

## Heuristic constructive algorithm for work-shift scheduling in bus rapid transit systems

César Augusto Marín Moreno<sup>a</sup>, Luis Miguel Escobar Falcón<sup>b</sup>, John Willmer Escobar<sup>c\*</sup>, Antonio Hernando Escobar Zuluaga<sup>a</sup> and Mauricio Granada Echeverri<sup>a</sup>

<sup>a</sup>Faculty of Engineering, Universidad Tecnológica de Pereira, Colombia

<sup>b</sup>Department of Systems Engineering, Universidad Libre Pereira, Colombia

<sup>c</sup>Department of Accounting and Finance, Universidad del Valle, Colombia

### CHRONICLE

*Article history:*

Received March 12, 2019  
Received in revised format:  
March 29, 2019  
Accepted April 10, 2019  
Available online  
April 10, 2019

*Keywords:*

*Crew Scheduling Problem  
Heuristic Techniques  
Massive Public Transport*

### ABSTRACT

This paper proposes a two-phase heuristic algorithm to solve the crew scheduling problem of the Megabus Bus Rapid Transit System. In the first stage, a division of the original schedules is performed using a recursive algorithm based on dynamic scheduling. In the second stage, work-shift construction based on graph theory is performed using a pairing algorithm (i.e., matching). The method is validated by applying it to the mass transit system of the Central Western Metropolitan Area (AMCO), operated by Integra SA, which serves 11 routes for a daily total of 2899 trips.

© 2018 by the authors; licensee Growing Science, Canada.

## 1. Introduction

The crew scheduling problem (CSP), or driver scheduling problem (DSP), is part of the operational planning of public transport systems and refers to the daily tasks assigned to the drivers required to cover a set of trips assigned to a set of vehicles (i.e., the vehicle scheduling problem (VSP)). That is, given a set of trips within a fixed period, the problem is to generate the schedule of feasible work shifts, including start and end times and various types of driver task for a series of trips. The work shifts must completely cover the operation schedules of all the vehicles to be used (Edmonds, 2965).

The creation of work shifts for the subsequent assignment of drivers is one of the most complex tasks faced by transport companies and requires a large amount of time to manage. This task is a problem of substantial interest because the worker payroll represents one of the highest costs in the budget of a public transport system (Ceder, 2007).

The importance of the CSP to public passenger transport companies motivates the development of new variants that fit the particular environment of each company. However, regardless of the different variables that each company confronts, the objective is always the fulfillment of scheduled itineraries

\* Corresponding author. Tel.: +57 300 5349187

E-mail address: [john.wilmer.escobar@correounivalle.edu.co](mailto:john.wilmer.escobar@correounivalle.edu.co) (J. W. Escobar)

and the reduction of human resources-related costs through optimization processes. Extensive review of the CSP problem could be consulted in Ernst et al. (2004a), Ernst et al. (2004b) and Xie et al. (2015). In addition real cases application of the CSP could be checked in Ma et al. (2016).

Each itinerary is a description of trips that must be executed within specific time periods and geographic sections (i.e., routes). The frequency with which an itinerary is operated corresponds to service conditions and mass transit public service needs as determined by the tactical planning of the entity that manages the massive public transport system. The combination of route and departure time is termed a service, and a group of services in the same section is defined as a timetable. The routes of public transport systems are determined by their strategic planners and typically do not undergo substantial change in the short or medium term.

Each route must be served by a vehicle with a certain frequency at a defined average speed. That is, a route must be defined in the tactical planning of a transport system given that all route requirements are based on the design of the network service (i.e., network route design) and precisely designed to meet needs identified in strategic planning.

During recent decades, the problem of scheduling driver work shifts has been extensively addressed to reduce costs, improve the distribution of duties and decrease the number of shifts. Several of the first studies of this type are presented by Assad (1983), (Alefragis P. G., 1998) and (Alefragis P. S., 2000). The CSP becomes increasingly difficult to solve as constraints or variables are added to the problem, even when the constraints or variables requirements are simple (Ibarra-Rojas, 2015).

One study on the DSP was performed in (Fores, 2002). The authors discuss the assignment of drivers and propose a method that combines overall linear scheduling with a heuristic technique to solve the problem without requiring prohibitive computing time. The method is successfully applied in planning scenarios of large-scale public transport systems.

In (Zhao, 2006), a heuristic method termed ZEST for ESTimator is used to analyze the DSP. The author divides the problem into two time periods (i.e., afternoon and morning) according to each day's hours of maximum demand. Each sub-problem is solved separately, and the solutions are only united at the end of the process to provide a general solution. The described method uses three sequential steps to examine the structure of a bus schedule created through a manual scheduling process used in the United Kingdom by expert programmers.

