International Journal of Advanced Science and Technology
Vol. 14, January, 2010

A genetic algorithm approach for a constrained employee scheduling problem as applied to employees at mall type shops

[1]Adrian Brezulianu, [2]Monica Fira, and [3]Lucian Fira

[1]Faculty of Electronics, Telecommunications and Information Technology, Technical University of Iasi
[2]Institute of Computer Science, Romanian Academy, Iasi Branch
[3]Faculty of Electronics, Telecommunications and Information Technology, Technical University of Iasi
[1]abrezu@etc.tuiasi.ro, [2]mfira@etc.tuiasi.ro, [3]lfira@etc.tuiasi.ro

Abstract

In this application of artificial intelligence to a real-world problem, the constrained scheduling of employee resourcing for a mall type shop is solved by means of a genetic algorithm. Chromosomes encode a one-week schedule and a constraint matrix handles all requirements for the population. The genetic operators are purposely designed to preserve all constraints and the objective function assures an imposed coverage, this is for people on both sections of the mall. The results demonstrate that the genetic algorithm approach can provide acceptable solutions to this type of employee scheduling problem with constrains.

Keywords: *Genetic Algorithms, Scheduling Problem.*

1. Introduction

The employee or team schedule problem has wide application. It is often encountered by industrial organizations, shopping centers, call centers, hospitals etc [1-4]. The employee scheduling, or more specific, the shift scheduling is defined as a problem of placing resources (employees) into slots in a pattern, in such way to accomplish given constraints. The patterns are denoting a set of legal shifts defined in the terms of work to be done ([5] citing [6]). Various approaches are popular to solve this problem including linear programming [7], branch-and-bound [8], genetic algorithms [9], evolutionary techniques [10] etc. We have chosen to provide solution with a genetic algorithm and to report our experiences here.

2. Problem formulation

The schedule generation problem for mall type markets consists in creating appropriate timetables in order to meet the following requirements:

- an imposed hourly coverage with employees during different working days;
- an imposed daily coverage with employees during different working days;
- a minimum of one employee scheduled every time;

- two free days per employee per week, grouped in three ways of choice: (1) consecutive in week-end, (2) consecutive during week, or (3) non-consecutive;
- a schedule of availability for each employee;
- a given department or section;
- different shift duration of 4, 6, and 8 hours for the part-time and full-time employees.

Table 1. Required daily coverage with employees for both sections

	Section #1	Section #2
Monday	0.71	1.09
Tuesday	0.79	1.03
Wednesday	0.95	0.94
Thursday	0.85	0.86
Friday	1.19	0.99
Saturday	1.30	0.99
Sunday	1.22	1.11
Total	7.00	7.00

Table 1, shows the requested coverages for employees in different week days for both sections. The required coverages are normalized to 1, to obtain a sum of 7 from the values in the whole week for a given department. The coverages for the two sections have different shapes: the first section has higher values on Friday, Saturday and Sunday, the second section has higher values on Monday, Tuesday, and Sunday. The shape of coverage series for the Section #2 is closer to be uniform than the Section #1 case, having a difference between maximum and minimum values of 0.25, respectively 0.51.

Table 2. Required hourly coverage with employees for the section #1

	8:00-13:00	13:00-18:00	18:00-22:00
Monday	0.25	0.4	0.35
Tuesday	0.25	0.4	0.35
Wednesday	0.25	0.4	0.35
Thursday	0.25	0.4	0.35
Friday	0.25	0.4	0.35
Saturday	0.3	0.3	0.4
Sunday	0.3	0.3	0.4

Table 3. Required hourly coverage with employees for the section #2

	8:00-13:00	13:00-18:00	18:00-22:00
Monday	0.2	0.35	0.45
Tuesday	0.4	0.35	0.25
Wednesday	0.4	0.35	0.25
Thursday	0.4	0.35	0.25
Friday	0.4	0.35	0.25
Saturday	0.2	0.35	0.45
Sunday	0.45	0.4	0.15

Tables 2 and 3 present the required coverages with employees for three hourly intervals (from 8:00 to 13:00, from 13:00 to 18:00 and from 18:00 to 22:00) for the Section #1 and Section #2, respectively. The sum of coverages for one day is normalized to 1.

