Application of Linear Programming in Scheduling Problem

Md. Mohedul Hasan* and Md. Rajib Arefin

Department of Mathematics, Dhaka University, Dhaka-1000, Bangladesh

Abstract

Scheduling problem is a great concern for several institutions. Larger organizations have to maintain proper scheduling of their employees to ensure good service. This type of problems can be solved using linear programming (LP) as a useful tool. In this paper, we study the application LP to the scheduling problem. We mainly present four different scheduling problems and formulate them using LP. Finally, we solve them using MATHEMATICA v9 software.

I. Introduction

In mathematics, linear programming (LP) problems are optimization problems in which the objective function to characterize optimality of a problem and the constraints to express specific conditions for that problem are all linear. The LP method was first developed by Leonid Kantorovich (1912-1986) in 1939. Leonid Kantorovich developed the earliest linear programming problems in 1939 for use during World War II to plan expenditures and returns in order to reduce costs to the army and increase losses to the enemy. The method was kept secret until 1947 when George B. Dantzig (1914–2005) published the simplex method and John von Neumann (1903–1957) developed the theory of duality as a linear optimization solution, and applied it in the field of game theory. LP is a powerful technique for dealing with the problem of allocating limited resources among competing activities, as well as other problems having a similar mathematical formulation. It has become an important field of optimization in the areas of science and engineering and has become a standard tool of great importance for numerous business and industrial organizations. The goal is to find an optimal solution, that is, the most favorable values of the objective function among feasible ones for which all the constraints are satisfied [2]. A schedule of tasks (or schedule) is the assignment of tasks to specific time intervals of resources, such that no two tasks are on any resource at the same time, or such that the capacity of the resource is not exceeded by the tasks. The goal of this problem is to find an optimal placement of tasks with respect to the costs of the resources assigned. The cost function is often minimized[8].

Employee scheduling problems (SP) arise in a variety of service delivery settings, including the scheduling of nurses in hospitals, police officers, restaurant, vehicles, airline, telephone operators, and others. A typical problem of this form requires the scheduler to give appropriate days off to each of a number of employees who work standard shifts with differing start times while assuring that the required number of employees are on duty throughout the day and week.

Researchers have been showing significant attention to various scheduling problems. Some of these works can be founded in [1, 3, 4, 5,7, 11].

Ernst et al.[7] presented a review of staff scheduling and rostering, which has become progressively important as business becomes more service oriented and cost conscious in a global environment. Optimized staff schedules can provide enormous benefits but require carefully implemented decision support systems if an organization is to meet customer demand in a cost effective manner while satisfying requirements such as flexible workplace agreements, shift equity, staff preferences, and part-time work. *Aickelin et al.*[1] presented a genetic algorithms approach to a manpower scheduling problem arising at a major UK hospital was proposed by using an indirect coding based approach on permutations of the nurses. A heuristic decoder built schedules from those permutations. For the enhancement of the results, they used a hybrid crossover operator and simple bounds to reduce the size of the solution space. *Fred et al.*[8] mentioned that the employee scheduling problem was previously handled by the approach of integer programming formulations, which was time consuming as well as costly. They proposed a new approach based on the combination of management science and artificial intelligence. By adopting this approach, they solved the same problem with 98-99% optimality within very short time by using microcomputer. *Teresa et al.*[11] made an exhaustive experimental evaluation of Genetic Algorithms specially designed to solve bus driver scheduling problems. The Genetic Algorithm they used, was a new coding scheme for the relaxed partitioning problem and considered a complex objective function that incorporated the most relevant features of a quality solution. The results proved that Genetic Algorithms was able to produce very satisfactory solutions, bringing automatic solutions closer to the planners' expectations. *Applegate et al.*[1] designed a new heuristic procedure for finding schedules using the available algorithms for solving the job-shop scheduling problem. They used a cutting plane method for obtaining lower bounds and a combinatorial branch with bound algorithm. That algorithm was able to solve the well-known $10{\times}10$ problem of J. F. Muth and G. L. Thomson within short time.

In this paper, we mainly present SP with a simple LP formulation. This formulation can easily be solved using the well-known Big-M simplex algorithm. Many of the practical problems can be modeled using this formulation. We present four real life oriented models. All of these problems are collected from the internet. In the first model, we represent that how the police department can minimize the total number of officers needed to meet all shift requirements.

