## Bus Driver Scheduling Problem using TLBO and JAYA Algorithm

Atul B. Patil

M.Tech Student Department of Computer Science and Engineering, Rajarambapu Institute of Technology, Rajaramngar (Sakharale), MH, India

## ABSTRACT

Bus driver scheduling problem is one of most important and complex problem faced by many companies and bus terminals. This paper attempts to solve this problem using paramterless evolutionary algorithms, TLBO and JAYA algorithm. The objective of this paper is to assign the drivers to duty on a particular day and block duty by satisfying the constraints. Algorithms are tested on four randomly generated datasets. In the work solution is obtained with no zero constraint violations. JAYA algorithm gives better results than TLBO algorithm.

#### **General Terms**

Algorithms, Constraint Satisfaction Problems, Scheduling Problems.

#### Keywords

Bus Driver Scheduling Problem, Teaching Learning Based Optimization, JAYA algorithm.

## **1. INTRODUCTION**

Scheduling is the allocation of resources for objects, in such a way as to minimize the total cost of satisfying the required constraints for the particular scheduling problem. Bus transportation is most important part of public transportation system available to everyone with cheap cost. Hence, here need to address the different issues of Bus Depot Scheduling Problem (BDSP). The Bus Depot system can separate into stages which are design of network, bus timetabling, bus diver scheduling and vehicle scheduling [1, 2] and all these problems treated as independent complex and heavy. The bus driver scheduling problem is the problem of generating set of legal duties and selection of duty for cover the all blocks. In [2, 3, 7, 8] the Bus Driver Scheduling is determined based on minimize the cost approach for generating and selection of a set of legal duties. For these problems the different techniques are used such as standalone software [1], the genetic algorithm approach [3], mathematical model and GRASP approximate solution [4], randomized multistate heuristics [5], and similar kind of problem for truck driver scheduling problem [6] in which mixed integer programming formulation and an iterative dynamic programming approach to minimizing the duration of driver's problem is presented. The main complications in solving these kind of problem are its required constraint nature, and it is different for different types of problem and surly depends on the formulation of the objective function.

The bus driver scheduling problem is defined as set generation and selection of legal duty to minimize the cost of driver covers the trip. It also defined as set cover partitioning or set covering problem [2, 3, 7, 8]. The formulation of the personal scheduling (Bus driver, Truck driver, Nurses in the hospital) problem may have different approaches for different application such as Nurse scheduling problem, Laboratory personal timetabling problem, Security Personal Scheduling Problem defined in [9,10,11]. The objective of this work is the assignment of staff to the particular period and minimization of personal cost. The scheduling problems like driver scheduling problem, nurse scheduling problems cannot solve within reasonable time.

This paper presents BDSP problem using the TLBO algorithm and JAYA algorithm. The BDSP problem can be formulated in terms of the objective function and Hard as well as soft constraints. During the procedure one problem specific function is introduced to avoid the duplication assignments of drivers to duty. There are four data instance created randomly, based on this population is generated for both algorithms.

The rest of the paper is organized as follows: In Section 2 Evolutionary algorithm TLBO and JAYA algorithm is introduced. Section 3 represents the problem formulation. Section 4 gives the implementation details and last section 5 discusses briefly about the results.

## 2. TLBO AND JAYA ALGORITHM

In literature many traditional and heuristic algorithms are applied to solve constraint satisfaction problems. Traditional algorithms fails to reach the global optimal or near to global optimal solution due to large search space. Evolutionary algorithms like Genetic algorithms, Particle swam optimization are found to be better for many optimization and scheduling problem. Literature indicates that tuning of different algorithmic parameters is essential to get better solutions to the problem under consideration [3, 7, 8]. TLBO and JAYA algorithm are parameterless algorithms. They require only common controlling parameter like population size and number of generation.

# 2.1 Teaching Learning Based Optimization (TLBO)

TLBO algorithm is an optimization algorithm based on effect of the influence of teacher on the output of learner to improve the knowledge [12, 13]. TLBO algorithm does not have any algorithm specific parameter, hence it is parameter less algorithm. TLBO algorithm does not degrade the solution as like GA, PSO, and ACO etc. for the wrong selection of the algorithm specific parameter. The working procedure of this algorithm is divided into two phases: Teacher Phase and Learner Phase. The working of teacher phase is as follows: after initialization of population, calculate the mean of each subject and accordingly modify (improve) result of each student for improvement the overall mean of class. Mi is the mean of each subject, Ti is the teacher selected as Mnew. It tries to move mean Mi towards, its own level. The solution of the population is updated according to the difference between the existing main and the new meaning given by

$$Difference Mean_{i} = r_{i}(M_{new} - T_{f}M_{i})$$
(1)

