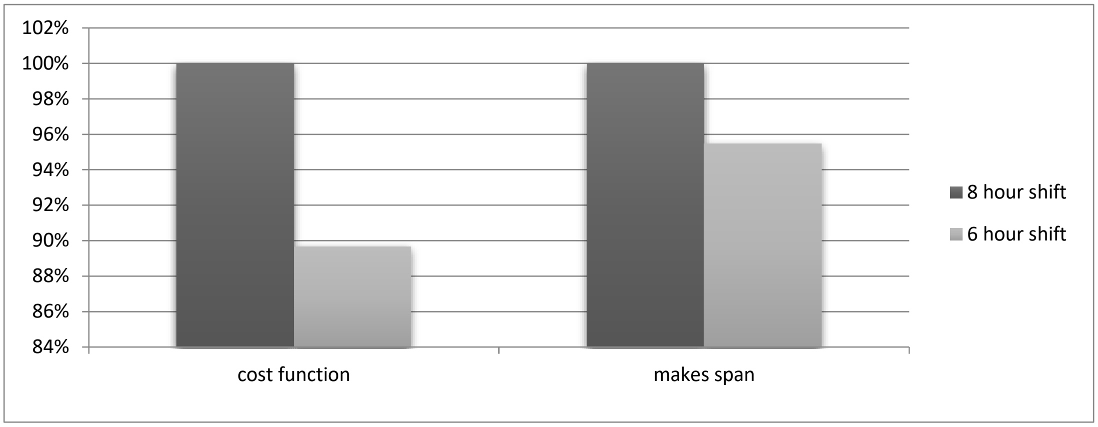

Figure 2: Comparison of proposed 6 hour shift with 8 hour shift

REFERENCES

1. Hoong Chuin Lau. On the complexity of man power shift scheduling, Computers & Operations Research 1996, Volume 23, pages 93-102
2. Wang Qun, Wang Yinluo, Multi-dimension dynamic testing of multi-hierarchy man-power and its expert system of group optimization, Computers & Industrial Engineering, Volume 31, Issues 3-4, December 1996, Pages 707-712
3. Burns RN, Carter MW. Work force size and single shift schedules with variable demands. Management Science 1985; 31(5):599–607.
4. Alfares HK. An efficient two-phase algorithm for cyclic days-off scheduling. Computers & Operations Research 1997; 25:913–23.
5. Bechtold SE, Jacobs LW. Implicit modeling of flexible break assignments in optimal shift scheduling. Manag Sci 1990; 36(11):1339–51
6. Jarrah AIZ, Bard JF, DE Silva AH. Solving large-scale tour scheduling problems. Manag Sci 1994; 40(9):1124–45.
7. Bard JF, Binici C, deSilva AH. Staff scheduling at the United States postal service. Comput Oper Res 2003; 30(1):745–71.
8. Abir Alharbi and Kholood AlQahtani, "A Genetic Algorithm Solution for the Doctor Scheduling Problem", ADVCOMP 2016: The Tenth International Conference on Advanced Engineering Computing and Applications in Sciences
9. Komgrit Leksakul and Phetsawat, "Nurse scheduling using genetic algorithm", Hindawai Publishing Corporation, Mathematical Problems in Engineering, Volume 2014, Article ID 246543.
10. Adrian Brezulianu, Monica Fira, and Lucian Fira, "A genetic algorithm approach for a constrained employee scheduling problem as applied to employees at mall type shops", International Journal of Advanced Science and Technology, Vol. 14, January, 2010.
11. J. H. Holland, Adaption in Natural and Artificial Systems. Cambridge, MA: MIT Press, 1975.
12. L. Davis, Handbook of Genetic Algorithms. New York: Van Nostrand Reinhold, 1991.
13. D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Reading, MA: Addison-Wesley, 1989.
14. Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Program, 2nd Ed. Berlin: Springer-Verlag, 1994.
15. D. Beasley, D. R. Bull, and R. R. Martin, "An overview of genetic algorithms: Part I-Fundamentals," Univ. Comput., vol. 15, no. 2, pp. 58-69, 1993.
16. D. Beasley, D. R. Bull, and R. R. Martin, "An overview of genetic algorithms: Part 2-Research topics," Univ. Comput., vol. 15, no. 4, pp. 170-181, 1993.
17. M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," Computer, pp. 17-26, June 1994.
18. D. Whitley, "A genetic algorithm tutorial," Dept. Comput. Sci, Colorado State Univ., Tech. Rep. CS-93-103, Nov. 1993.
19. M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco, CA: Freeman 1979.
20. R. Nambiar and P. Mars, "Adaptive IIR filtering using natural algorithms," in Workshop on Natural Algorithms in Signal Processing, Chelmsford, Essex, Nov. 1993, pp. 20/1-20/10.
21. P. B. Wilson and M. D. Macleod, "Low implementation cost IIR digital filter design using genetic algorithms," in Workshop on Natural Algorithms in Signal Processing, Chelmsford, Essex, Nov. 1993, pp. 4/1-4/8.
22. R. C. Caponetto, L. Fortuna, G. Manganaro, and M. G. Xibilia, "Chaotic system identification via genetic algorithm," in 1st IEENEEEE Int. Con$ GAIs in Engineering Systems: Innovations and Amlications, Sheffield, 1. U.K., 1995: pp. 178-174.
23. L. 0. Chua, K. Eckert, Lj. Kocarev, and M. Itoh, "Experimental chaos synchronization in Chua's circuit," Int. J. Bifurcation, Chaos, vol. 2, no. 3, pp. 705-708, 1993.
24. Lj. Kocarev, K. S. Halle, K. Eckert, U. Parlitz, and L. 0. Chua, "Experimental demonstration of secure communications via chaos synchronization," Int. J. Bifurcation, Chaos, vol. 2, no. 3, pp, 709-713, 1992.
25. D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, Mass, USA, 1989.
26. M. Hamada, H. F. Martz, C. S. Reese, and A. G. Wilson, "Finding near-optimal Bayesian experimental designs via genetic algorithms," The American Statistician, vol. 55, no. 3, pp. 175–181, 2001.
27. D. G. Mayer, J. A. Belward, and K. Burrage, "Robust parameter settings of evolutionary algorithms for the optimization of agricultural systems models," Agricultural Systems, vol. 69, no. 3, pp. 199–213, 2001.
28. F. Ortiz Jr., J. R. Simpson, J. J. Pignatiello Jr., and A. Heredia-Langner, "A genetic algorithm approach to multiple-response optimization," Journal of Quality Technology, vol. 36, no. 4, pp. 432–450, 2004.
29. R. L. Haupt and S. E. Haupt, Practical Genetic Algorithms, Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, Second edition, 2004, With 1 CD-ROM (Windows).