---

* Author for correspondence. e-mail: mohedul.math@gmail.com, arefin.math@du.ac.bd

The second model represents that how many servers should report for work at the start of each time period in order to minimize the staff required for one day's operation. In the third model, we represent a problem to find the minimum number of bus drivers to schedule. In the last model, we represent a nurse scheduling problem. All of these scheduling problems have been modeled using LP and solved using MATHEMATICA.

II. Linear Programming Problem

Linear programming is a mathematical method for determining a way to achieve the best outcome such as maximum profit or lowest cost in a given mathematical model for some list of requirements represented as linear relationships. More formally it is a technique for optimization of a linear objective function subject to linear equality or linear inequality constraints. Its feasible region in a convex polyhedron, which is a set defined as the intersection of finitely many half spaces each of which is defined by a linear inequality. A linear programming algorithm finds a point in the polyhedron where this function has the largest or smallest value if such a point exits [9].

General Mathematical Form of Linear Programming Problem (LPP)

The mathematical expression of a general linear programming problem is as follows:

Maximize (or Minimize)

$$z = \sum_{j=1}^{n} c_j x_j \qquad (1)$$

Subject to

$$\sum_{j=1}^{n} a_{ij} x_j \, \{\leq, =, \geq\} b_j \, ;$$
$$i = 1,2, \ldots \ldots . m \qquad (2)$$

Where one and one of the signs ($\leq$, $=$, $\geq$) holds for each constraints in (1) and the sign may vary from one constraint to another. Here c_j ($j = 1,2, \ldots \ldots \ldots, n$) are called profit (or cost) coefficients, $x_j(j = 1,2, \ldots \ldots \ldots, n)$ are called the decision variables. The set of feasible solution to LP is $S = \{(x_1, x_2, . ., x_n)^T : (x_1, x_2, . ., x_n)^T \epsilon R^n\}$ and equation (1) holds at $(x_1, x_2, . ., x_n)^T$. The set S is called the constraints set, feasible region of LP. In vector notation the above problem can be written as:

Maximize (or Minimize)

$$z = c^T x$$

Subject to

$$Ax \, (\leq, =, \geq) \, b$$

where A is an $m \times n$ matrix, x is an $(n \times 1)$ column vector and c is a $(1 \times n)$ row vector.

III. The Scheduling Problem

This section presents an admittedly simplistic scheduling problem that assigns employees to shifts. Typical examples would be bus drivers, hospital nurses, or other shift workers. It must be noted that the formulation presented below does not distinguish between individuals, it only states that, for instance, five drivers are assigned to a specific shift, but it does not indicate who these drivers are and if their allocation would cause a violation of rules set out in their collective agreement (e.g., working shifts too close together, working to many hours, or similar regulations). It also does not take into consideration the possibility of different abilities of the employees. This problem can be seen as the first stage in an employee scheduling systems. The second phase would then be the allocation of individual employees to shifts. This can, of course, also be done in a single stage as shown below.

To formalize, assume that the relevant time slots have been numbered $j = 1, 2 \ldots n,$ and that the smallest number of employees that have to be present during time slot j is denoted by b_j. For notational convenience, we define $b_{m+1} = b_1$. We can then define variables x_j that denote the number of employees that start working in time slot j. Again, for convenience we define $x_{n+1} = x_1$. It now depends how the time slots are defined in relation to the length of the shifts that employees have to work. For instance, if the time slots are defined as four hours (as the need for employees is constant during any 4-hour period) and each shift is 8 hours, then we note that an employee who starts work at the beginning of time slot j will then be available during time slots j and $j + 1$. If the time slots are defined on a two-hour basis, then an employee who starts to work in time slot j will work during time slots $j, j + 1, j + 2,$ and $j + 3$. Assuming that time slots are four hours and shifts last eight hours and given the objective to minimize the number of employees that will be needed, we can formulate the problem as

P: Min $z = \sum_j x_j$

Subject

$$\left. \begin{array}{c} x_j \, + \, x_{j+1} \geq \, b_{j+1}, \, \forall j \\ x_j \geq 0, \, \forall j \end{array} \right\} \qquad (3)$$

Problems of this structure can be solved using Big-M simplex method.

Notice the special structure of the matrix of coefficients. It is a so-called Hoffman-Kruskal matrix [6].