Where,

 $r_i = any random number between 0 and 1$  $T_f = Teaching factor$ 

 $M_i$  = it is mean of population at particular iteration

 $M_{new}$  = best result that is teacher in population

In equation (1) the value of teaching factor is probably the part which decides the mean value to be changed. The value of ri is random no within range of 0 and 1. The value of teaching factor TF can be either 1 or 2 and decided randomly with equal probability as,

$$T_f = round[1 + rand(0,1)\{2 - 1\}]$$
(2)

This difference modifies the solution according to the following equation

$$X_{new} = X_{old} + Difference\_Mean$$
(3)

The working of Learning Phase as follows:

Learner can increase their knowledge by following ways:

- $\checkmark$  Input through the teacher to increase the knowledge
- $\checkmark$  Another way is the interaction between them

Select randomly another  $X_i$  and  $X_j$  such way that  $(i \neq j)$ 

If 
$$f(X_i) < f(X_j)$$
  
 $X_{new} = X_{old,i} + r_i(X_i - X_j)$   
Else  $f(X_i) > if f(X_j)$   
 $X_{new} = X_{old,i} + r_i(X_i - X_i)$ 

Accept  $X_{new}$  if it gives a better function value and check termination condition, also finalize the solution.

## 2.2 JAYA Algorithm

JAYA algorithm [14], is another powerful paramterless algorithm for finding optimal solutions. In this algorithm also only common control parameter is required like TLBO algorithm that is population size and number of generation size. The major advantage of the algorithm is at every iteration JAYA algorithm is rapidly moving towards an optimal solution because it removes the worst solution in every iteration. So JAYA algorithm always tries to improvise the solution to avoid the worst solution. JAYA algorithm has only one phase and it is relatively simpler to use for any specific problems. The working of the JAYA algorithm is a lot different from JAYA algorithm. Working procedure of JAYA contains one phase and it is as follows:

First of all needs to initialize the population size, iteration number which is a common control parameter. Then identify the best and worst solution for a given population. The important step of the algorithm is a modification of the solution which uses following equation number (4) by evaluating the best and worst result for the particular objective function and type problem defined for minimization or maximization. r1 and r2 are any two random numbers in between 0 to 1.

$$X_{new} = X_{old,i} + r_{1,i} \left[ \left( X_{best,i} - \left| X_{old,i} \right| \right) \right] - r_{2,i} \left[ \left( X_{worst,i} - \left| X_{worst,iold,i} \right| \right) \right]$$

Based on equation number 4 all the solution of the population is updated, and according to the set objective function of minimization or maximization the best solution is accepted for the particular iteration and taken this solution of the population for further iteration.

Major strength of JAYA algorithm is its procedure to improvise of result by removing the worst result at every iteration and then next section presents the information about the problem formulation.

## 3. BUS DRIVER SCHEDULING PROBLEM (BDSP)

One of the difficulties in solving scheduling problem is constraints of the problem changes with organization, region, legal issues etc. In literature different researchers attempted bus driver scheduling / rostering problem [3, 7, 8, 15] which are specific to some enterprise / city. For this work, the visit is arranged to Islampur (District Sangali, Maharashtra) Bus Depot to understand the current system and based on that formulated the problem under consideration. This section presents the assumptions and constraints of the problem. Objective of the bus driving scheduling is to allocate the drivers to buses by fulfilling legal, organizational constraints and requirements.

#### 3.1 Assumptions

- i. Number of drivers and buses available are known.
- ii. **Block:** It is set of trips of a particular bus. Two types of blocks are considered with respect to time and total distance of all trips.
  - a. **Normal Blocks:** The block consisting the set of trips and has duration 8hr's to 9hr's duration.
  - b. **Long Blocks:** The block consisting the set of trips and has duration more than 9hr's.

Following are the objective function, Hard Constraint and Soft Constraint for the BDSP problem.

#### 3.2 Objective Function

The main objective is the assignment of driver for the particular duty on a particular day by minimization of the personal cost of driver's assignment in violation of constraint. The objective function for the problem is represent as follows

$$min(z) = \sum_{i=1}^{n} (H_i * a) + \sum_{j=1}^{m} \left( S_j * b \right)$$
(5)

- Where,
- H is i<sup>th</sup> hard constraint and a is the penalty cost for hard constraint.
- S is j<sup>th</sup> soft constraint and b is the penalty cost respectively for soft constraint

Solution of the problem requires the characteristics by based on task and drivers.

