

A hybrid algorithm for the multi-depot vehicle scheduling problem arising in public transportation**César Augusto Marín Moreno^a, Luis Miguel Escobar Falcón^{a*}, Rubén Iván Bolaños^a, Anand Subramanian^b, Antonio Hernando Escobar Zuluaga^c and Mauricio Granada Echeverri^c**^aUniversidad Tecnológica de Pereira – Integra S.A., Colombia^bUniversidade Federal da Paraíba, Brazil^cUniversidad Tecnológica de Pereira, Colombia**CHRONICLE***Article history:*

Received December 18 2018

Received in Revised Format

January 26 2019

Accepted February 3 2019

Available online

February 4 2019

*Keywords:**Vehicle Scheduling**Mathheuristics**Set Partitioning**Tactical Planning**Bus Rapid Transit***ABSTRACT**

In this article, a hybrid algorithm is proposed to solve the Vehicle Scheduling Problem with Multiple Depots. The proposed methodology uses a genetic algorithm, initialized with three specialized constructive procedures. The solution generated by this first approach is then refined by means of a Set Partitioning (SP) model, whose variables (columns) correspond to the current itineraries of the final population. The SP approach possibly improves the incumbent solution which is then provided as an initial point to a well-known MDVSP model. Both the SP and MDVSP models are solved with the help of a mixed integer programming (MIP) solver. The algorithm is tested in benchmark instances consisting of 2, 3 and 5 depots, and a service load ranging from 100 to 500. The results obtained showed that the proposed algorithm was capable of finding the optimal solution in most cases when considering a time limit of 500 seconds. The methodology is also applied to solve a real-life instance that arises in the transportation system in Colombia (2 depots and 719 services), resulting in a decrease of the required fleet size and a balanced allocation of services, thus reducing deadhead trips.

© 2019 by the authors; licensee Growing Science, Canada

1. Introduction

From a technical point of view, the Operational Planning of public transportation systems includes scheduling of work shifts for bus drivers, scheduling of preventive maintenance work, scheduling of the services, as well as personnel allocation for each shift. Recently, this task has involved the control in real time of the operational system fleet with the incorporation of Communication and Information Technologies. Each one of the aforementioned problems represents a challenge when they are faced by the companies. All of these problems have already been studied widely in the specialized literature, and due to their computational complexity (they are classified as NP-hard), they are usually approached independently. This work deals with the Multiple Depot Vehicle Scheduling Problem (MDVSP), in which a set of vehicles must perform a set of trips with a given frequency at specific moments during the day. When examining the vehicle scheduling literature, we identified several works in the context of public transportation system. It was verified that applying optimization approaches contributed to the development of an efficient transportation system, capable of meeting the mobility needs required in cities Ibarra-Rojas et al. (2015).

* Corresponding author Tel.: +573217938145

E-mail: luisescobarf@utp.edu.co (L. M. Escobar Falcón)

The creation of itineraries for each vehicle is one of the most complex tasks in transportation planning and is known to be NP-hard. Despite its difficulty, this has been a problem of great interest given that the cost represented by the vehicles, either by their acquisition or by use, is one of the highest in the budget of operation of public transportation systems (Ceder, 2007). The reality of public transportation companies highlights the importance of efficiently solving the MDVSP, thus motivating the study of new alternatives that fit the particular environment of each company dedicated to the operation of public transportation services. However, regardless of the particularities of each scenario, the objective is likely to be framed in the total fulfillment of itineraries and reduction of costs related to system operation. Each itinerary or route is a description of the trips that must be done in specific times and areas, satisfying a frequency according to the conditions of the service and the needs of the mass transportation service determined by tactical planning. Therefore, the combination of route and departure times is denoted as service and a group of services in the same area is defined as timetable. The public transportation system routes are defined by their strategic planning and usually do not have substantial changes in the short and medium terms.

Each route must be served within a given frequency, at a defined average speed. This must be defined in the strategic planning of the transportation system, since all these route requirements are defined from the design of the service network itself and are designed precisely to meet the requirements identified during the strategic planning. In this work, we propose a matheuristic algorithm to solve an Operational Planning problem of a Public Transportation System that can be modeled as a MDVSP. The algorithm was first tested by benchmark instances consisting of 2, 3 and 5 depots, and a service load ranging from 100 to 500. The results obtained show that the proposed algorithm was capable of finding the optimal solution in most cases when considering a time limit of 500 seconds. We then applied the methodology to solve a real-life scenario of a Colombian public transportation system involving 2 depots and 719 services, and the results obtained imply in a decrease of the required fleet size and a balanced allocation of services, thus reducing deadhead trips.