In the doctoral thesis presented in (Weider, 2007), a method is proposed based on an algorithm known as IS-OPT that integrates the problems of vehicle and driver assignment in passenger public transport in Germany. The author demonstrates how to replace a linear scheduling (LS) method with an inaccurate, proximal one in the context of the generation of columns. Through Lagrangian relaxation, a new set of activities is presented that approximates a large-scale set partitioning problem (SPP). Finally, a two-phase algorithm to solve the pricing problem is proposed, which is modeled as a shorter path problem with additional constraints and a nonlinear objective function. The method can solve real medium-size cases.

In (Steinzen, 2007), the author describes a hybrid evolutionary algorithm with different column generation methods to evaluate the objective function of each individual. This strategy facilitates solving an integrated vehicle and driver assignment problem while considering multiple public transport hubs. The algorithm is based on a decomposition problem that first assigns trips to each hub and thus resolves an integrated problem for each hub. Each sub-problem can be solved in polynomial time. This method is capable of resolving problems of medium size.

A method that combines the algorithms of iterative local search, descending variable neighborhood search and tabu search with adaptive relaxation to solve an integrated vehicle and driver assignment problem in mass public transport is presented in (Souza, 2010). To investigate the solution space, the

authors propose six movements of relocation and exchange, both for a VSP and a DSP. Tests were performed with real data from a transport system of a Brazilian city. The results reflect the method's effectiveness.

In (Mesquita, 2011) is presented a new mathematical expression to describe a joint vehicle-crew-rostering problem (VCRP). The authors propose a sequential algorithm that facilitates solving a nonlinear multi-objective optimization model. The algorithm awards high priority to the objectives framed within the combined VCRP, while the difficulties presented by the driver rostering problem are awarded low priority. Within the methodology, the authors develop a heuristic procedure that facilitates decision-making concerning vehicle, driver and roster scheduling while attempting to comply as closely as possible with company interests and policies as well as driver preferences. The method is validated using real data from a bus company in Portugal.

A new mathematical expression to solve the CSP under a set of special constraints according to the labor and transport laws of Italy is presented in (De Leone, 2011). The proposed model can accurately solve small and medium-size problems. For real size problems, the authors propose a greedy randomized adaptive search procedure (GRASP). The method is applied to a set of real problems, and the results are compared with the current operation of the system under study.

Chen and Niu present in (Chen, 2012) a design of a binary linear scheduling model to solve a DSP considering various task types, which differ according to the time they are performed. To this end, the authors divide the duties of each driver by time period (i.e., morning, afternoon and evening) according to the hours of maximum demand. With this model, the authors seek to minimize the break time of drivers under a set of complicated constraints that correspond to work regulations. The objective function has direct implications for operating costs. The model is solved using a tabu search metaheuristic algorithm with a neighborhood structure that enables the search space to be adequately investigated.

A flexible and efficient method based on cutting and joining (CAJ) to solve a CSP is presented in (Tóth, 2013). This algorithm enables initial solutions to be quickly generated. Because of the flexibility of the proposed approach, it is possible to develop an iterative heuristic method with which a more robust algorithm is obtained to solve the DSP in real instances.

Recent papers dealing with variants of the CSP problem have been proposed by Boyer et al. (2018), Sargut et al. (2017), Xie et al. (2017), Ma et al. (2017) and Xie & Suhl (2015). The Flexible Vehicle and Crew Scheduling Problem for urban bust transport are studied by Boyer et al. (2018). A mixed-integer linear programming model and a variable neighborhood search have been proposed for this problem showing the efficiency of the proposed approaches by considering a large set of instances.

Sargut et al. (2017) consider a real planning problem for public bus transportation. A multi-objective set partitioning problem is proposed considering crew rostering and assignment decisions. Xie et al. (2017) also considers multi-objective crew rostering problem with the weighted sum of all objectives. Using several methods such as ant colony optimization, simulated annealing, and tabu search methods solves the problem.

Ma et al. (2017) propose a model considering fairness in the problem of crew scheduling for bus drivers (CSP-BD). The considered problem is solved by using a hybrid ant-colony optimization approach. Xie & Suhl (2015) proposed a method for generating a personalized roster for each driver/group of drivers in a Bus System Transit. The problem is formulated as a multi-commodity network flow problem.

Finally, in this paper, a constructive two-phase heuristic algorithm is proposed. In the first stage, the division of the shift schedule (i.e., timetable) based on a dynamic scheduling algorithm is performed. In the second phase, a matching algorithm is implemented (i.e., matching) to construct the work shifts of drivers at Integra SA, which operates the Megabus system of the Central Western Metropolitan Area

(AMCO). The approach consists of minimizing the scheduling time of driver work shifts, maximizing compliance with labor considerations and ensuring compliance with all constraints. A final result is presented that is viable in terms of actual operations.

## 2. Problem description and formulation

The problem of work-shift generation considered here can be formulated in a general way using graph  $G = (V, A)$ , where  $V = \{0, 1, 2, \dots, n, n + 1\}$  represents the nodes of graph  $G$ . Node 0 and node  $n + 1$  represents the same location, which is where the set of available drivers is to start the various trips required. In this way, set  $V' = 1, 2, \dots, n$  represents the set of tasks or services that must be performed. Each task  $i$  must be performed by a driver between start time  $s_i$  and final time  $e_i$ . The set of arcs  $A$  contains all the pairs  $(i, j)$  such that task  $j \in V$  can be performed after task  $i \in V$ . Each arc has an associated travel time  $\tau_{ij} \leq (s_j - e_i)$  and a travel cost  $d_{ij}$ , which is the time a driver spends traveling from where task  $i$  ends to where task  $j$  begins.