- Based on task:
  - Normal duty task
  - Long duty task

- Planning set of duties
- Based on driver characteristics
  - Skill set (based on performance observation by depot manager)
  - Availability in planning periods
  - Based on group of driver (based on region and it's working capacity)

The above defined points are importantly taken into consideration before finalization of hard and soft constraints. Hard Constraints are those must be satisfied. The violation of these constraints causes degrade the quality of solution. The Soft constraints are those that are desirable to produce good quality of solution but violation are allowed to satisfy the hard constraint.

#### **3.3 Hard Constraint**

1. Driver does not have more than one duty per day.

- 2. Driver does not work more than 6 days.
- 3. The driver is assigned to duty according age factor. (Aged people can work only normal duty).

#### 3.4 Soft Constraint

- 1. The driver is assigned to duty accordance to long duty and normal duty. (based on working hr.'s, performance verification by depot manager)
- 2. The driver can work in depot based on duty type and driver working group.
- 3. Equal distribution of duties to driver.
- 4. Every driver must have exact 1 non-working duty in depot.

In this work data instance is generated randomly based on which initial population are generated. Next chapter shows details about the implementation details.

#### 4. IMPLEMENTATION DETAILS

TLBO algorithm and JAYA algorithm is implemented through MATLAB R2014a. In the work source code of TLBO algorithm used which is developed by Prof. R.V.Rao, S.V NIT, Surat to solve the bus driver scheduling problem.

The Table 1 shows the Input data instance. Data instance are consider for 7 days problem. Table 1 shows the input dataset for the problem where Table 2 shows the predefined data based on assumptions described in 3.4 for the Input dataset1. Likewise the data is prepared for input dataset2, dataset3 and dataset4. In the implementation all 4 Inputs are used. For this implementation randomly population is generated for the allocation of driver to duty for particular day. Table 3 shows 1 sample population for input 1.

| Table 1. | . Dataset | for im | plementation | of | problem |
|----------|-----------|--------|--------------|----|---------|
|----------|-----------|--------|--------------|----|---------|

| Input           | Input1 | Input2 | Input3 | Input4 |
|-----------------|--------|--------|--------|--------|
| Population Size | 20     | 20     | 20     | 20     |
| No of iteration | 200    | 200    | 200    | 200    |
| No of Blocks    | 10     | 21     | 28     | 35     |

| Total Driver | 14 | 15 | 20 | 25 |
|--------------|----|----|----|----|
| No of Days   | 7  | 7  | 7  | 7  |

Table 2. Dataset for implementation of problem

| nb=[1,3,7,8,10]        | lb=[2,4,5,6,9]        |
|------------------------|-----------------------|
| nbd=[2,4,6,8,10,12,14] | lbd=[1,3,5,7,9,11,13] |
| eb=[3,7,8,4,5]         | wb=[1,2,6,9,10]       |
| ed=[4,6,10,7,9,11,13]  | wd=[1,2,3,5,8,12,14]  |
| srd=[2,4,6,8]          |                       |

#### Where

| -, | nb-norm                | nal dut    | y block                 | lb- long di          | uty blo | ock    |  |
|----|------------------------|------------|-------------------------|----------------------|---------|--------|--|
|    | nbd-normal duty driver |            |                         | lbd-long duty driver |         |        |  |
|    | eb-east s              | side bl    | ock                     | wb-west s            | ide blo | ock    |  |
|    | ed-east                | side<br>sı | driver<br>d- senior dri | wd-west<br>ver group | side    | driver |  |

According the inputs the data population is generated in matrix format as block vs days. So the Concept of block is in such way that the depot has arranged bunch of trips according to rule of time and distance need to travel by the driver. So depends on time and distance of each trip, block contains n number of trips. This Block is divides into long duty block and normal duty block, the senior group drivers can allocate to the only normal duty blocks so all this assumption of data shown in Table 2 and Table 3.