The remainder of the paper is organized as follows. Section 2 presents a literature on the MDVSP, Section 3 formally describes the problem. Section 4 includes the proposed methodology. Section 5 contains the results of the computational experiments. Section 6 presents the concluding remarks.

2. Literature Review

In this section we review the recent MDVSP literature, briefly describing the corresponding solution approaches. Huisman et al. (2004) proposed a dynamic model to solve the VSP. The approach addressed by the authors consists of solving a set of optimization problems in a sequential manner, taking into account different scenarios in future travel times. During the first phase, trips are assigned to the different depots (clustering), solving the static problem. Next, in the second stage, a simple VSP is solved dynamically. The proposed methodology was evaluated in a bus operator company in the Netherlands. The data set consists of 1,104 trips and four depots. Gintner et al. (2005), considered the MDVSP with multiple types of vehicles. The authors proposed a two-phase method that provides near-optimal solutions. The mathematical formulation of the problem is based on a space-time network and a vehicle is allowed to start from one depot and return to a different one, aiming at minimizing deadhead and stop times. In practical cases, the number of trips is over a thousand, which is the reason the authors combined the model of a space-time network with a heuristic approach, in order to solve large problems and add new practical considerations. Hadjar et al. (2006) proposed a Branch-and-Bound algorithm to solve the MDVSP, which combines Column Generation (CG), Fixed Variables and Cutting Planes. The authors studied two mathematical formulations that are based on CG schemes to solve the Lagrange relaxation of the Linear Programming problem. The algorithm was tested on randomly generated cases as well as benchmark instances. In addition, they also applied their algorithm on a set of real world instances that were derived from data belonging to the Montreal Transport Society (MTS). The MTS operates a network that includes seven depots, 665 bus lines with 380 completion points and 17,037 trips.

In the study conducted by Wang and Shen (2007), a new version of the problem – denoted as VSP with Route and Fueling Time Constraints (VSPRFTC) – was presented in which they consider electric buses. This implies taking into account two new constraints: the duration of the route and the time of vehicle recharging. The authors propose a new mathematical formulation and use of the ant colony algorithm as a solution method. Laurent and Hao (2009) proposed an iterated local search (ILS) algorithm to solve the MDVSP. The authors developed a new neighborhood operator called block movement (Block Moves). The methodology uses a so-called auction algorithm to generate the initial vehicle scheduling. It then integrates a Two-Step Perturbation mechanism as a diversification procedure. The developed approach was tested on a set of 30 benchmark MDVSP instances. Hassold and Ceder (2014) presented a methodology based on a minimum cost network flow model, where the authors dealt with the heterogeneous fleet VSP (MVT-VSP) to solve a real case in New Zealand and the results showed an improvement of 15%, in terms of the cost of the vehicle fleet.

By means of a heuristic framework that makes use of a space-time network, the work presented by Guedes and Borenstein (2015) addressed the heterogeneous fleet MDVSP using truncated column generation and reduction of the state space. The results obtained were promising and constitute a feasible alternative to efficiently solve the problem. Shui et al. (2015) put forward a cloning algorithm and two heuristics of travel time readjustments. The approach obtained small CPU times and the ability to solve large instances such as the operation of buses in Nanjing, China. Recently, Wen et al. (2016), solved the VSP involving electric buses denoted as Electric VSP (E-VSP), whose main constraint lies in the buses driving ranges associated with battery recharging, which can be fully or partially recharged. The mathematical formulation proposed for the E-VSP involves, in the first step, minimizing the number of vehicles needed to carry out all scheduled trips (Timetable) and, secondly, minimizing the traveled distance, which is equivalent to minimizing deadhead trips. The problem was formulated as a mixed integer programming problem and was solved using an Adaptive Large Neighborhood Search (ALNS) heuristic. The solution methodology was capable of finding good quality solutions for large size problems and near-optimal solutions in small cases.

Schöbel (2017) presented an integrated approach in the context of planning process of public transportation system. The author argues that, instead of optimizing each stage of transportation systems, it would be more beneficial to consider the entire process in an integrated fashion. To this end, a model that jointly covers route planning, Timetables and VSP was proposed as well as an iterative heuristic algorithm. The results obtained were promising and the author listed a series of challenges for future studies regarding the integrated point of view of optimizing public transportation system planning.

3. Problem description and formulation

In the context of public transportation, the MDVSP can be represented through a set T_1, T_2, \dots, T_n of trip timetables that must be performed in a time horizon τ . Each trip T_i is characterized by the time and place of origin, as well as the destination. Let D_1, D_2, \dots, D_k be the set of depots and d_l the number of vehicles located at depot D_l .