In case of the first section, the need with employees is low in the morning (the first interval), from Monday to Friday. During week-end, the required coverage tends to be uniform. In case of the second section, number of required employees is very low Monday and Saturday morning and Sunday evening (the third interval).

3. Problem solution

In this schedule generation problem, the Genetic Algorithm (GA) minimizes the error, i.e., the difference between imposed and obtained hourly/daily coverages. A target presence matrix (DM) stores the result of multiplying the imposed daily coverage matrix and the imposed hourly coverage matrix and the total number of available employees. Using this multiplication operation, the relative values from coverage matrix are transformed into absolute values, tacking into account the number of available employees for each section. Then, DM is compared to PM, the presence matrix that is computed directly.

The employees get recorded in a database containing all pertinent information: ID, name, department, availability, options for free days etc. GA generates appropriate schedules for all employees taking into account all constraints and employee data.

3.1. Chromosome encoding

The GA chromosomes are arrays for all employees, and each array contains 13 real values for a given employee:

- the working start hour for each of the 5 working days (hij, i = 1..n, j = 1..5)
- the duration of the working time for each of the 5 working days (dij, i = 1..n, j = 1..5)
- the first free day (fdi1, i = 1..n)

- the second free day (fdi2, i = 1..n)
- the department were the employee is assigned (depi, i = 1..n)

Table 4. Chromosome encoding and allowed ranges for each element

	h11	d11	…	h15	d15	fd11	fd12	dep1	…	hn1	…	depn
min	8	8		8	8	1	6	1		8		1
max	22	8		20	8	5	7	2		22		2

This type of representation can completely encode a weekly schedule for all departments and for all employees. By specifying of the minimum and maximum values for the starting working hour, together with the duration of the working time, the availability to be scheduled for every employee can be taken into account. The full-time and part time shifts are modeled using the corresponding duration values. By using appropriate values for the minimum and maximum of the first and second free day, all three cases for the free day's configurations can be modeled.

3.2. Objective function

First, the objective function computes the number of employees who are present at work for each hourly interval, and this gets saved to the presence matrix, PM. Second, the square of distance between desired presence matrix (DM) and computed presence matrix (PM) is calculated. A penalty relation was added to forfeit the timetables for each case when 0 employees are scheduled. The minimization of the function (F) expressed as in the equation (1) is the goal of the used GA:

$$F = sum((DM - PM).\wedge^{2} + 50 \times (PM == 0)) \tag{1}$$

In equation (1), the penalty is implemented using the logic expression (PM = = 0) which returns a value of 1 for each element of 0 from PM; in the other cases, the returned value is 0. The ".^2" operator stands for the element-by-element squaring operation.

3.3. Constraint setting

The minimum and maximum values are set for each gene. The limit values are grouped in a 2-rows constraint matrix with a number of columns equal to the chromosome's length. In this way, all constraints are specified.

The constraints are then checked to be fulfilled during the generation of the initial population and also after application of the mutation operator. The cross-over operator is preserving the constraints because, in the used encoding, the minimum and maximum values of a gene on a certain position *k* are the same for every chromosome.

3.4. Initial population

An initial population is generated using a default functionality of the "MATLAB Genetic Algorithms and Direct Search" toolbox [11]. The generated population satisfies the minimum and maximum values from the constraint matrix. A value of 2000 was used for the GA population size.

3.5. Mutation operator

A customized algorithm was implemented for the mutation operator. This modifies the values of the genes with a random value that preserves the allowed range; mutated genes were kept between the allowed minimum and maximum.

$$mutation\_fr = 1 - crossover\_fr - \frac{elitism\_individuals}{population\_size} \tag{2}$$

The fraction for mutation was set to approx 0.4 and it is computed like in equation (2), where the used terms are self-explanatory.

3.6. Cross-over operator

The customized crossover operator is a multipoint crossover version, as it is implemented in the GADS toolbox [11]. By itself, this operator preserves the range of the allowed values for genes. The crossover fraction was set at 0.6.