$R = (0, i_1, i_2, \dots, i_h, n + 1)$  is defined as a path in  $G$ , which starts at node 0, traverses a set of tasks  $i \in V'$  and ends at node  $n + 1$ . Thus,  $i_1$  represents the first node  $V'$  visited after leaving node 0, and  $i_h$  represents the last visited node  $V'$  before reaching node  $n + 1$ . Therefore,  $R$  is a feasible path of  $G$ , which represents a work shift to be worked by a driver. Each path  $R$  must meet the following constraints:

$$\tau_{0i_1} + (e_{i_h} - s_{i_1}) + \tau_{i_h n+1} \leq ET \quad (1)$$

Eq. (1) refers to the real time a driver requires to execute a work timetable, that is, the time that elapses from the start of the driver's workday, including the departure of the vehicle from the depot (that is, from node 0) until the vehicle reaches node  $n + 1$ , where the timetable ends (which is not necessarily a hub) and whose duration must be less than maximum elapsed time  $ET$ . This is because a driver can perform part of a shift in the morning and another part in the afternoon or evening. This constraint avoids shifts that require all the driver's time and avoids forcing a driver who ends a shift late at night to wake up early (and vice versa).

$$\sum_{i \in R} w_i \leq T \quad (2)$$

where  $w_i = e_i - s_i$ .

Constraint (2) refers to the maximum work time  $T$  a driver must perform in a workday.

In contrast, in the literature, different objectives have been considered when solving the CSP, such as minimizing driver break time, minimizing the number of drivers and maximizing work time. In this study, we seek to distribute the work time of each driver uniformly. To this end, the difference between the regulated daily working time and the actual working time of each driver is minimized, as expressed in the following equation:

$$Z = \min \left\{ \sum_{i \in C} |T - T_i| \right\} \quad (3)$$

Where  $T$  corresponds to the regulated daily working time and  $T_i$  is the actual work time performed by driver  $i \in C$ .

The objective function presented in (3) seeks similar working times for all drivers, which ensures that human resources are used appropriately (i.e., the time worked approaches the regulated working time) and endeavors to eliminate the driver cost associated with overtime pay.

### 3. Method

Based on the various service tables of  $TT$  (i.e., timetables) offered by the managing entity (here, Megabus), the problem consists of determining the set of work shifts that must be performed by the drivers. The tables assigned by Megabus to the operating company Integra S.A. vary in size (in time and number of trips) and normally cannot be assigned to a single driver, either because the cumulative time of all trips exceeds the regulated number of work hours (in terms of the time that a driver may work continuously and in terms of the regulated daily working time). Therefore, each service table must be appropriately partitioned such that each partition coincides with the regulations concerning continuous work  $T_s$  and such that the daily shift does not exceed the maximum allowed working time, for which it is necessary to pair one or more partitions. In Figure 1, the proposed method is described in a general way. Thus, Figure 1a shows that the process starts with a set of schedules (i.e., timetables), each of which contains a set of trips or services. Figure 1b illustrates the process of dividing the tables into sub-tables (i.e., sub-paths in  $G$ ), which comply with a set of technical working constraints. Finally, Figure 1c shows the pairing process according to which two or more sub-paths  $r$  can be joined to form one path  $R \in G$  that represents a task or work shift for a driver.