A pair of trips (T_i, T_j) is compatible, if trip T_j can be executed immediately after trip T_i by the same vehicle. This is possible if the time of trip completion T_i plus the time of empty travel (deadhead) from T_i to T_j (added with a sufficient safety margin), is less than the start time of trip T_j . The objective of MDVSP is thus to connect a subset of trips to be performed by a vehicle, starting and ending at the same depot in such a way that the sum of the total travel costs is minimized. The formulation proposed by Mesquita and Paixão (1992) can be considered a suitable alternative in our case, mainly because it contains a polynomial number of variables and constraints. Let $G = (V, A)$ be a directed graph where a set of vertices $V = \{1, 2, \dots, n + k\}$ is partitioned into $I = \{1, 2, \dots, n\}$, which corresponds to trips T_1, T_2, \dots, T_n , and a set $D = \{n + 1, n + 2, \dots, n + k\}$, which refers to depots D_1, D_2, \dots, D_k . Set A

contains the arcs (i, j) representing the feasible trip pairs (T_i, T_j) and all the arcs leaving and arriving the depots $(i, n + 1), (n + 1, i)$, for $l = 1, 2, \dots, k$ and $i = 1, 2, \dots, n$. A cost c_{ij} is associated to each arc, which estimates the fuel consumption necessary for the deadhead trip from i to j , and other penalties that the transportation company intends to apply. The costs associated to the arcs from or to a depot are denoted as: $c_{n+1, n+1} = 0$, $l = 1, 2, \dots, k$ and $c_{n+l, n+l'} = \infty$, $l, l' = 1, 2, \dots, k$, $l \neq l'$.

In the particular case in which transportation companies seek to simultaneously minimize the size of the fleet and the costs of deadhead trips, then $c_{n+1, i}(c_{i, n+1})$ represents the cost of the deadhead trip from depot D_l to the starting point of the trip T_i (from the place of completion of trip T_i to depot D_l), plus half the cost incurred by the use of the vehicle belonging to depot D_l .

The MDVSP aims at finding the set of least-cost circuits in such a way that: (i) Each vertex $i \in I$ is covered exactly once by a circuit, (ii) Each circuit contains exactly one vertex of set D and (iii) The number of circuits that covers vertex $n + 1$ must never exceed $d_l = (l = 1, 2, \dots, k)$.

3.1 Definition of decision variables

The decision variables of the mathematical model proposed by Mesquita and Paixão (1992) are specified as follows:

$$x_{ij} = \begin{cases} 1, & \text{if trip } i \text{ immediately proceeds trip } j \\ 0, & \text{otherwise} \end{cases}$$

$$x_{i, n+l} = \begin{cases} 1, & \text{if after trip } i \text{ the bus returns to depot } D_l \\ 0, & \text{otherwise} \end{cases}$$

$$x_{n+l, j} = \begin{cases} 1, & \text{if a bus starts a trip } j \text{ from depot } D_l \\ 0, & \text{otherwise} \end{cases}$$

$$y_{i, l} = \begin{cases} 1, & \text{if trip } i \text{ is carried out by a vehicle from depot } D_l \\ 0, & \text{otherwise} \end{cases}$$

3.2 Mathematical formulation

$$\min \sum_{i=1}^n \sum_{j=i}^n c_{ij} x_{ij} + \sum_{l=1}^k \left(\sum_{i=1}^n c_{i, n+l} x_{i, n+l} + \sum_{j=1}^n c_{n+l, j} x_{n+l, j} \right) x_{n+l, j} \quad (1)$$

subject to

$$\sum_{j=1}^{n+k} x_{ij} = 1 \quad i = 1, 2, \dots, n \quad (2)$$

$$\sum_{j=1}^{n+k} x_{ij} = 1 \quad i = 1, 2, \dots, n \quad (3)$$

$$\sum_{j=1}^n x_{n+l, j} \leq d_l \quad l = 1, 2, \dots, k \quad (4)$$

$$x_{n+l, j} - y_{j, l} \leq 0 \quad j = 1, 2, \dots, n; l = 1, 2, \dots, k \quad (5)$$

$$x_{i, n+l} - y_{i, l} \leq 0 \quad i = 1, 2, \dots, n; l = 1, 2, \dots, k \quad (6)$$

$$y_{i, l} + x_{ij} - y_{j, l} \leq 1 \quad i, j = 1, 2, \dots, n; l = 1, 2, \dots, k \quad (7)$$

$$\sum_{l=1}^k y_{i, l} = 1 \quad i = 1, 2, \dots, n \quad (8)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 1, 2, \dots, n; (i, j) \neq (n + l, n + l'); l, l' = 1, 2, \dots, k \quad (9)$$

$$y_{il} \in \{0,1\}, \quad i = 1,2, \dots, n; l = 1,2, \dots, k \quad (10)$$

$$\sum_{l=1}^k \sum_{j=1}^n x_{n+l,j} \leq UB_{Vehicle} \quad (11)$$

The meaning of the constraints is described as follows:

- Constraints (2) ensure that, for each $i = 1,2, \dots, n$; when finishing trip i , the bus must return to a depot or must start another trip $j \in I$.
- Constraints (3) assign, for each $j = 1,2, \dots, n$, a bus to a trip j , either from the depot or from another trip.
- Constraints (4) enforce that, for $l = 1,2, \dots, k$ the capacity d_l of a depot must not be violated.
- Constraints (5) impose, for each pair j and l , the assignment between trip j and depot D_l , as long as trip i is the first trip of an itinerary done by a vehicle that starts its service from depot D_l .
- Constraints (6) ensure that, for each pair i and l , the assignment between trip i and depot D_l , as long as trip i is the last trip of an itinerary done by a vehicle that returns to depot D_l .
- Constraints (7) establish that for each pair $(i, j) \in I$, if a trip i is connected directly with trip j and trip i is assigned to depot D_l , then, trip j is assigned to depot D_l .
- Constraints (8) guarantee that each trip $i \in I$ is assigned exactly to one depot.
- Constraints (9) and (10) define the domain of the decision variables.
- Constraint (11) impose an upper bound on the maximum number of vehicles that is obtained by a simple constructive procedure as described in Section 4.2.

4. Proposed methodology

This section describes the proposed hybrid algorithm that combines (i) a GA procedure; (ii) a SP approach; and (iii) a MIP formulation. The GA metaheuristic is based on the methodology presented by Chu and Beasley (1998), whose initial pool of itineraries is built using different constructive procedures. The best combination of the itineraries generated while executing the GA is then obtained by the SP approach which in turn possibly returns an improved solution. Such solution is then provided as a starting point to a MIP solver which aims at finding an even better solution using the formulation proposed by Mesquita and Paixão (1992). The algorithm will be referred to as **GA+SP+MIP** and its basic description can be seen in Fig. 1.

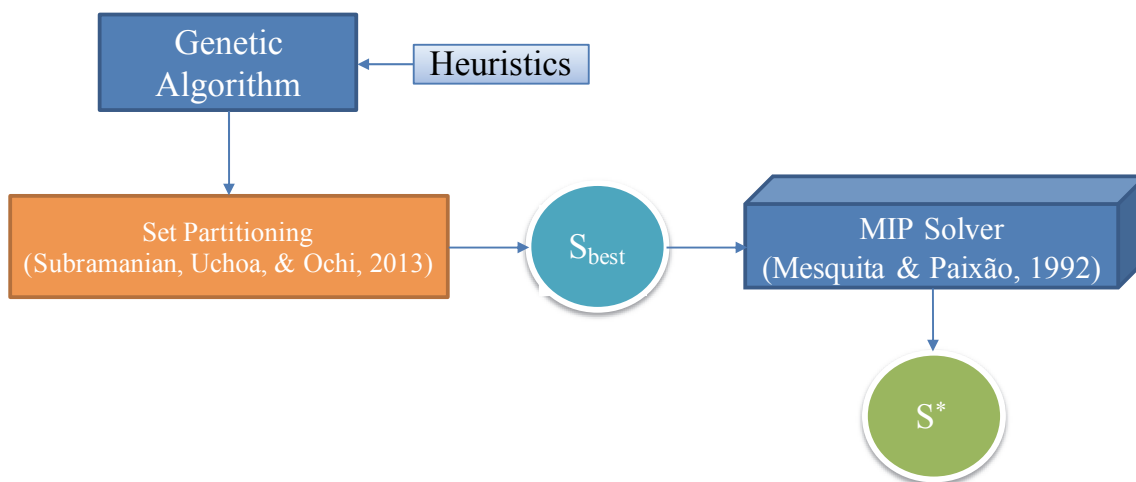


Fig. 1. Methodology.

4.1 Constructive Procedures

The three constructive algorithms implemented to generate initial solutions are described as follows.

4.1.1 Clustered Concurrent Scheduler (CCS)

In this procedure, we apply the traditional Concurrent Scheduler method (Dell'Amico et al., 1993) adding a first clustering stage, that is, the algorithm starts by determining from which depot D_l each of the trips T_i should be attended. To this end, each trip is assigned in a heuristic fashion taking into account the lowest value $c_{n+l,i}$, as shown in Fig. 2.