3.7. Elitism

The elitism operator is set to select 1 individual (the best one) for the next generation.

3.8. Other genetic algorithm settings

It was used tournament selection with proportional scaling of the fitness.

4. Results

A database with 29 peoples was used for the experiment. 19 employees are assigned to the section #1 and the remaining 10 employees are assigned for the section #2.

Figure 1 shows the Gantt type charts of an one-week schedule for employees from the first section/department. Only the employees who belong to the first department are scheduled for the section #1.

Figure 2 shows the schedule of employees from the section #2 as Gantt type charts. Similarly, only the employees who belong to the second department are scheduled for the section #2.

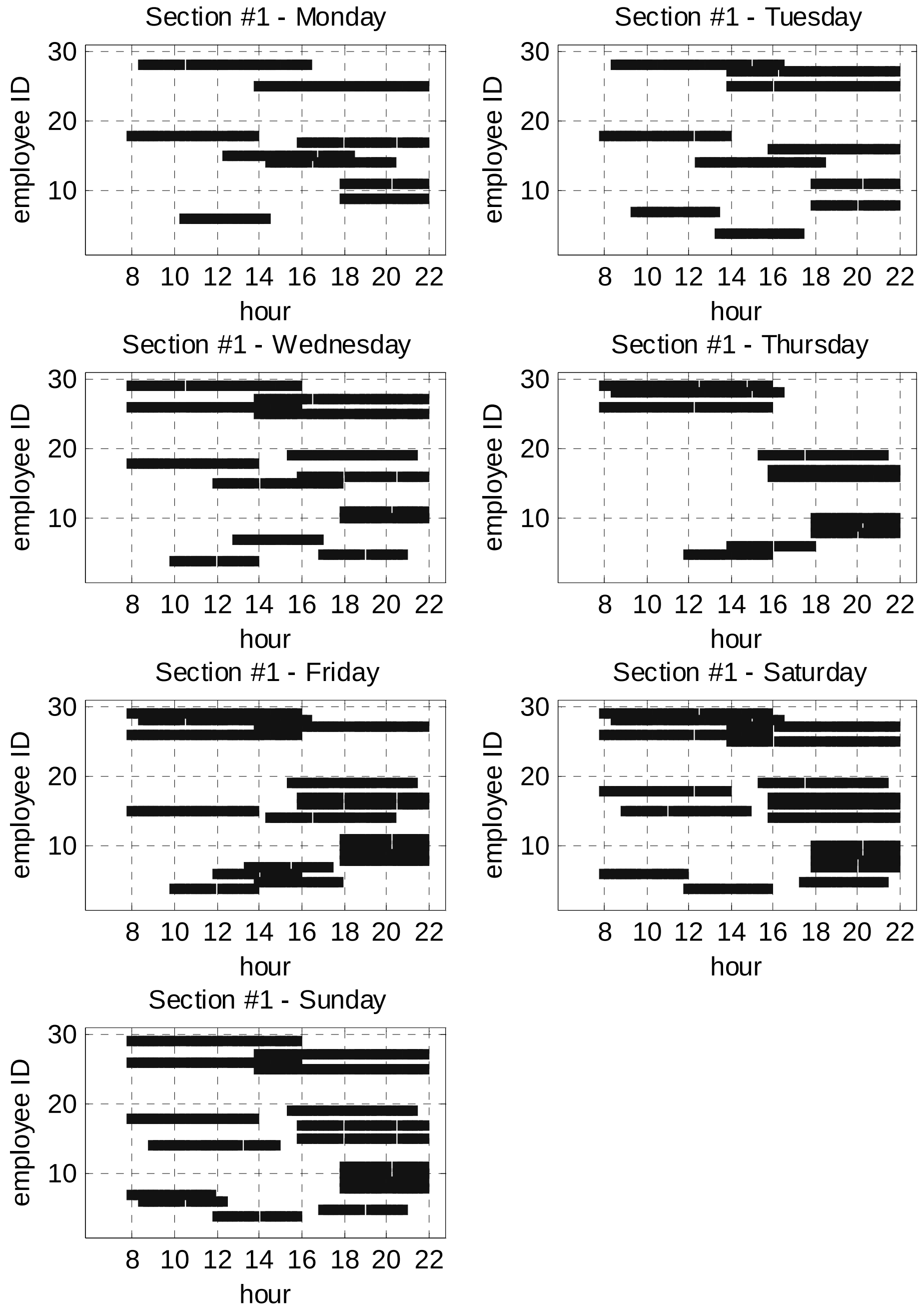

Figure 1. Gantt chart for the employees from the section 1. The black bar drawn on the line <ID> represents the working time for the employee having the corresponding id. The points, from which the bars are consisting, are plotted centered on the corresponding abscissa.