$A = (a_{ij})$ with

$$a_{ij} = \begin{cases} 1 & \text{if } \, i = j \\ 1 & \text{if } i = j + 1 \\ 1 & \text{if } i = 1 \text{ and } j = n \\ 0 & \text{Otherwise} \end{cases}$$

It is easy to demonstrate (e.g., by way of evaluating the determinant with respect to the first row) that **A** has a determinant of two, if n is odd and a determinant of zero, if

n is even. With four-hour time slots, a day has six timeslots, so that n is even if we plan for one or multiple full days. This ensures the integer solutions of the problem provided that all b_j are integers[6].

The more complex scheduling problem which we address has wide applicability, especially in the supermarket, reservation office and fast food fields. It differs rather dramatically from the days-off and the shift scheduling problems by including important real world features that resist practical solution by methods of formal analysis. We first describe the problem informally and indicate the features that a practical solution system must have in order to deal with the problem effectively. We report the result of applying or approach to problems from real world settings and discuss the implications of our empirical results.

Informally, a schedule may be viewed as an assignment of tasks to resources. Scheduling is considered to be the activity of creating such a schedule. In the next section, we develop a model to find minimum number of police officers needed to meet all shift requirements and we solved this model using MATHEMATICA software.

IV. The Police Scheduling Problem

The clark county sheriff's department schedules police officers for 8-hour shifts. The beginning times for the shifts are 8:00am, noon, 4:00pm, 8:00pm, midnight and 4am. An officer beginning a shift at of these times works for the next 8 hours. During normal and weekday operations, the number of officers needed varies depending on the time of the day. The department staffing guidelines require the following minimum number of officers on duty.

Time of day	Minimum officers on duty
8.00am-Noon	5
Noon-4.00pm	6
4.00pm-8.00pm	10
8.00pm-Midnight	7
Midnight-4.00am	4
4.00am-8.00am	6

The department wants to minimize the total number of officers needed to meet all shift requirements.

Solution

We develop a linear program for this problem that will determine the number of officers that should be scheduled to begin the 8-hour shifts at each of the six times (8:00am, noon, 4:00pm, 8:00pm, midnight and 4am) in order to minimize the total number of officers required. Let x_1 be the number of officers beginning work at 8am, and x_2 be the number of officers beginning work at noon and so on.

x_1 = the number of officers that begin work at 8am & work shifts (8am - Noon) & (Noon to 4pm)

x_2 = the number of officers that begin work at Noon & work shifts (Noon – 4pm) & (4pm – 8pm)

x_3 = the number of officers that begin work at 4pm & work shifts (4pm – 8pm) & (8pm to Midnight)

x_4 = the number of officers that begin work at 8pm & work shifts (8pm - Midnight) & (Midnight- 4am)

x_5 = the number of officers that begin work at Midnight & work shifts (Midnight – 4am) & (4am to 8am)

x_6 = the number of officers that begin work at 4am & work shifts (4am – 8am) & (8am - Noon)

For an eight hour shift ((4am-8am) + (8am-Noon)) we need to at least 6 officers from (4am-8am) and 5 officers from (8am-Noon). Now using (3) we have $b_{j+1} = 5$. Therefore first constraint can be written as

$$x_6 + x_1 \geq 5$$

Similarly we can formulate other constraints for the remaining 4 hours schedules.

$$x_1 + x_2 \geq 6$$
$$x_2 + x_3 \geq 10$$
$$x_3 + x_4 \geq 7$$
$$x_4 + x_5 \geq 4$$
$$x_5 + x_6 \geq 6$$

Non-negativity constraints

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0, x_6 \geq 0.$$

Objective function:

Minimize the total number of officers needed to meet all shift requirements.

Min $z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6$

Now using MATHEMATICA software, the optimal basic feasible solution is $x_1 = 3, x_2 = 3, x_3 = 7, x_4 = 0, x_5 = 4$ and $x_6 = 2$ with the minimum value of z is 19.

Therefore the minimum number of officers is 19 to meet all shift requirements.

In the next section, we develop a model to determine the minimum number of servers who should report for work at the start of each time period for one day's operation.

V. The Restaurant Employee Scheduling Problem

The Boston's famous Limoges Restaurant is open 24 hours a day. Server report for duty at 3am, 7am, 11am, 3pm., 7pm., 11pm. and works an 8-hour shift. The following table shows the minimum number of workers needed during the 6 periods into which the day is divided.