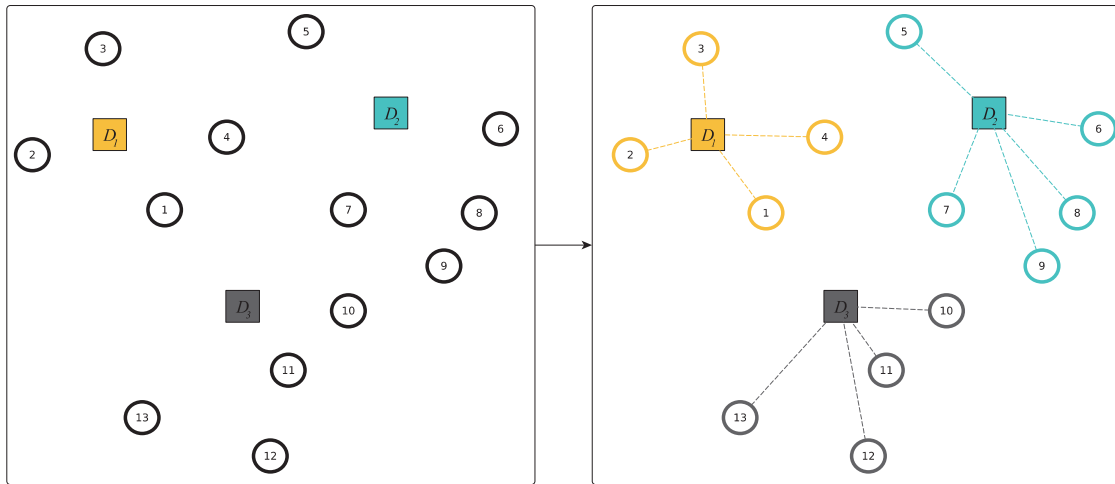


Fig. 2. Clustering of trips for each depot D_l .

The next step is to sort the trips assigned to each depot in ascending order with respect to the start time. Each itinerary is then generated by only allowing feasible assignments. When the first infeasibility occurs, the itinerary is completed and assigned to a vehicle of the corresponding depot. The procedure is repeated until there are no trips to be assigned to each of the clusters (Figure 3).

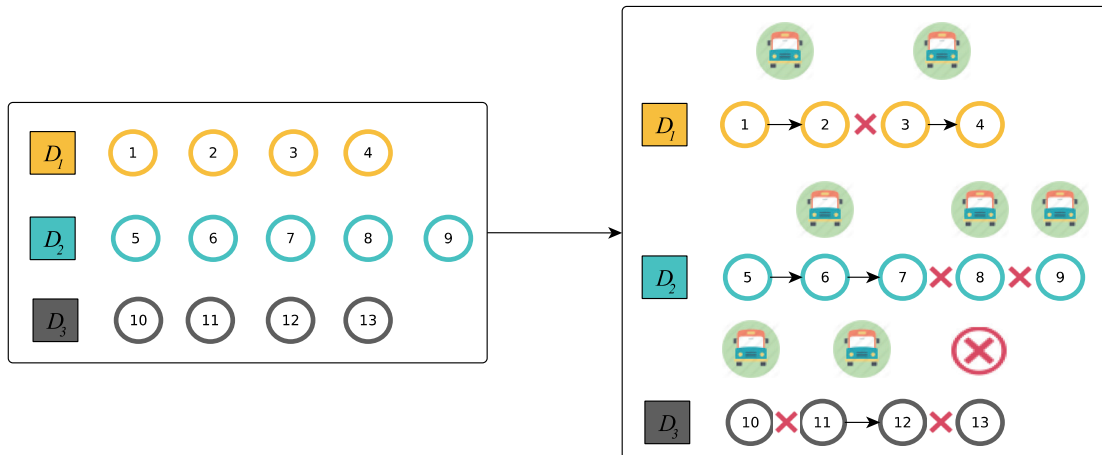


Fig. 3. Construction of itineraries for each depot's D_l fleet

In case the least-cost arc that connects a given depot to service T_i runs out of vehicles, the procedure attempts to reassign such service to the next depot also using the least-cost arc criterion, as shown in Fig. 4. The algorithm terminates when all trips have been assigned.