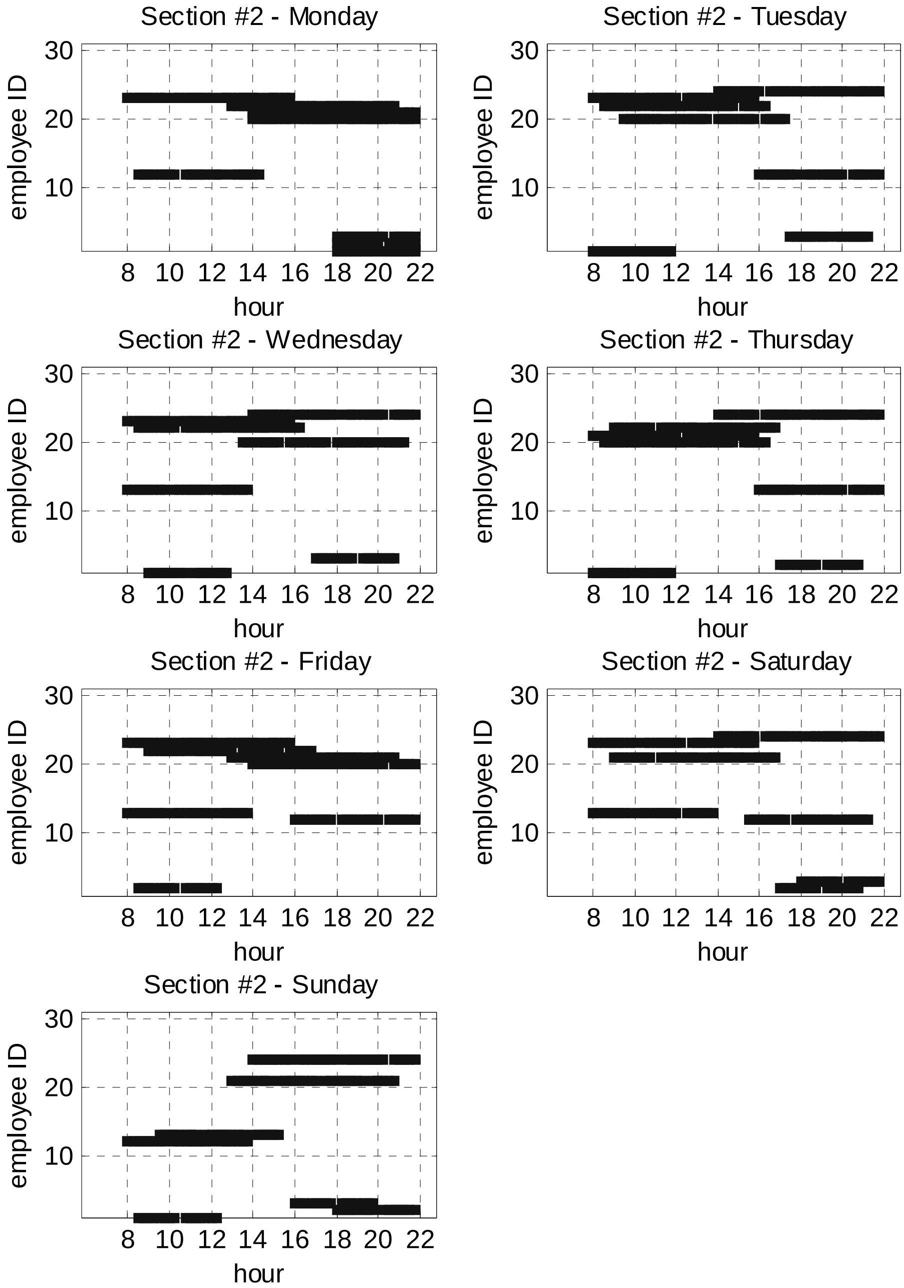

Figure 2. Gantt chart for the employees from the section 2. The black bar drawn on the line <ID> represents the working time for the employee having the corresponding id. The points, from which the bars are consisting, are plotted centered on the corresponding abscissa.

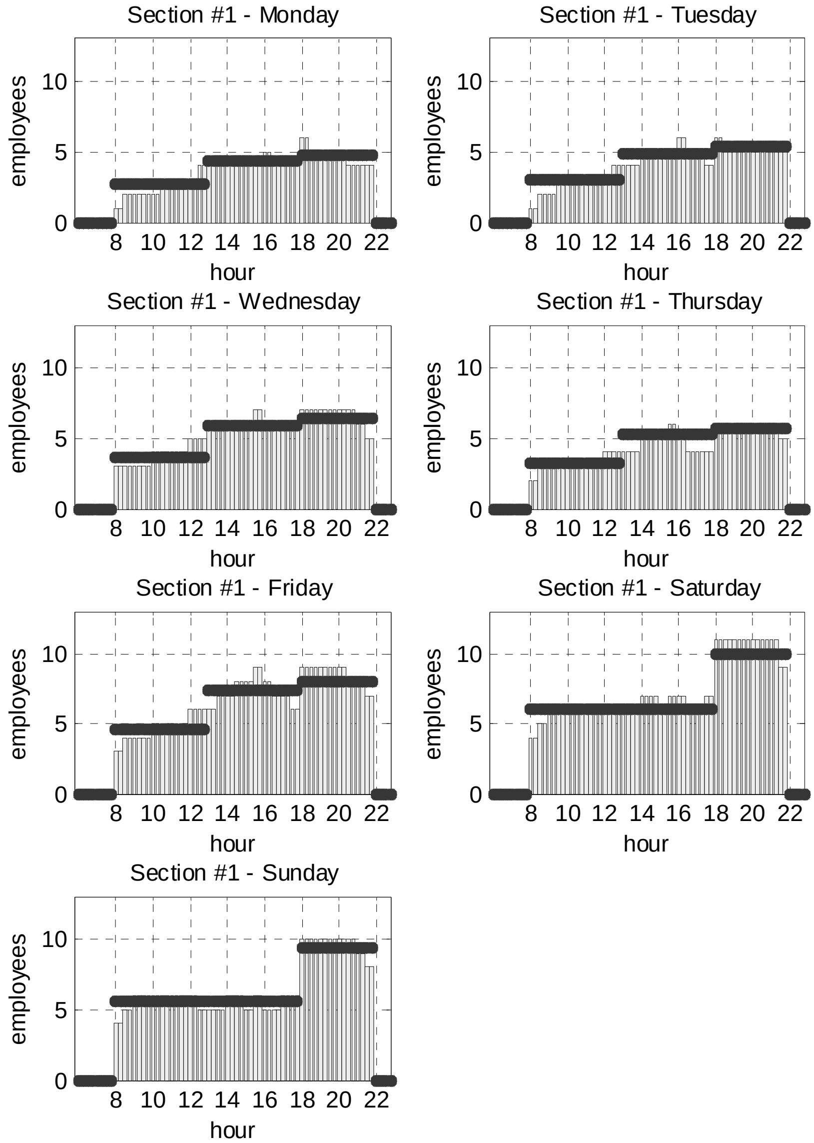

Figure 3. The coverage with employees for the section 1. The light-gray bars represent the counted scheduled employees. The dark-gray bullets are the real-number values for the target coverages.

The height of the light-gray bars from the figures 3 and 4 are obtained by counting, for each time division, the number of the scheduled employees, in fact the number of black bars found for each abscissa in figures 1 and 2, respectively.