Period	Time	Number of servers required
1	3am – 7am	3
2	7am – 11am	12
3	11am – 3pm	16
4	3pm – 7pm	9
5	7pm – 11pm	11
6	11pm – 3am	4

Owner Michelle Limoges, scheduling problem is to determine how many servers should report for work at the start of each time period in order to minimize the staff required for one day's operation.

Solution

We develop a linear program for this problem that will determine the number of servers that should be scheduled to begin the 8-hour shifts at each of the six times (3am, 7am, 11pm, 3pm, 7pm and 11pm) in order to minimize the total number of servers required.

Let x_1 = the number of servers that begin workat 3am and work shifts (3am – 7am) and (7am – 11am)

x_2 = the number of servers that begin work at 7am and work shifts (7am – 11am) and (11am- 3pm)

x_3 = the number of servers that begin work at 11am and work shifts (11am – 3pm) and (3pm – 7pm)

x_4 = the number of servers that begin work at 3pm and work shifts (3pm - 7pm) and (7pm – 11pm)

x_5 = the number of servers that begin work at 7pm and work shifts (7pm – 11pm) and (11pm – 3am)

x_6 = the number of servers that begin work at 11pm and work shifts (11pm– 3am) and (3am – 7am)

For an eight hour shift ((11pm-3am) + (3am-7am)) we need to at least 4 servers from (11pm-3am) and 3 servers from (3am-7am). Now using (3) we have $b_{j+1} = 3$. Therefore first constraint can be written as

$$x_6 + x_1 \geq 3$$

Similarly we can formulate other constraints for the remaining 4 hours schedules.

$$x_1 + x_2 \geq 12$$
$$x_2 + x_3 \geq 16$$
$$x_3 + x_4 \geq 9$$
$$x_4 + x_5 \geq 11$$
$$x_5 + x_6 \geq 4$$

Non-negative constraints:

$$x_i \geq 0, \qquad i = 1, \ldots, 6$$

Define objective function:

Minimize the total number of servers needed to meet all shifts requirements,

Min $z = x_1 + x_2 + x_3 + x_4 + x_5 + x_6$

We solve this model by using MATHEMATICA software input and output are given below:

Input:

Minimize $[x_1 + x_2 + x_3 + x_4 + x_5 + x_6, x_1 +$
$x_2 \geq 12$ && $x_2 + x_3 \geq 16$ && $x_3 + x_4 \geq$
9 && $x_4 + x_5 \geq 11$ && $x_5 + x_6 \geq 4$ && $x_6 +$
$x_1 \geq 3$ && $x_1 \geq 0$ && $x_2 \geq 0$ && $x_3 \geq$
0 && $x_4 \geq 0$ && $x_5 \geq 0$ && $x_6 \geq$
$0, \{ x_1, x_2, x_3, x_4, x_5, x_6 \}]$

Output:

$\{30, \quad \{x_1 \to 3, \ x_2 \to 14, \ x_3 \to 2, \ x_4 \to 7, \ x_5 \to 4, \ x_6 \to 0\}\}$

So the minimum number of servers is 30, can report for work at the start of each time period for one day's operation.

In the next section, we develop a model to determine how many drivers to schedule at each starting time to cover the requirements for buses[10].

VI. The Bus Driver Scheduling Problem

Consider a bus company is scheduling drivers for its buses. The requirement for buses varies from hour to hour because of customer demand as shown in the figure. Time 0 on the figure represents midnight, and times are shown with a 24 hour clock starting at midnight. The problem is to determine how many drivers to schedule at each starting time to cover the requirements for buses. Drivers work eight hour shifts. Note that although a driver can be hired for an eight hour period, there is no requirement that he drive a bus for the entire period. He might be idle for a four hour interval within the period.
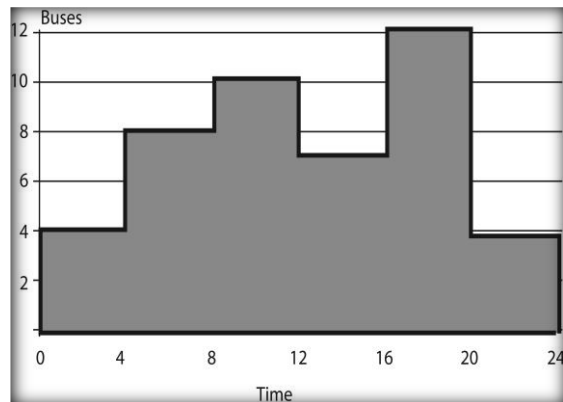
Solution

We need to develop a linear program for this model that will determine the minimum number of drivers. From figure, we see that four buses must run from midnight to 4am, eight buses must run from 4am until 8am, ten buses must run from 8am to 12pm, seven buses must run from 12pm to 16pm, twelve buses must run from 16pm to 20pm, four buses must run from 20pm to 24am, We assume that the bus requirements are the same every day.