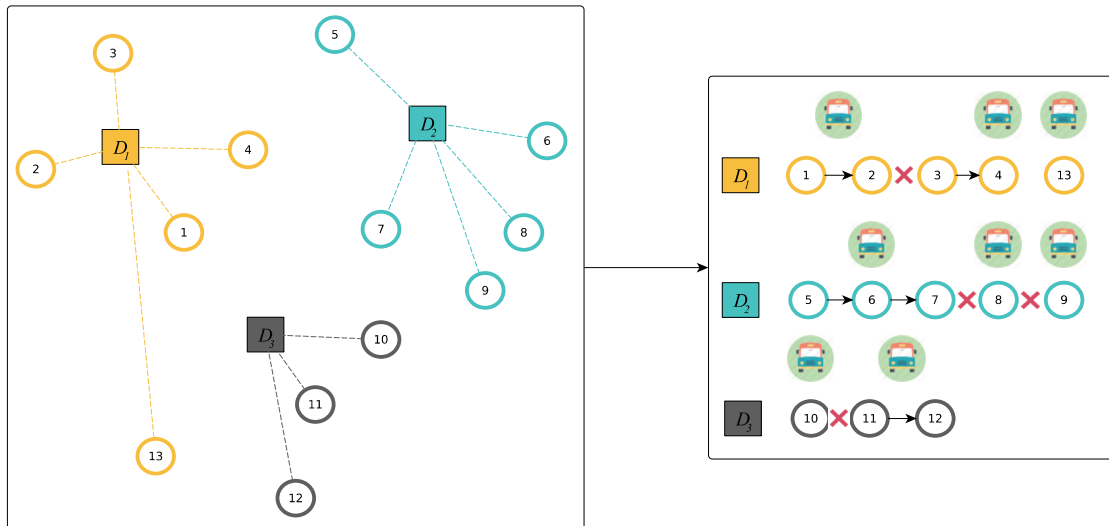


Fig. 4. Reassignment of trip 13 for fleet availability

4.1.2 Minimal Cost Attention Sequence (MCAS)

The method is based on the construction of a general sequence with all trips T_i , that is, only the nodes of set I are taken into account. The first trip in the sequence corresponds to the one with smaller start time. The following trips in the sequence are assigned according to the feasible nearest neighbor criterion. In addition, the terminal node must have at least degree 2 (Jungnickel, 2007). When there are no more feasible assignments or the minimum degree requirement is not met, the procedure sets the subsequence as an itinerary. The construction of the general sequence continues with the unassigned trips, taking as a junction trip the one which has the highest degree of feasible departure. The previous procedure is repeated until all trips are assigned to the general sequence, as illustrated in Figure 5.

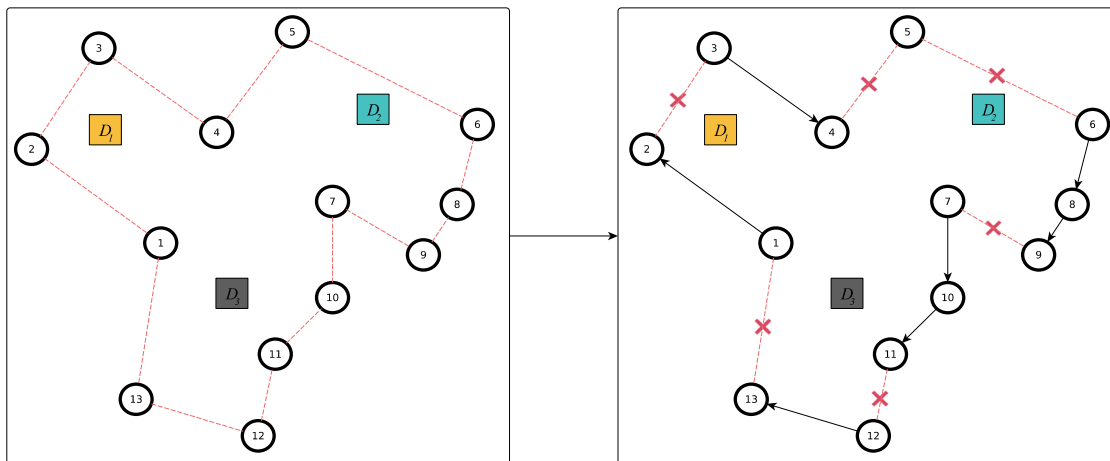


Fig. 5. Sequence generated connecting routes using the nearest neighbor criterion

Finally, the subsequences become supernodes (Figure 6) and the Generalized Assignment Problem (GAP) is solved in order to relate it to the set of D depots, and to have a complete MDVSP solution.