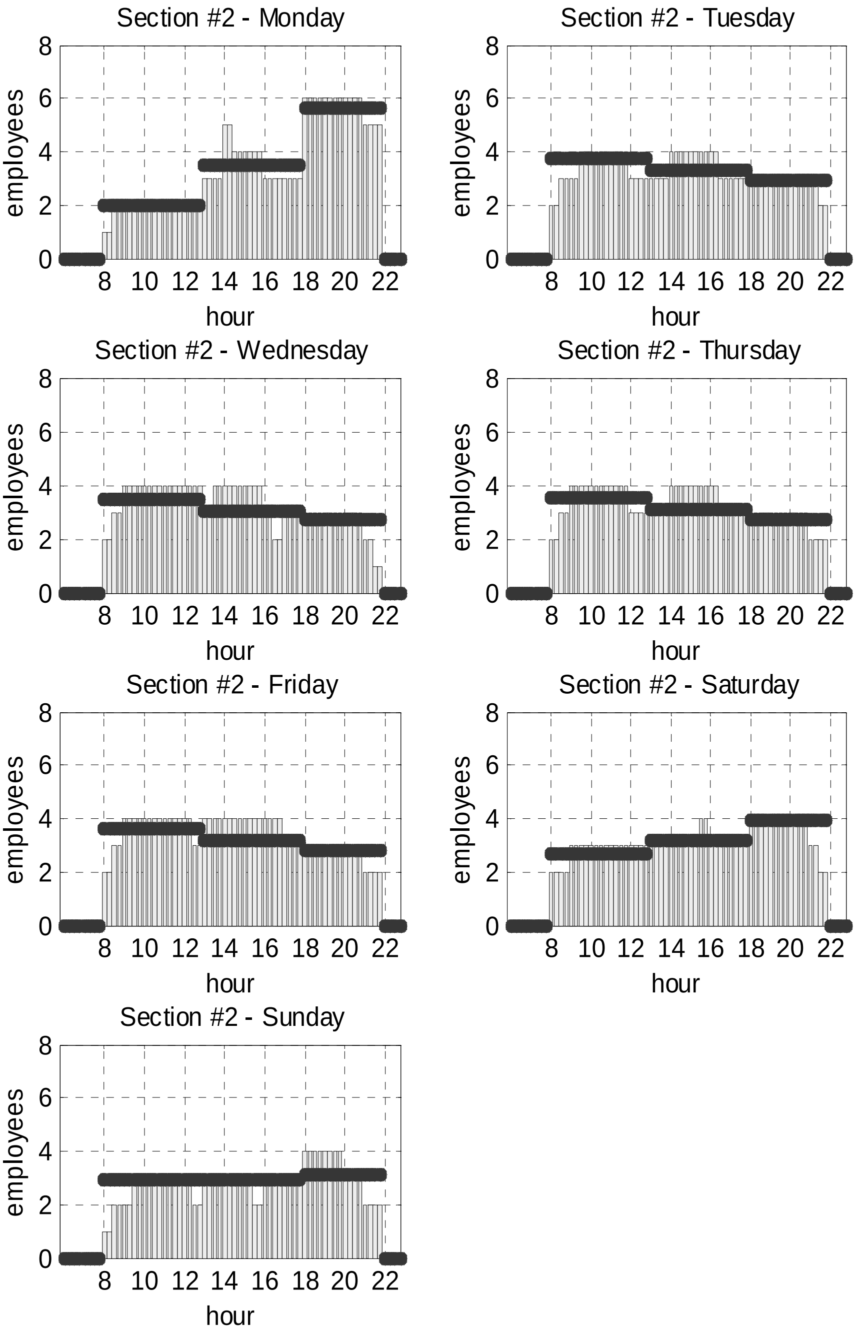

Figure 4. The coverage with employees for the section 2. The light-gray bars represent the counted scheduled employees. The dark-gray bullets are the real-number values for the target coverages.