In the population Normal duty (N) and long duty(S) is considered which will be 8-9 hr.'s duration and more than 9 hr.'s duration. So in TLBO algorithm and JAYA algorithm driver number can be work as design variable which will update in various iteration and results are collected in terms cost of violation of constraints by applying the penalty. Table 4 shows the penalty cost.

| Table 3. | Sample | population | for | problem |
|----------|--------|------------|-----|---------|
|----------|--------|------------|-----|---------|

| Shift | b/d | D1 | D2 | D3 | D4 | D5 | D6 | D7 |
|-------|-----|----|----|----|----|----|----|----|
| Туре  |     |    |    |    |    |    |    |    |
| Ν     | B1  | 1  | 7  | 2  | 2  | 8  | 7  | 7  |
| L     | B2  | 9  | 7  | 13 | 7  | 6  | 6  | 10 |
| Ν     | B3  | 3  | 12 | 6  | 1  | 11 | 7  | 7  |
| L     | B4  | 2  | 6  | 4  | 6  | 4  | 14 | 10 |
| L     | B5  | 2  | 5  | 12 | 4  | 2  | 8  | 12 |
| L     | B6  | 11 | 11 | 3  | 11 | 6  | 10 | 10 |
| Ν     | B7  | 12 | 3  | 14 | 6  | 10 | 12 | 7  |
| Ν     | B8  | 1  | 3  | 7  | 3  | 9  | 5  | 4  |
| L     | B9  | 1  | 4  | 3  | 11 | 13 | 10 | 12 |
| Ν     | B10 | 3  | 14 | 13 | 1  | 2  | 8  | 2  |

Table 4. Penalty Cost for Hard and Soft Constraint

| Constraint      | Penalty Cost |
|-----------------|--------------|
| Hard Constraint | 5            |
| Soft Constraint | 1            |

Before provides the actual input to the problem it is important to introduce new function for removing the duplicate assignments of driver to duty. The constraint no 4 from soft constraint comes in existence at last by checking Hard constraint 1,2and 3 as well as soft constraint 1, 2, 3. By consideration above mentioned input, population it provides to both algorithms and results are collected and described in next chapter.

#### 5. RESULT AND DISSCUSION

This section gives details obtained by implementation of TLBO and JAYA algorithm for BDSP problem and comparative results are presented.

Table 5. Experimental results with Different input files (Time and Cost)

| Input   | Time (in Sec | cond)    | Cost |      |  |
|---------|--------------|----------|------|------|--|
|         | TLBO         | JAYA     | TLBO | JAYA |  |
| Input 1 | 880.598      | 467.951  | 113  | 96   |  |
| Input 2 | 1156.136     | 778.5946 | 189  | 65   |  |
| Input 3 | 1636.668     | 1136.757 | 240  | 215  |  |
| Input 4 | 1852.659     | 1132.002 | 326  | 294  |  |

From Table 5 it clears that TLBO algorithm can moved towards near optimal solution. With comparison of TLBO and JAYA algorithm, Jaya algorithm can gives best results in terms of near optimal solution best cost for the all specified inputs as shown in Table 5.

TLBO algorithm can contains two phases' that is teacher phase and learner phase and solution is updated and accepted as shown in 2.1 and equation 1, 2 and 4. It contains more computations for solution update. Where as in JAYA algorithm there is only one phase as shown in 2.2 and it contains only one equation that is equation 4. The JAYA algorithm can provides the best results (Best cost) as shown for different inputs in fig 1, fig 2, fig3 and fig.4. The main reason is that at each iteration JAYA algorithm improve the result by removing worst result of each population and hence it moves toward optimal solution rapidly. Also another reason is the computational procedure in both TLBO and JAYA algorithm. It takes less time for computation than TLBO algorithm as shown in Table 5.

So JAYA algorithm gives the near optimal solution for the problem than TLBO algorithm. These are the points which affects for the BDSP problem's solution taken by the both algorithms.



Fig 1: Iteration vs Best Cost for Input 1



Fig 2: Iteration vs Best Cost for Input 2



Fig 3: Iteration vs Best Cost for Input 3.



Fig 4: Iteration vs Best Cost for Input 4

The Fig 1 shows the output result of input 1 and the fig 2 shows the result of input 2 which contains data as 15 blocks, 15 driver and 7 days. The fig 2 shows that the rapid movement toward near optimal solution by JAYA algorithm. Also TLBO algorithm can rapidly move towards the near optimal solution but solution obtains constant cost value for each iteration after some interval. The same kind of result nature is clearly visible from fig 2 as shown the result for fig 1, but there is less vibrant results in this figure. As problem size is varies the solution moves toward near optimal after some interval of iteration result varies suddenly which tries to move to words near optimal solution as shown in fig3 and fig 4. Another problem occurs that after half part of iteration that algorithm can give same best cost after some particular iteration shows in all fig 1, 2, 3 and 4

So importantly, JAYA algorithm can gives best result for the BDSP problem than TLBO algorithm. As the block size and driver size of the algorithm can increase the JAYA algorithm work efficiently for obtained the near optimal solution where as TLBO algorithm can gives vibrant solutions. The reason is same for this that at every iteration the JAYA algorithm removes the worst result and helps to move towards the better results.