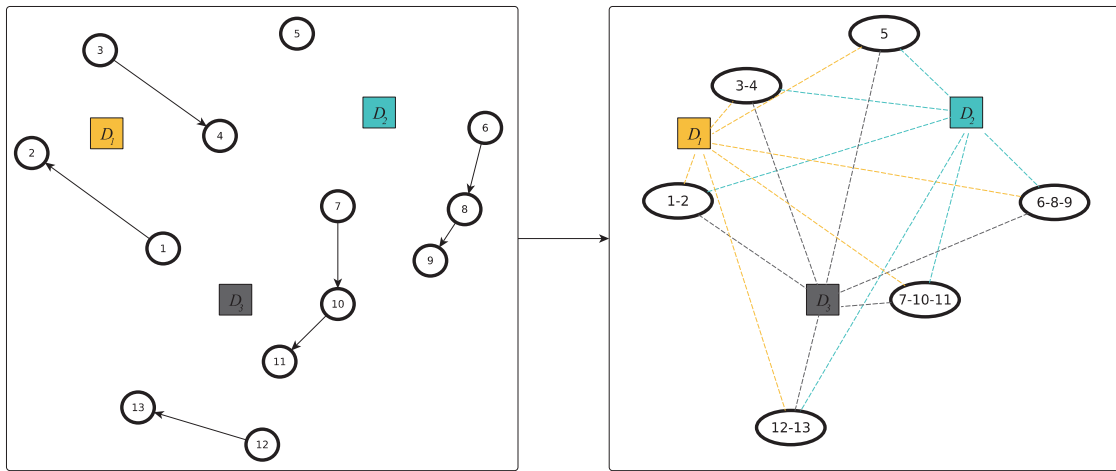
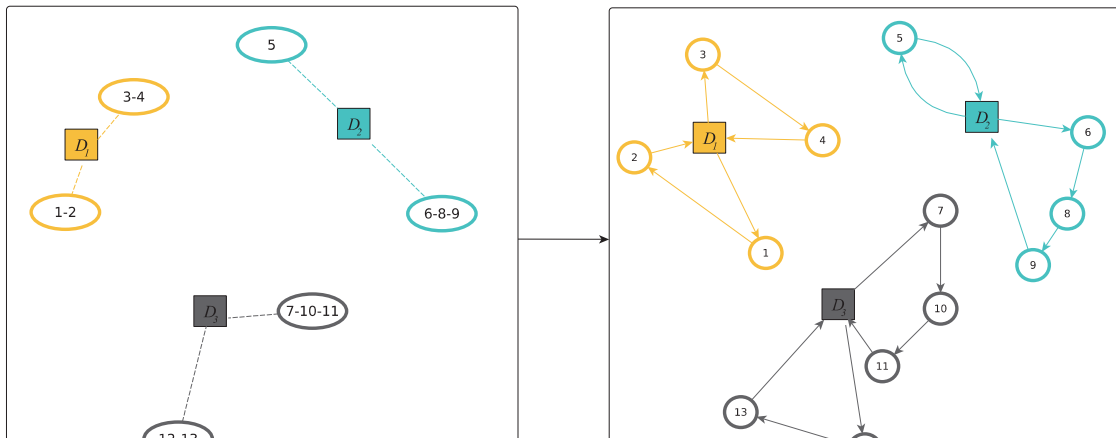


Fig. 6. Supernodes corresponding to travel itineraries (GAP)

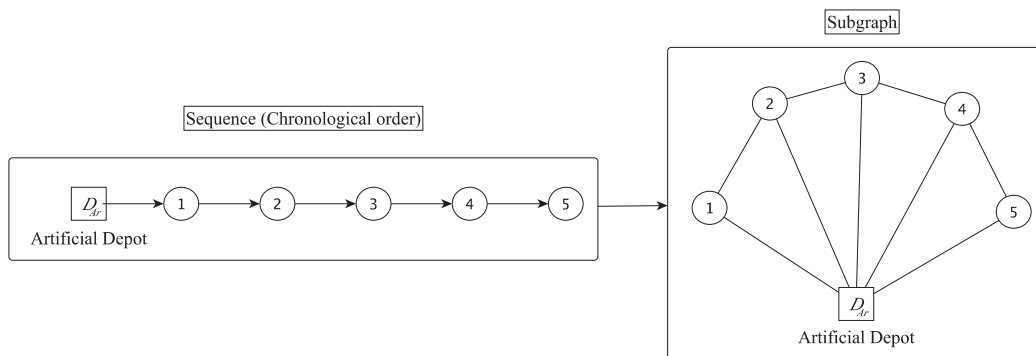
Fig. 7 shows the GAP solution and the corresponding node decoding.