In figure 3 and figure 4, for the sections #1 and #2, the requested coverages with employees are shown with thick line and the obtained coverages are shown as bars. The coverages are computed from 30 to 30 minutes; but on graphic, for every quarter of hour, a point and a bar, respectively, are placed.

Table 5. Requested and scheduled employees for the section #1, Monday and Sunday

	Monday			Sunday		
	R	S	D	R	S	D
8.00 - 8.30	2,73	1	1	5,63	4	1
8.30 - 9.00	2,73	2	0	5,63	5	0
9.00 - 9.30	2,73	2	0	5,63	6	0
9.30 - 10.00	2,73	2	0	5,63	6	0
10.00 - 10.30	2,73	2	0	5,63	6	0
10.30 - 11.00	2,73	3	0	5,63	6	0
11.00 - 11.30	2,73	3	0	5,63	6	0
11.30 - 12.00	2,73	3	0	5,63	6	0
12.00 - 12.30	2,73	3	0	5,63	6	0
12.30 - 13.00	2,73	4	1	5,63	5	0
13.00 - 13.30	4,37	4	0	5,63	5	0
13.30 - 14.00	4,37	4	0	5,63	5	0
14.00 - 14.30	4,37	4	0	5,63	6	0
14.30 - 15.00	4,37	4	0	5,63	6	0
15.00 - 15.30	4,37	4	0	5,63	5	0
15.30 - 16.00	4,37	4	0	5,63	6	0
16.00 - 16.30	4,37	5	0	5,63	5	0
16.30 - 17.00	4,37	4	0	5,63	5	0
17.00 - 17.30	4,37	4	0	5,63	6	0
17.30 - 18.00	4,37	4	0	5,63	6	0
18.00 - 18.30	4,78	6	1	9,38	10	0
18.30 - 19.00	4,78	5	0	9,38	10	0
19.00 - 19.30	4,78	5	0	9,38	10	0
19.30 - 20.00	4,78	5	0	9,38	10	0
20.00 - 20.30	4,78	5	0	9,38	10	0
20.30 - 21.00	4,78	4	0	9,38	10	0
21.00 - 21.30	4,78	4	0	9,38	9	0
21.30 - 22.00	4,78	4	0	9,38	8	1
Total	109,2		3	187,6		2
Relative error			**2,75**			**1,07**

Table 5 presents an example of requested (R column) and scheduled (S column) employees computed for the section #1, on Monday and Sunday. The absolute value of the integer part of the difference between the number of requested and scheduled employees (see equation 3) can be found in D column.

$$D = abs(int(R - S)) \tag{3}$$

The relative scheduling error is computed using the equation 4,

$$rel\_err = 100 \times \frac{\sum D}{\sum R} \tag{4}$$

were D and R are the difference and requested columns from Table 5.

Table 6. Relative daily scheduling error

	Daily error for Section #1	Daily error for Section #2
Monday	2,75	1,00
Tuesday	4,11	1,07
Wednesday	2,73	3,47
Thursday	4,58	1,14
Friday	3,81	1,11
Saturday	2,00	1,11
Sunday	1,07	3,62
Average of error	3,01	1,79

In Table 6, the relative scheduling errors computed for each day in both sections are shown. The average error for the first section is 3.01% and 1.79% for the second section, leading to an average of 2.40%. The maximum error is 4.58%; it was obtained Thursday, in the section #1. The minim error was obtained Monday, on the section #2; the error value was 1.00%.

5. Conclusion

The accuracy obtained for coverage with employees for the three intervals in a day, for all week days, and for both departments, are all inline with the customer's level of satisfaction and acceptance.

In conclusion, the results appear to show that this employee scheduling problem with constraints has an acceptable solution via the genetic algorithm approach.

As further work, the employees' skills to work on both sections will be taken into account. A new optimization task is to allow that employees from one section to be scheduled for the other section too. We expect to get a better precision in obtaining the target coverages.

References

[1] F. Gloyer and C. McMillan, "The General Employee Scheduling Problem: an Integration of MS and AI", Comput. & Ops. Res. Vol. 13, No.5, 1986, pp. 563-573.

