International Symposium on Research in Innovation and Sustainability 2014 (ISoRIS '14) 15-16 October 2014, Malacca, Malaysia
Sspecia; Issue

OPTIMIZATION IN DRIVER's SCHEDULING FOR UNIVERSITY SHUTTLE BUS USING HARMONY SEARCH

Zatul Alwani Shaffiei[1], Zuraida Abal Abas [2], Ahmad Fadzli Nizam Abdul Rahman[2], Abdul Samad Shibghatullah[2] and Burairah Hussin[2]

Optimization, Modelling, Analysis, Simulation and Scheduling (OptiMASS) Research Group
Faculty of Information and Communication Technology,
Universiti Teknikal Malaysia Melaka (UTeM),
Hang Tuah Jaya, 76100 Durian Tunggal, Melaka
Email: [1]zatul.alwani@gmail.com, {[2]zuraidaa, fadzli, samad, burairah}@utem.edu.my

ABSTRACT: *This paper is about driver scheduling which is categorized as a difficult combinatorial problem. Driver scheduling is the process of assigning shift to the drivers based on hard and soft constraints. The proposed technique is Harmony Search, which is a recently developed population-based meta heuristic optimization algorithm. This study will be implemented to UTeM Shuttle Bus. The number of drivers involved is about 25. It is viewed that the objective function evaluated by Harmony Search is less than a manual solution. Therefore, the result produced for this project is quite promising since the objective function obtained is better than the real schedule which is done manually.*

KEYWORDS: Driver**'s** Scheduling, Optimization, Harmony Search, Meta Heuristic

1.0 INTRODUCTION

Universiti Teknikal Malaysia Melaka (UTeM) is a university located in Melaka and it has three campuses spread within the locality of the state of Melaka. Further, since this university has yet to have a permanent hostel for students, students enrolled in the university stay in hostels outside the three campuses. In this situation, shuttle buses have been used to transport students from the hostels to the respective campuses so that they can attend classes, lectures and any activities in campus. At the university, the Departement of University Vehicle Management, known as the BPKU (its acronym in Malay language), is responsible to prepare the schedule of the shuttle bus to ensure that efficient transportation is provided for the students. It has been a common practice for this department to prepare the shuttle bus schedule every semester according to students' timetable and placement. In planning the bus schedule, it is crucial to determine the duties, tasks or shifts of the drivers. Considering that BPKU prepare the schedule manually every month, it is difficult to plan an optimum and balanced schedule for the drivers. Hence, this study focuses on the problem of driver scheduling (DSP). There are various methods that have been proposed to solve driver (crew) scheduling problems. Lourenco *et al.* [1] brought a multi objective crew scheduling model, which was tackled using a tabu search technique, meta heuristics, and genetic algorithms. Valouxis and Housos [3] presented a quick search approach that used the main algorithm such as the cost matching, set partitioning and shortest path. Chen and Niu [2] proposed a tabu search algorithm to solve bus crew scheduling problem, which considers early day and late duty modes with time shift and work intensity constraints. While Chen and Li [4] and Dias *et al.* [5] used genetic algorithm in their research, Li and Kwan [6] used hybrid genetic algorithm, with fuzzy set theory as their research approach. Belen *et al.* [7] proposed two approaches to obtain solutions, which are the Grasp algorithm and Scatter Search algorithm.

In this study, the proposed technique is Harmony Search, which is a recently developed population-based meta heuristic optimization algorithm by Geem *et al.* [8][9]. This paper is organized as follows: Section 2.0 defines the DSP and Section 3.0 presents the proposed harmony search algorithm. This is followed by Section 4.0 that presents the results, and finally, concluding remarks are presented in Section 5.0.

2.0 DRIVER SCHEDULING PROBLEM (DSP)

DSP can be defined as the process of assigning the shift to drivers based on two types of constraints which are the hard constraints (HC) and soft constraints (SC) as shown in Table 1. In creating the schedule, the hard constraints must be satisfied, but not all of the the soft constraints need to be satisfied. However, in order to produce a better and more optimum result than the existing schedule, the violation of soft constraints should be minimized [8]. There are three objectives of this study: To generate a driver schedule automatically using Harmony Search, to formulate DSP into mathematical equation and to compare the result with the existing schedule.