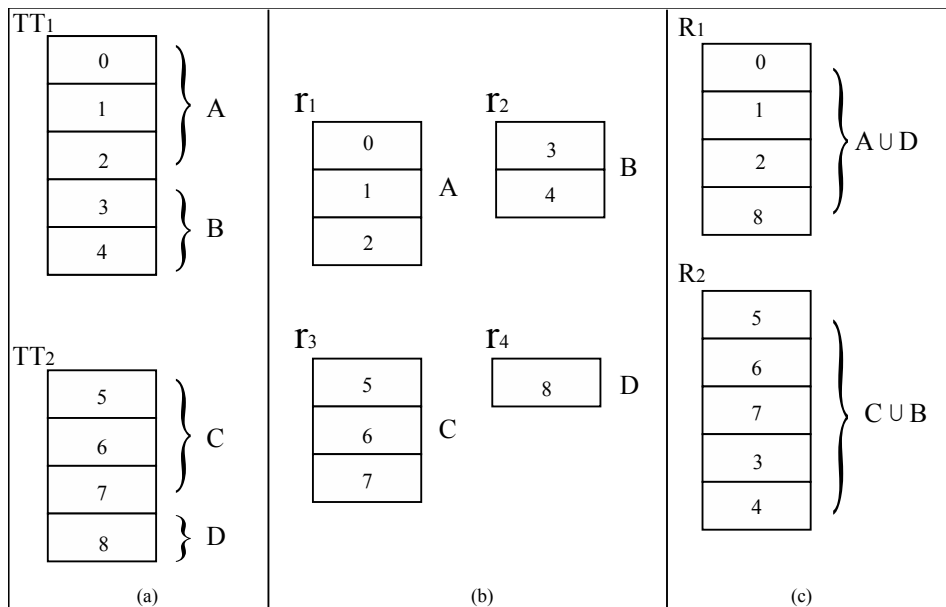


Fig. 1. Method

#### 3.1 Table division algorithm

Because of the technical and labor constraints included in the optimization process, it is necessary to perform a division process for each timetable. Because of the number of possible ways to divide the tables and because there is a set of strict operational constraints, this process is combinatorial:

Fig. 1. The first trip of the work shift generated by the division must start at the depot (node 0).

Fig. 2. The last trip of the work shift generated by the division must end at a hub or the depot (node  $n + 1$ ).

Fig. 3. The sum of the times  $w_i$  of all trips included in a division  $r$  must be less than or equal to the maximum time  $T_s$  a driver may work continuously, as expressed in Eq. (4)

$$\sum_{i \in r} w_i \leq T_s \tag{4}$$

It is clear that a timetable can be divided in several ways and into several sub-tables  $r$ . Therefore, in this paper, we propose a recursive algorithm based on dynamic scheduling to find the best set of subdivisions for each timetable. To determine if division  $a$  is better than another division  $b$  (for the same table  $TT_i$ ), the following comparison measure is used in the recursive process:

$$\min \left\{ \sum_{r_a \in TT} \left( T_s - \sum_{i \in r_a} w_i \right)^2 ; \sum_{r_b \in TT} \left( T_s - \sum_{j \in r_b} w_j \right)^2 \right\} \quad (5)$$

As an example, suppose that there are two different divisions for the same timetables  $TT_a$  and  $TT_b$  and each of them has the time of each service (Fig. 2). Assuming  $T_s = 45$ , the algorithm will determine the best division from the expression presented in 5) as follows:

$$\min(TT_a, TT_b)$$

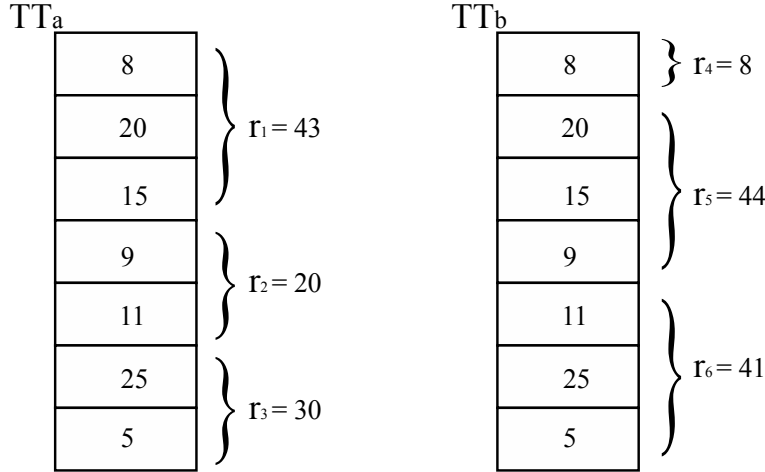
$$\min(\{(45 - 43)^2 + (45 - 20)^2 + (45 - 30)^2\}, \{(45 - 8)^2 + (45 - 44)^2 + (45 - 41)^2\})$$

(5')

$$\min(\{(2)^2 + (25)^2 + (15)^2\}, \{(37)^2 + (1)^2 + (4)^2\})$$

$$\min(854, 1242)$$

In this manner, it can be determined that the best division of the service table is that which occurs in partition  $a$  of the Fig. 2, which includes the set of sub-paths  $\{r_1, r_2, r_3\}$ .



**Fig. 2.** Better division.

The pseudocodes of the table division process are shown in Algorithm 1 and in Algorithm 2.

**Algorithm 1** Division of tables

**Require:**  $TT, T_s, N \leftarrow size(TT)$ .

**Ensure:** Better set of divisions

- 1: **for** Each Table in  $TT$  **do**
- 2:     Divisions  $\leftarrow$  Divisions  $\cup$  DynamicScheduling(0,0);
- 3: **end for**
- 4: **return** Divisions