## 6. CONCLUSION

In the work, TLBO and JAYA algorithm is implemented for BDSP problem. Both algorithm are paramterless algorithm, so there is less chance for degrading the solution. For the defined problem formulation, the obtained results are one of the solution to near optimal solution. In the implementation the randomly population is generated so there is no unique result obtained by the algorithm. These algorithm can be tested on different input dataset which shows the vibrant behaviour result by TLBO algorithm. JAYA algorithm gives better results with steady nature in every iteration. With increasing the block size and driver size the TLBO reacts and provide the worst solutions, due to its large number of computation. There is no any procedure in TLBO algorithm for removing the worst solution from the population which helps to move towards near optimal solution as like TLBO algorithm.

This work is based on assumption of Bus driver data prepared randomly. In the algorithm one function is introduced to remove the unwanted assignments of driver to particular duties. That causes algorithm gives the possibly near optimal solution. So from obtained, result it clear that it is necessary to check the problem specific points to introduced in algorithm causes the results may improve. The factor such as time and preferences by the drivers' for particular duty and holiday is not considered. So with this factor the problem complexity should be increased. It is very challenging task to find solution for the Bus driver Scheduling Problem by introducing problem specific points, function as one kind of function introduce for removing unwanted assignments.

## 7. REFERENCES

- Chun, H.W., 1997, December. A distributed constraintbased search architecture for bus timetabling and duty assignment. In Software Engineering Conference, 1997. Asia Pacific... and International Computer Science Conference 1997. APSEC'97 and ICSC'97. Proceedings (pp. 82-88). IEEE.
- [2] Ma, J., Liu, T. and Song, C., 2012, December. A Lagrangian relaxation-based heuristic for the bus driver scheduling problem: A case study of beijing. In Information Science and Control Engineering 2012 (ICISCE 2012), IET International Conference on (pp. 1-5). IET.
- [3] Dias, T.G., Sousa, J.P. and Cunha, J.F., 2001, July. A genetic algorithm for the bus driver scheduling problem. In 4th Metaheuristics International Conference.

- [4] De Leone, R., Festa, P. and Marchitto, E., 2011. A Bus Driver Scheduling Problem: a new mathematical model and a GRASP approximate solution. Journal of Heuristics, 17(4), pp.441-466.
- [5] De Leone, R., Festa, P. and Marchitto, E., 2011. Solving a bus driver scheduling problem with randomized multistart heuristics. International Transactions in Operational Research, 18(6), pp.707-727.
- [6] Goel, A., 2012. The Canadian minimum duration truck driver scheduling problem. Computers & Operations Research, 39(10), pp.2359-2367.
- [7] Valouxis, C. and Housos, E., 2002. Combined bus and driver scheduling. Computers & Operations Research, 29(3), pp.243-259.
- [8] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.41.2393&rep=rep1&type=pdf
- [9] Adamuthe, A.C. and Bichkar, R., 2011. Hybrid Genetic Algorithmic Approaches for Personnel Timetabling and Scheduling Problems in Healthcare. In International Conference on Technology Systems and Management.
- [10] Adamuthe, A.C. and Bichkar, R.S., 2012, October. Tabu search for solving personnel scheduling problem. In Communication, Information & Computing Technology (ICCICT), 2012 International Conference on (pp. 1-6). IEEE.
- [11] Adamuthe, A.C., Mane, S.U. and Thampi, G.T., 2012, October. Genetic algorithmic approach for security personnel scheduling. In Communication, Information & Computing Technology (ICCICT), 2012 International Conference on (pp. 1-6). IEEE.
- [12] Rao, R.V., Savsani, V.J. and Vakharia, D.P., 2012. Teaching–learning-based optimization: an optimization method for continuous non-linear large scale problems. Information Sciences, 183(1), pp.1-15.
- [13] Rao, R.V., Savsani, V.J. and Vakharia, D.P., 2011. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-Aided Design, 43(3), pp.303-315.
- [14] Rao, R., 2016. Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. International Journal of Industrial Engineering Computations, 7(1), pp.19-34.
- [15] Moz, M., Respício, A. and Pato, M.V., 2009. Biobjective evolutionary heuristics for bus driver rostering. Public Transport, 1(3), pp.189-210.