Table 1: DSP Constraints

HC	Assign one type of shift and route per week for each driver.
	Weekly coverage demand of each shift type and route needs to be fulfilled.
SC	Assigns each driver two weeks for morning shift and two weeks for evening shift in a month.
	Assigns each driver with different routes for every week in a month.

3.0 THE HARMONY SEARCH ALGORITHM FOR DSP

HSA is one of the meta heuristic methods that has been recently developed by Geem *et al.* It mimics the improvisation process of music players [10]. In music performance, each musician plays one musical note at a time. These musical notes are combined together to form a harmony, measured by aesthetic standards [11]. The procedure of HSA has the following five main steps with pseudo code:

3.1 Initialization of HSA parameters

The initialization value of HMCR (harmony memory consideration rate (accumulative search)), pitch adjustment rate (PAR, analogous to local search) and BW (bandwidth) is very important because these values will be used during the improvisation process to produce new values.

3.2 Initialize the Harmony Memory (HM)

```
begin
    for i = (1 to WEEK) do /* WEEK is the number
        of weeks */
        for j = (1 to HMS) do
            $x^j = ø$;
            for k = (1 to n) do /* n is the number
                of drivers */
                choose one pattern randomly for driver k
                based on demands
            endfor
            evaluate the soft constraints
            calculate the objective function value $f(x^j)$
            add $x^j$ to HM
        endfor
    endfor
end
```

Table 2: Combined Pattern of Route and Shift

Patterns	Combination of Route and Shift	Demand (driver needed)
0	Route 1 Shift M	5
1	Route 1 Shift E	2
2	Route 2 Shift M	3
3	Route 2 Shift E	2
4	Route 3 Shift M	2
5	Route 3 Shift E	2
6	Route 4 Shift M	1
7	Route 4 Shift E	1
8	Route 5 Shift M	1
9	Route 6 Shift M	3
10	Route 7 Shift M	2
11	Route 7 Shift E	1

With reference to the pseudo code above, the patterns are randomly initialized based on demand as shown in Table 2. The drivers are assigned for each pattern. For example, five drivers (demand) are assigned to Pattern 0 and so on. The numbers of drivers assigned to these routes are based on the total number of students in each hostel and peak hours. It is called as demand. In this step, the objective function is calculated after the evaluations of soft constraints are made. The equations below show how the objective function is calculated. The mathematical formulation for DSP constraints is provided as below, *i* denotes a driver, *j* denotes a shift, *k* denotes a route, *l* denotes a week and *x* denotes a schedule solution:

HC1: $$\sum j \in J \sum k \in K \; x_{ijkl} = 1; \forall i \in I, \forall l \in L \qquad (1)$$

HC2: $$\sum i \in I \; x_{ijkl} = d_{jk}; \forall l \in L$$

where d_{jk} is coverage demand of shift j and route k

(2)

SC1: $$\sum j \in J \sum l \in L \; x_{ijkl} = 2; \forall i \in I$$

(3)

SC2: $$\sum k \in K \sum l \in L \; x_{ijkl} = 1; \forall i \in I$$

(4)

Based on the problem formulation for soft constraints, the objective function is formed, *c* denotes a soft constraint, *fc* denotes value to soft constraint c, *fc(X)* denotes value given by the soft constraint *c* applied to the solution *X* and *mc* denotes value given by the soft constraint c applied to the solution *X*. The goal is to find a solution for *X* that satisfies *fc(X) > mc* for every soft constraint *c*.

Minimize $c_{soft} = \sum c \in C(x_{ijkl}) \; f_c(X)$ subject to:

Equation (1), (2), (3) and (4)

Equation 1 and 2 above refers to the mathematical equation of hard constraints, while equation 3 and 4 refers to mathematical equation of soft constraints. Therefore, to consider this objective function, equation 3 and 4 is closely related since the objective function is to minimize the soft constraint violation. From the objective function above, the summation is the overall value of soft constraint c and subtracted by 1 to get a value that approaches 0.