**Algorithm 2** Dynamic Scheduling

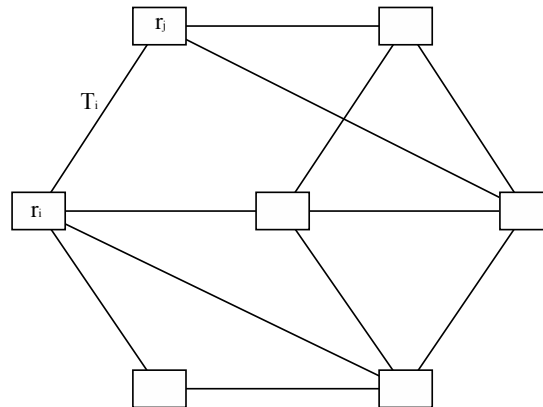
```

1: DynamicScheduling(Pos x, Pos y);
2: if ( $x == \text{size}(\text{Table}) \parallel y == \text{size}(\text{Table})$ ) then
3:   if  $x == y$  then
4:     return List of empty divisions
5:   else
6:     return null
7:   end if
8: end if
9: if  $M_{x,y} \neq \text{null}$  then
10:  return  $M_{x,y}$ 
11: end if
12:  $DT \leftarrow \text{DynamicScheduling}(y+1, y+1)$ 
13: if ( $DT == \text{null}$ )  $\parallel$  ( $\text{Duration}(DT) > T_S$ )  $\parallel$  ( $y \neq \text{depot}$ ) then
14:   $A \leftarrow \text{null}$ 
15: else
16:   $A \leftarrow A \cup DT$ 
17: end if
18:  $B \leftarrow \text{DynamicScheduling}(x, y+1)$ 
19: if  $A == \text{null}$  then
20:   $M_{x,y} \leftarrow B$ 
21:  return  $B$ 
22: else
23:  if  $B == \text{null}$  then
24:     $M_{x,y} \leftarrow A$ 
25:    return  $A$ 
26:  end if
27: else
28:   $\text{Better} \leftarrow \min(A, B) // \text{According to equation (5)}$ 
29:  return  $\text{Better}$ 
30: end if

```

**3.2 Matching algorithm**

With the set of sub-tables or sub-paths  $r$  obtained in the division process of section 3.1, graph  $G' = (V'', A')$  is formed (see Fig. 3).



**Fig. 3.** Graph of sub-tables  $r$

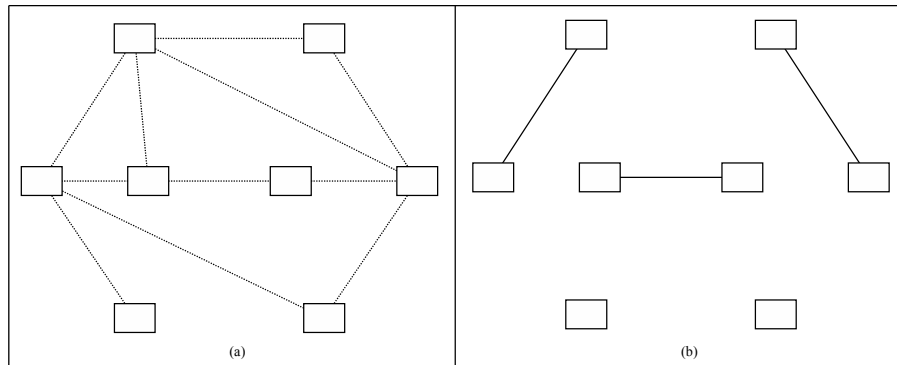
Each node in set  $V''$  represents a partition or sub-path  $r$ . The set of edges  $A'$  is formed by the pair  $(i, j)$  with  $i, j \in V''$  if the union of two sub-tables  $(r_i \cup r_j)$  generates a feasible  $R$  work shift. That is, each shift  $r$  that is generated must meet the following set of constraints:

1. The total work time of shift  $R = r_i \cup r_j$  must be less than or equal to the maximum work time  $T$ , as indicated by the set of constraints imposed in Eq. (2).
2. The number of mixed shifts (which arise by joining a sub-table of schedules pertaining to a feeder line with a sub-table of a main line) must be less than or equal to a parameter  $MaxMix$ . This constraint is due to the fact that not all drivers are licensed to operate articulated buses.
3. The amount of time required to perform a shift must be less than or equal to  $ET$ , as stipulated in constraint (1), which ensures that a driver who works late must not wake up early (and vice versa).
4. The time between the completion of one partition  $r_i$  and the beginning of another  $r_j$  must be greater than rest time  $T_d$  plus the time of travel between the locations where travel ends  $r_i$  and  $r_j$  begins, which ensures a break for each driver after having worked consecutively for a period of time  $T_s$ .

The second phase of the proposed method is based on graph  $G'$ , for which a matching algorithm is developed based on the algorithm proposed by Jack Edmonds in (Edmonds, 1965).

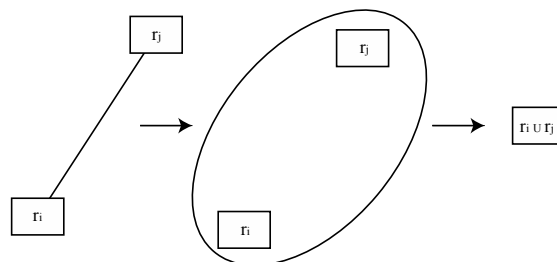
The method consists of making different pairings recursively on the same graph, as described in the following:

- A simple pairing of nodes is performed according to graph  $G'$  considering probability  $\rho_1$ . Each shift formed can only contain trips from up to two different original tables, and mixed shifts are not allowed. Fig. 4 shows a general matching process.