[2] I. Blochliger, "Modeling Staff Scheduling Problems", European Journal of Operational Research, 158 (2004), pp. 533-542.

[3] A. Caprara, M. Monaci, and P. Toth, "Models and algorithms for a staff scheduling problem", Math. Program., Ser. B (2003) 98: 445–476.

[4] S. H. Kim, "Analysis of Real World Personnel Scheduling Problem: An Experimental Study of Lp-Based Algorithm for Multi-Time Multi-Period Scheduling Problem for Hourly Employees", International Journal of Soft Computing, 1 (2): 143-148, 2006.

[5] S. Petrovic, "Towards the Benchmarks for Scheduling Problems", The International Conference on Automated Planning and Scheduling, ICAPS07, Providence, Rhode Island, USA, September 22 - 26, 2007.

[6] A. Wren, "Scheduling, timetabling and rostering - a special relationship?", In Burke E. and Ross P. (eds), Lecture Notes in Computer Science Vol. 1153, 46–75: Springer, 1996.

[7] L. Trilling, A. Guinet, D. Le Magny, "Nurse scheduling using integer linear programming and constraint programming", A Proceedings volume from the 12th IFAC International Symposium, Saint-Etienne, France, 17-19 May 2006, Elsevier, Elsevier , Volume 3, Saint-Etienne, France, p.651-656.

[8] C. Artigues, M. Gendreau, L. Rousseau, and A. Vergnaud, "Solving an integrated employee timetabling and job-shop scheduling problem via hybrid branch-and-bound," *Computers & Operations Research*, vol. 36, no. 8, pp. 2330-2340, August 2009. [Online]. Available: http://dx.doi.org/10.1016/j.cor.2008.08.013

[9] X. Cai and K. N. Li, "A genetic algorithm for scheduling staff of mixed skills under multi-criteria," *European Journal of Operational Research*, vol. 125, no. 2, pp. 359-369, September 2000. [Online]. Available: http://dx.doi.org/10.1016/S0377-2217(99)00391-4

[10] C. Cheong, K. Tan, and B. Veeravalli, "A multi-objective evolutionary algorithm for examination timetabling," *Journal of Scheduling*. [Online]. Available: http://dx.doi.org/10.1007/s10951-008-0085-5

[11] Mathworks, Genetic Algorithms and Direct Search toolbox, http://www.mathworks.com/products/gads/technicalliterature.html (Accessed: April, 2009)

Authors

Adrian Brezulianu was born in Iasi, Romania in 1967. He received the Diploma Engineer and PhD degree in electronic engineering from the "Gh. Asachi" Technical University Iasi, Romania in 1992 and 1999, respectively.

Since 1992 he was successively Assistant, Lecturer and Associate Professor with the Faculty of Electronics, Telecommunications and Information Technology, Technical University of Iasi. From 2008 he is also vice-dean of the Faculty of Electronics, Telecommunications and Information Technology, TU Iasi.

His main research interests include biomedical applications, neural networks, fuzzy systems and genetic algorithm.

Catalina Monica Fira received the B.S. and M.S. degrees in Biomedical Engineering from the “Gr. T. Popa” University of Medicine and Pharmacy Iasi, Romania in 2001 and 2002, respectively. In 2006, she received the Ph.D. degree in Electronics Engineering from "Gh. Asachi" Technical University Iasi, Romania. She is now with the Institute for Theoretical Informatics of the Romanian Academy, Iasi Branch. Her research interests include electrical heart activity analysis, biomedical signal processing and neural networks.

Lucian Iulian Fira received the B.S. degree in Biomedical Engineering from the “Gr. T. Popa” University of Medicine and Pharmacy Iasi, Romania in 2001 and the B.S. degree in Computers and Systems Science from the "Gh. Asachi" Technical University of Iasi, Romania in 2004. In 2002, he received the M.S. degree in Biomedical Engineering from the "Gh. Asachi" Technical University of Iasi, Romania.

He is now Ph.D. student in Electronics Engineering from "Gh. Asachi" Technical University Iasi, Romania.

His research interests include genetic algorithms, neural networks, time series analysis and prediction.