3.3 Improvise New Solutions

```
begin
    for i = (1 to NI) do /* NI is the maximum number of
        improvisations*/
        $x^{new}$= ø
        for j = (1 to n) do /* n is the number of drivers */
            if (rand (0, 1) ≤ HMCR) then
                choose pattern randomly from the HM
                if (rand(0,1) ≤ PAR) then
                    adjust the chosen pattern according to
                    bandwidth (BW)
                    /* BW = {1,-1}; is used to change the
                    index of the selected shift pattern */
                else
                    add the chosen shift pattern to solution
                    $x^{new}$ without changing
                endif
            else
                choose pattern randomly from the patterns
                pool
            endif
                repeat process until fulfilled the demand
                add the chosen shift pattern to solution
                $x^{new}$
        endfor
    endfor
end
```

Table 3: 10, 100, 1000 and 10000 iterations with HMCR = 0.1, PAR = 0.1

Number of Iteration	HMCR	PAR	Objective Function				Convergence Rate at th iteration
			HMS 1	HMS 2	HMS 3	HMS 4	
10	0.1	0.1	0.34	0.34	0.35	0.35	10
100	0.1	0.1	0.28	0.31	0.31	0.31	98
1000	0.1	0.1	0.29	0.29	0.29	0.29	427
10000	0.1	0.1	0.26	0.27	0.27	0.27	8913

Table 4: 10, 100, 1000 and 10000 iterations with HMCR = 0.5, PAR = 0.3

Number of Iteration	HMCR	PAR	Objective Function				Convergence Rate at th iteration
			HMS 1	HMS 2	HMS 3	HMS 4	
10	0.5	0.3	0.36	0.36	0.36	0.37	8
100	0.5	0.3	0.30	0.30	0.31	0.31	68
1000	0.5	0.3	0.27	0.27	0.28	0.29	456
10000	0.5	0.3	0.26	0.27	0.27	0.27	9070

Table 5: 10, 100, 1000 and 10000 iterations with HMCR = 0.9, PAR = 0.9

Number of Iteration	HMCR	PAR	Objective Function				Convergence Rate at th iteration
			HMS 1	HMS 2	HMS 3	HMS 4	
10	0.9	0.9	0.34	0.37	0.37	0.38	10
100	0.9	0.9	0.30	0.31	0.31	0.32	71
1000	0.9	0.9	0.28	0.28	0.29	0.29	411
10000	0.9	0.9	0.23	0.26	0.26	0.26	6546

The pseudo code above shows the algorithm for improvisation in HS. The new pattern and objective function will be compared with the existing objective function in HM. If the value of new objective function is less than the value of objective function in the existing HM, it will replace the worst one. The improvisation is based on HMCR, PAR and randomization. All of the new patterns produced for every week will be combined together to calculate the objective function. The improvisation for each driver will be iterated until it fulfills the demand for every week.

3.4 Update the HM

```
begin
  for i = (1 to HMS) do
      find the highest value of objective function (the worst
          one) in HM
      compare the objective function of NCHV $x^{new}$ with the
          worst one $x^j$
  endfor
  if ($x^j > x^{new}$)
      replace $x^j$ with $x^{new}$
  else
      remain $x^j$ in HM
  endif
  sort HM in ascending order based on f(x)
end
```

After NCHV is produced and new objective function is obtained, it will be compared to the objective function in the existing HM. If the new objective function is less than the objective function in the existing HM, the worst one in HM will be eliminated and replaced with a new NCHV. However, if the new objective function is more than the objective function in the existing HM, there will be no replacement or elimination. Subsequently, the HM needs to be sorted in ascending order.

3.5 Check the Termination Criteria

Step 3 and 4 will be iterated until it reaches the termination criteria. In this project, the numbers of iteration set are 10, 100, 1000 and 10000 to see the difference between these changes iteration. However, if more iteration occurs, the result produced will be better.

4.0 RESULT

There are two comparisons made in this section. First, the comparison of result for different number of iteration, and second the comparison of the schedule produced using HS with the existing schedule.