**Fig. 4.** Matching

- From the previous pairing, “super nodes” are created. That is, the edge used in a pairing is contracted and considered the union of  $r_i$  with  $r_j$  as a single node (see Figure 5). Again, the previous step is repeated with probability  $\rho_2$ . The new shifts may contain services from up to three different original tables, and mixed shifts are not allowed.



**Fig. 5.** Super nodes.



- The previous process is repeated recursively eight times with probabilities  $\rho_1$ ,  $\rho_2$ ,  $\rho_3$  and  $\rho_4$ . In some of the recursive calls, mixed tables are allowed. Algorithm 3 shows the pseudocode of the pairing process.

**Algorithm 3** Matching

**Require:** Divisions,  $T$ , MaxIter.

**Ensure:** Work shifts

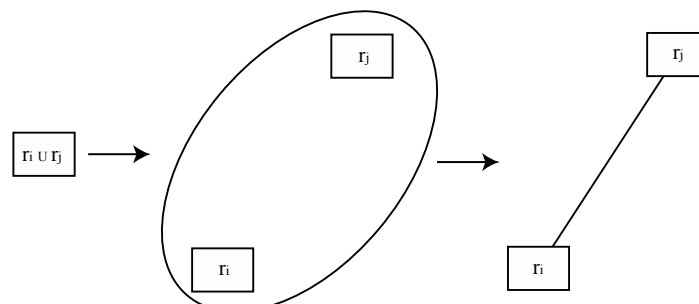
```

1: Matching (List of Divisions, Time  $T$ );
2: Create graph  $G'$  with all divisions
3: while  $Iter \leq MaxIter$  do
4:   for each Edge in  $A'$  do
5:     if  $random < \rho$  then
6:       Perform perturbation mechanism
7:     end if
8:   end for
9:   //The Edmonds Algorithm can be found in (Edmonds, 1965)
10:   $G' \leftarrow Edmonds(G', Double, \rho_1, Mixed = False)$ 
11:   $G' \leftarrow Edmonds(G', Triple, \rho_2, Mixed = False)$ 
12:   $G' \leftarrow Edmonds(G', Double, \rho_3, Mixed = True)$ 
13:   $G' \leftarrow Edmonds(G', Triple, \rho_3, Mixed = True)$ 
14:   $G' \leftarrow Edmonds(G', Double, \rho_1, Mixed = True)$ 
15:   $G' \leftarrow Edmonds(G', Triple, \rho_2, Mixed = True)$ 
16:   $G' \leftarrow Edmonds(G', Double, \rho_3, Mixed = False)$ 
17:   $G' \leftarrow Edmonds(G', Triple, \rho_4, Mixed = False)$ 
18:  if  $G'$  is better than  $BestSolution$  then
19:     $BestSolution \leftarrow G'$ 
20:  end if
21: end while
22: return BestSolution

```

### 3.3 Perturbation mechanism

To further examine the solution space, a perturbation mechanism is applied to the solution found. This mechanism consists of separating the shifts found. For this purpose, the probability is  $\rho$ . The process is inverse to the super nodes. That is, a super node is adopted, and the original nodes of graph  $G'$  are recovered, which provides the possibility that these nodes are paired differently. Figure 6 illustrates the perturbation process.



**Fig. 6.** Perturbation process

## 4. Results

The proposed algorithm was implemented in Java 1.8 on an Intel Core i5 CPU with 2.4 GHz and 4 GB RAM in the Yosemite 10.10 operating system. The proposed method was validated by application to the work shift schedule of the AMCO mass transportation system operated by Integra SA. The system has 11 routes between feeders and main lines. The managing entity designs and delivers 47 scheduling tables, 26 of which correspond to feeder lines and 21 to main lines, for a total of 2899 services and a

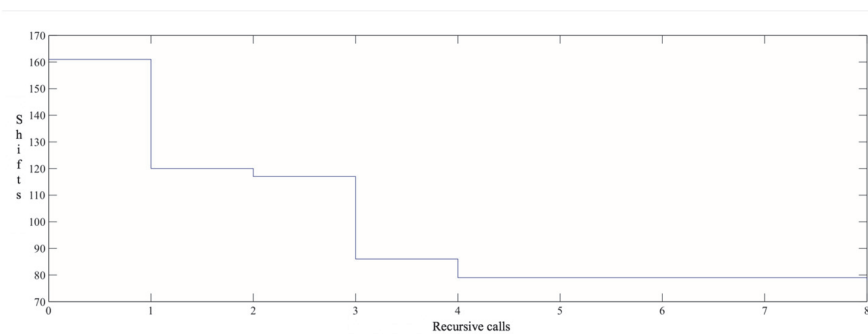
total programmed time of 692 hours and 20 minutes. The tests were performed during 10 simulation processes, and the following set of parameters was used in each execution of the proposed algorithm:  $\rho = 0.30$ ,  $\rho_1 = 0.25$ ,  $\rho_2 = 0.50$ ,  $\rho_3 = 1.00$ ,  $\rho_4 = 0.10$ ,  $T_S = 5.50$  h,  $T = 10$  h,  $T_D = 40$  min,  $ET = 13$  h,  $MaxMix = 40$ , and  $MaxIter = 1000$ .