Drivers work 8 hour shifts i.e. start at times 0, 4, 8, 12, 16 or 20. For example, a driver starting at time 0 can drive a bus

from time 0 to 8. A driver scheduled to start at time 20 works for the final four hours of the day and the first four hours of the next day. The goal is to minimize the number of drivers used. One feasible solution to this problem is to schedule 8 drivers at time 0, 10 drivers at time 8, and 12 drivers at time 16. This solution will cover all the requirements and use a total of 30 drivers. The problem is to find the smallest number of drivers.

Define Variables:

$x(t)$: Number of drivers scheduled at time t, $t = 0, 4, 8, 12, 16, 20$

We assume that this problem continues over an indefinite number of days with the same bus requirements and that $x(t)$ is the number used in every day at time t. We need constraints that assure that the drivers scheduled at the times that cover the requirements of a particular interval sum to the number required. For the interval from time 0 to 4, drivers starting at time 20 of the previous day and time 0 of the current day cover the needs from time 0 to time 4.

Now using (3), we can formulate the following constraints:
$$x(20) + x(0) \geq 4$$
$$x(0) + x(4) \geq 8$$
$$x(4) + x(8) \geq 1$$
$$x(8) + x(12) \geq 7$$
$$x(12) + x(16) \geq 12$$
$$x(16) + x(20) \geq 4$$

Non-negativity constraints:

$x(0) \geq 0, x(4) \geq 0, x(8) \geq 0, x(12) \geq 0,$
$$x(16) \geq 0, x(20) \geq 0$$

So the objective function is

Minimize $z = x(0) + x(4) + x(8) +$
$$x(12) + x(16) + x(20)$$

We solve this model by using MATHEMATICA software input and output are given below:

Input:

$\text{Minimize}[x(0) + x(4) + x(8) + x(12) +$

$x(16) + x(20), x(20) + x(0) \geq 4 \text{ \&\& } x(0) +$

$x(4) \geq 8 \text{ \&\& } x(4) + x(8) \geq 10 \text{ \&\& } x(8) +$

$x(12) \geq 7 \text{ \&\& } x(12) + x(16) \geq$

$12 \text{ \&\& } x(16) + x(20) \geq 4 \text{ \&\& } x(0) \geq$

$0 \text{ \&\& } x(4) \geq 0 \text{ \&\& } x(8) \geq 0 \text{ \&\& } x(12) \geq$

$0 \text{ \&\& } x(16) \geq 0 \text{ \&\& } x(20) \geq$

$0, \{x(0), x(4), x(8), x(12), x(16), x(20)\}]$

Output
$\{26, \{ x(0) \rightarrow 4, x(4) \rightarrow 10, x(8) \rightarrow 0, x(12) \rightarrow 8, x(16)$
$\rightarrow 4, x(20) \rightarrow 0\}\}$

Therefore the minimum number of drivers is 26 to meet the schedule.

In the last section, we develop a model the hospital wants to minimize the total number of nurses to meet all shifts and all requirements. The model has been solved by using MATHEMATICA software.

VII. The Nurse Scheduling Problem

In this problem we must search for a solution satisfying as many wishes as possible while not compromising the needs of the hospital. Some examples of constraints are:

- A nurse doesn't work the day shift, night shift and late night shift on the same day (for obvious reasons).

- A nurse may go on a holiday and will not work shifts during this time.

- A nurse doesn't do a late night shift followed by a day shift the next day.

We now discuss an example about nurse schedule as follows:

The emergency unit of a hospital intends to plan a schedule for its nurses. The hospital schedules nurses for 16 hours shifts. The beginning times for the shifts are 6:00am, 2:00pm and 10:00pm. A nurse beginning a shift of these times works for the next 16 hours. During normal and week day operations, the number of nurses needed varies depending on the time of the day. The hospital stuffing guidelines require following minimum number of nurses on duty.