4.1 Comparison Between Different Number of Iterations and Parameters

Table 3, 4 and 5 shows the result obtained for different number of iterations and different parameters. From these three tables, the best value obtained is when the parameters for HMCR and PAR are equal to 0.9 with 10000 iterations and the result produced has a slight diference. This means that the changes of parameters do not significantly affect the result. However, it is clear that the more iteration occurs, the better the result is produced. The average of the best objective function is 0.26. The convergence rate is also included in the three tables, which means the objective function produced during the debugging starts to become static and there was no changes. For example, for iteration

1000, at 427th iteration in Table 3, the objective function produced for each HMS starts to become static until the last iteration. From these three tables, it shows that with higher HMCR, the the faster the achievement of the convergence rate.

4.2 Comparison of Real Schedule and Schedule using HS

Table 6 below shows the comparison of result using HS and real schedule in UTeM. There are 5 real data that have been compared which are from March, April, May, September and October. The objective function gained for schedule using HS is 0.23. This is better than the objective function in real schedule, which are 0.32, 0.33, 0.33, 0.37 and 0.41. Even though the implementation of HS technique towards DSP is not really promising because of the result produced is still unachievable, the most optimum value, which is 0 still outperforms the objective function obtained from the real schedule. Therefore, it is clear that the HS technique has been successfully applied and it gives better result. Other advantage of using HS technique is from the aspect of time consuming.

Table 6: Comparison of Existing Schedule and Schedule Produced Using HS

Schedule	Objective Function
Existing (March)	0.37
Existing (April)	0.41
Existing (May)	0.33
Existing (September)	0.33
Existing (October)	0.32
Schedule using HS	0.23

5.0 CONCLUSION

This paper has proposed the harmony search algorithm for solving the driver scheduling problem. All of the objectives stated in Section 2.0 are achieved. A recently developed population-based meta heuristic optimization algorithm, HSA which is inspired by the improvisation process of musicians developed by Zong Woo Geem in 2001, seemed to be a quite promising technique in solving optimization problems since the results produced outperformed and produced better schedule than real schedule for DSP. For future works, hybridization with other techniques in this driver scheduling problem will be applied to get an excellent and better result.

ACKNOWLEDGEMENTS

This work was supported by research grants RAGS/2013/FTMK/ICT02/01/B00039 under Ministry of Education of Malaysia.

REFERENCES

[1] H. R. Lourenço, "Metaheuristics for The Bus-Driver Scheduling Problem," pp. 1–26, 2005.

[2] M. Chen and H. Niu, "A Model for Bus Crew Scheduling Problem with Multiple Duty Types," *Discrete Dynamics in Nature and Society*, vol. 2012, pp. 1–11, 2012.

[3] C. Valouxis and E. Housos, "Combined bus and driver scheduling," vol. 29, no. January 2000, 2002.

[4] Q. Chen and C. Li, "An Approach to Bus-Driver Scheduling Problem," *2010 Second WRI Global Congress on Intelligent Systems*, pp. 379–382, Dec. 2010.

[5] T. G. Dias, "A Genetic Algorithm for the Bus Driver Scheduling Problem," pp. 35–40, 2001.

[6] J. Li and R. S. K. Kwan, "A fuzzy genetic algorithm for driver scheduling," *European Journal of Operational Research*, vol. 147, no. 2, pp. 334–344, Jun. 2003.

[7] M. Belén, V. García, B. B. Zanón, and E. C. Rodríguez, "Combining Metaheuristic Algorithms to Solve a Scheduling Problem," 2012.

[8] M. A. Al-Betar, A. T. Khader, and M. Zaman, "University Course Timetabling Using a Hybrid Harmony Search Metaheuristic Algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 5, pp. 664–681, Sep. 2012.

[9] G. V. Loganathan, "A New Heuristic Optimization Algorithm: Harmony Search," *Simulation*, vol. 76, no. 2, pp. 60–68, Feb. 2001.

[10]R. Diao and Q. Shen, "Feature selection with harmony search.," *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics : a publication of the IEEE Systems, Man, and Cybernetics Society*, vol. 42, no. 6, pp. 1509–23, Dec. 2012.

[11]M. Hadwan, M. Ayob, N. R. Sabar, and R. Qu, "A harmony search algorithm for nurse rostering problems," *Information Sciences*, vol. 233, pp. 126–140, Jun. 2013.