For each simulation, 1000 iterations of the algorithm were performed. Table 1 reports the results of the 10 simulations, including the number of final shifts. The final shifts are divided into double and triple shifts, which correspond to shifts consisting of trips from two and three different original tables, respectively. Mixed shifts correspond to those shifts in which the final table consists of trips on a feeder line and trips on a main line. Shifts shorter than eight hours correspond to the tables that do not exceed the minimum daily work time. Finally, the execution time of each simulation is presented. The best solution found by the algorithm corresponds to simulation number 5, which reports a total of 78 work shifts. That is, 78 drivers are required to perform all the trips scheduled by the managing entity. Although there are other solutions with the same number of shifts, this solution is better because the number of shifts of fewer than eight hours of work is four, which is a good indicator of efficient crew use.

**Table 1**  
Results

Simulation	Shifts	Double Shifts	Triple Shifts	Mixed Shifts	Shifts<8h	Time [s]
1	78	23	55	36	7	6,34
2	79	24	55	38	6	6,27
3	79	23	56	38	5	5,81
4	78	23	55	36	8	6,39
<b>5</b>	<b>78</b>	<b>24</b>	<b>54</b>	<b>38</b>	<b>4</b>	<b>6,38</b>
6	78	23	55	37	7	5,86
7	80	24	56	38	5	6,23
8	79	22	57	38	5	5,83
9	78	23	55	38	6	6,14
10	78	23	55	37	9	5,91

Fig. 7 shows the evolution of work shifts in the best solution. The graph shows that before starting the pairing process the number of shifts is high (161). As the algorithm progresses in each of the recursive calls according to the parameters used, the number of shifts begins to decrease until the obtaining the best solution reported in simulation 5 of Table 1.



**Fig. 7.** Number of work shifts.

It should be noted that the Integra SA shift schedule was manually created by the company's operations department, which consists of an expert team. These manual results are compared with the results obtained in terms of the number of shifts found and in the time required to complete the shift schedule. Thus, the manual scheduling has a total of 82 work shifts, and the operations team required approximately 24 hours to create the schedule. However, through the proposed method, a total of 78 work shifts are obtained. This outcome indicates that the company can save 4.878% of the cost of the drivers. In addition, the answer is obtained in 6.38 seconds. Thus, scheduling changes can be made

without affecting the operations of the transport system. Previously, updating a route involved several hours of manual changes. Thus, driver scheduling efficiency is improved.

For illustrative purposes, company savings are calculated based on a reduction of four drivers under normal company operations after the application of the proposed method. Therefore, if the method had been implemented in 2015, the average monthly cost of a driver was \$ 1'978,949 COP, and the calculated salary increases with the average rates of legal wage increases of the last three years in Colombia were as shown in Table 2. The remaining operation time of the company is eight years, and the opportunity rate is 6%.

**Table 2**  
Salary increase

Year	Salary Increase (%)
2012	4.02
2013	4.50
2014	4.60
Average	4.37

Table 3 shows in detail the increase in the salary of each driver projected over the subsequent eight years based on the average annual increase in the minimum wage in Colombia for the last three years. The table also shows the net value of the four drivers who were eliminated and the value saved in the period  $t = 8$ . In this manner, a savings of \$ 1'083,109.607 COP over the next eight years for Integra SA in terms of driver payroll in year 8 is determined.

**Table 3**  
Results

Period $t$	Salary per Driver (\$COP)	Number of Drivers	Annual Salary (\$COP)	Value saved in the period $t = 8$ (\$COP)
1	1'978.949	4	94'989.552	142'829.165
2	2'065.495	4	99'143.762	140'637.321
3	2'155.826	4	103'479.649	138'479.649
4	2'250.107	4	108'005.159	136'354.025
5	2'348.512	4	112'728.585	134'261.548
6	2'451.220	4	117'658.581	132'201.182
7	2'558.420	4	122'804.183	130'172.434
8	2'670.309	4	128'174.820	128'174.820
Total				1'083.109.607

## 5. Conclusion

In this article, a constructive heuristic algorithm with two phases was presented that facilitates solving the problem of generating work shifts in the AMCO Bus Rapid Transit System. The algorithm's first stage is based on dynamic scheduling. In its second phase, a pairing algorithm and a perturbation mechanism enable the investigation of the solution space. The proposed method enables work shifts to be generated for the drivers of the operator Integra SA, finding as a solution a total of 78 work shifts for a total of 2,899 services. The payroll cost of drivers is reduced compared to the manual method used by the company, which generated 82 work shifts. That is, previously, the company required 82 drivers, while with the implementation of this method, 78 are required. The algorithm can find a solution in less than seven seconds. Thus, schedule changes to the routes can be made quickly and without affecting transport system operations.