	6:00am-2:00pm	2:00pm -10:00pm	10:00pm – 6:00am
Time Slot	1	2	3
Nurses needed	8	7	5

The hospital wants to minimize the total number of nurses to meet all shifts and all requirements.

Solution

We need to develop a linear program for this problem that will determine the number of nurses that should be scheduled to begin the 16-hours shifts at each of three times (6:00am, 2:00pm and 10:00pm) in order to minimize the total number of nurses required.

Define Variable:

x_1 = the number of nurses that begin work at 6:00am and work shift (6:00am – 2:00pm) and (2:00pm- 10:00pm)

x_2 = the number of nurses that begin work at 2:00pm and work shift (2:00pm – 10:00pm) and (10:00pm – 6:00am)

x_3 = the number of nurses that begin work at 10:00pm and work shift (10:00pm- 6:00am) and (6:00am – 2:00pm)

For a sixteen hour shift ((10pm – 6am) + (6am-2pm)) we need to at least 5 nurses from (10pm – 6am) and 8 nurses from (6am-2pm)

Now using (3) we have $b_{j+1} = 8$. Therefore first constraint can be written as

$$x_3 + x_1 \geq 8$$

Similarly we can formulate other constraints for the remaining 16 hours schedules.

$$x_1 + x_2 \geq 7$$

$$x_2 + x_3 \geq 5$$

Non-negativity constraints:

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0$$

Define the objective function:

To minimize the total number of nurses needed to meet all shift requirements. We have

$$\text{Minimize } z = x_1 + x_2 + x_3$$

Using MATHEMATICA software, the optimal basic feasible solution is $x_1 = 5, x_2 = 2, x_3 = 3$. Therefore the minimum value of z is 10 that is, the minimum number of nurses is 10 to meet all shift requirements.

VIII. Conclusion

Scheduling Problem (SP) is an important decision making problem for larger organizations and industries. We all need to get the proper use of our limited resources. LP can play a significant role in this regard. In this paper, we mainly study application of LP to the scheduling problems. Some well-known real life oriented models have been presented namely, police scheduling problem, employee scheduling problem, nurse scheduling problem and bus driver scheduling problem. All the problems have been solved using MATHEMATICA software package. Therefore, it can be concluded from this work that the LP model can be incorporated in formulating and solving scheduling problems of various institutions to take proper decisions. We can further extend this work to solve various SPs of different organizations in our country's perspective.

References

1. Aickelin, Uwe & Dowsland, A. Kathryn, 2004. An Indirect Genetic Algorithm for a Nurse Scheduling Problem. Computers & operations Research **31(5)**, 761-778.

2. Akira Kawaguchi and Andrew Nagel, 2008. Linear Programming in Database, In Tech Open Publication, 340-342.

3. D. Applegate and W. Cook, 1991.A computational study of job-shop scheduling, ORSA Journal on Computing **3(2)**, 149-156.

4. Augustine, Lizzy, Faer, Morgan, Kavountzis, Andreas, Reema Patel, 2009.A Brief study of the Nurse Scheduling problem (NSP).

5. Burke E. K., T. Curtois, R. Qu, G. Vanden Berghe, 2010. A Scatter search methodology for the nurse rostering problems. J. oper. Res. Soc. **61**. 1667-1669.

6. Eiselt H. A. & C. L. Sandblom, 2007. Linear Programming and its Applications, Springer-Verlag Berlin Heidelberg Publication,80-81.

7. Ernst A. T., H. Jiang, M. Krishnamoorthy, and D. Sier,2004.Staff Scheduling androstering, A review of application, methods and models, European Journal of Operational Research, 153:3-27.

8. Fred Gloyer & Claude McMillan, 1986. The General Employee Scheduling Problem, Pergamon Journals Ltd,**13(5)**,563-564.

9. Gupta P. K. & D. S. Hira, 2005. Problems in Operation Research, S. Chand & Company Ltd.

10. Paul Jensen A. & F. Jonathan Bard, 2004. Operations Research Models and Methods, John Wiley and Sons, Inc.

11. Teresa G. D., P. S. Jorge, and F. C. 2001. A Genetic Algorithm for the Bus Driver Scheduling Problem, 4[th]Metaheuristics International Conference, 35-40.