Future research on this topic should focus on the implementation of evolutionary metaheuristic algorithms with local improvement mechanisms that facilitate maintaining sub-optimal solutions.

## References

- Assad, A., Ball, M., Bodin, L., & Golden, B. (1983). Routing and scheduling of vehicles and crews. *Computers & Operations Research*, 10(2), 63-211.
- Alefragis, P., Goumopoulos, C., Housos, E., Sanders, P., Takkula, T., & Wedelin, D. (1998, September). Parallel crew scheduling in PAROS. In *European Conference on Parallel Processing* (pp. 1104-1113). Springer, Berlin, Heidelberg.
- Alefragis, P., Sanders, P., Takkula, T., & Wedelin, D. (2000). Parallel integer optimization for crew scheduling. *Annals of Operations Research*, 99(1-4), 141-166.
- Boyer, V., Ibarra-Rojas, O. J., & Ríos-Solís, Y. Á. (2018). Vehicle and crew scheduling for flexible bus transportation systems. *Transportation Research Part B: Methodological*, 112, 216-229.
- Ceder, A. (2007). *Public transit planning and operation: Theory. Modeling and practice*. Oxford.
- Chen, M., & Niu, H. (2012). A model for bus crew scheduling problem with multiple duty types. *Discrete Dynamics in Nature and Society*, 2012.
- De Leone, R., Festa, P., & Marchitto, E. (2011). A bus driver scheduling problem: a new mathematical model and a GRASP approximate solution. *Journal of Heuristics*, 17(4), 441-466.
- Edmonds, J. (1965). Paths, trees, and flowers. *Canadian Journal of Mathematics*, 17(3), 449-467.
- Ernst, A. T., Jiang, H., Krishnamoorthy, M., & Sier, D. (2004a). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research*, 153(1), 3-27.
- Ernst, A. T., Jiang, H., Krishnamoorthy, M., Owens, B., & Sier, D. (2004b). An annotated bibliography of personnel scheduling and rostering. *Annals of Operations Research*, 127(1-4), 21-144.
- Fores, S., Proll, L., & Wren, A. (2002). TRACS II: a hybrid IP/heuristic driver scheduling system for public transport. *Journal of the Operational Research Society*, 53(10), 1093-1100.
- Ibarra-Rojas, O. J., Delgado, F., Giesen, R., & Muñoz, J. C. (2015). Planning, operation, and control of bus transport systems: A literature review. *Transportation Research Part B: Methodological*, 77, 38-75.
- Ma, J., Ceder, A., Yang, Y., Liu, T., & Guan, W. (2016). A case study of Beijing bus crew scheduling: a variable neighborhood-based approach. *Journal of Advanced Transportation*, 50(4), 434-445.
- Ma, J., Song, C., Ceder, A. A., Liu, T., & Guan, W. (2017). Fairness in optimizing bus-crew scheduling process. *PloS One*, 12(11), e0187623.
- Mesquita, M., Moz, M., Paías, A., Paixão, J., Pato, M., & Respício, A. (2011). A new model for the integrated vehicle-crew-rostering problem and a computational study on rosters. *Journal of Scheduling*, 14(4), 319-334.
- Sargut, F. Z., Altuntaş, C., & Tulazoğlu, D. C. (2017). Multi-objective integrated acyclic crew rostering and vehicle assignment problem in public bus transportation. *OR Spectrum*, 39(4), 1071-1096.
- Souza, M. J. F., Ribas, S., & Coelho, I. M. A hybrid metaheuristic algorithm for the Integrated Vehicle and Crew Scheduling Problem. Working Paper.
- Steinzen, I., Becker, M., & Suhl, L. (2007, September). A hybrid evolutionary algorithm for the vehicle and crew scheduling problem in public transit. In *Evolutionary Computation, 2007. CEC 2007*. IEEE Congress on (pp. 3784-3789). IEEE.
- Tóth, A., & Krész, M. (2013). An efficient solution approach for real-world driver scheduling problems in urban bus transportation. *Central European Journal of Operations Research*, 21(1), 75-94.
- Weider, S. (2007). *Integration of vehicle and duty scheduling in public transport*. Cuvillier Verlag.
- Xie, L. (2015). Crew Rostering: State of the Art. In *Decision Support for Crew Rostering in Public Transit* (pp. 25-36). Springer Gabler, Wiesbaden.
- Xie, L., & Suhl, L. (2015). Cyclic and non-cyclic crew rostering problems in public bus transit. *OR Spectrum*, 37(1), 99-136.
- Xie, L., Merschformann, M., Kliewer, N., & Suhl, L. (2017). Metaheuristics approach for solving personalized crew rostering problem in public bus transit. *Journal of Heuristics*, 23(5), 321-347.
- Zhao, L. (2006). A heuristic method for analyzing driver scheduling problem. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 36(3), 521-531.