4.1.3

This and trips

:004) of all

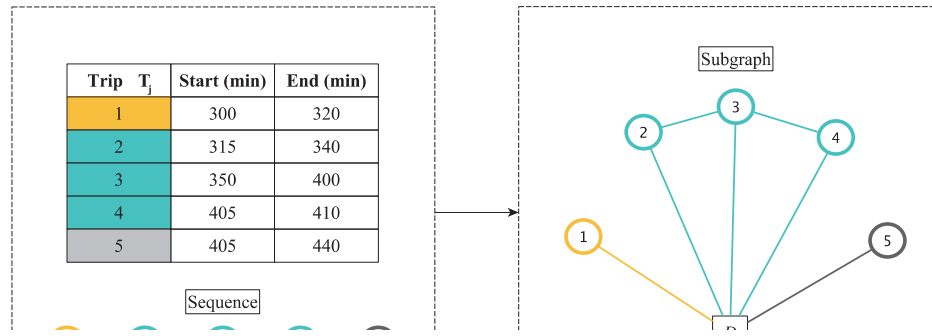


Fig

uph

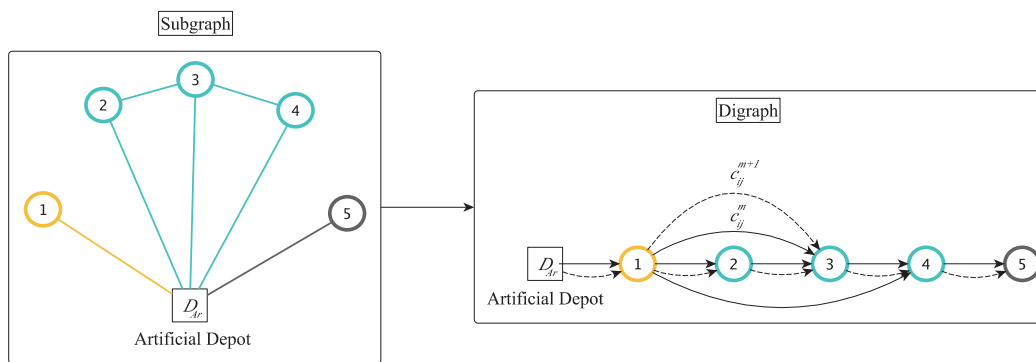
From one (9).

ation (Fig.



Beca depo digra math

er of 1 the used



We 1 benc (Fig.

f the 2

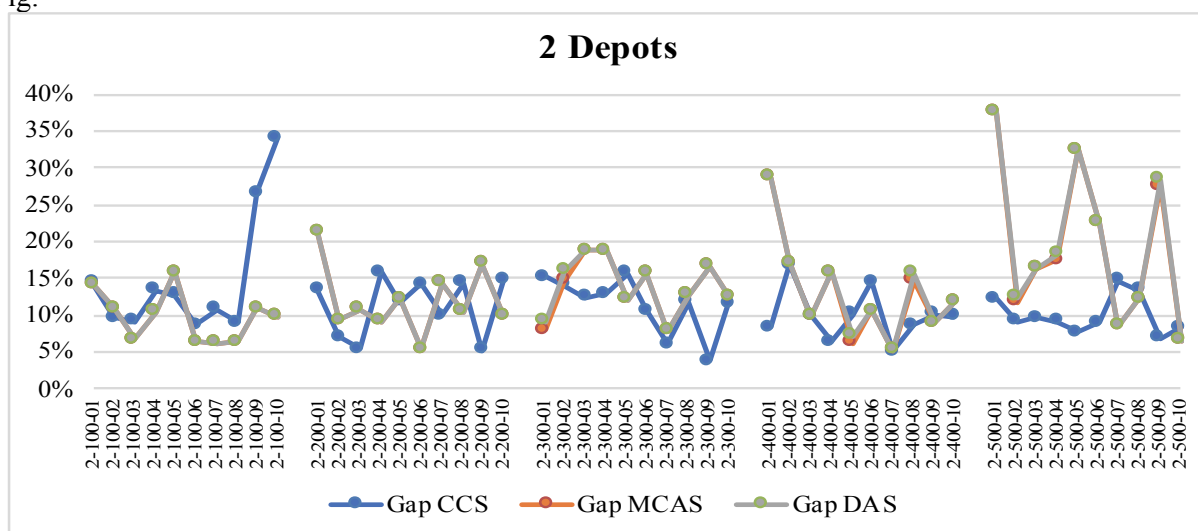


Fig. 11. Gap constructive procedures vs. optimal solution (2 Depots)

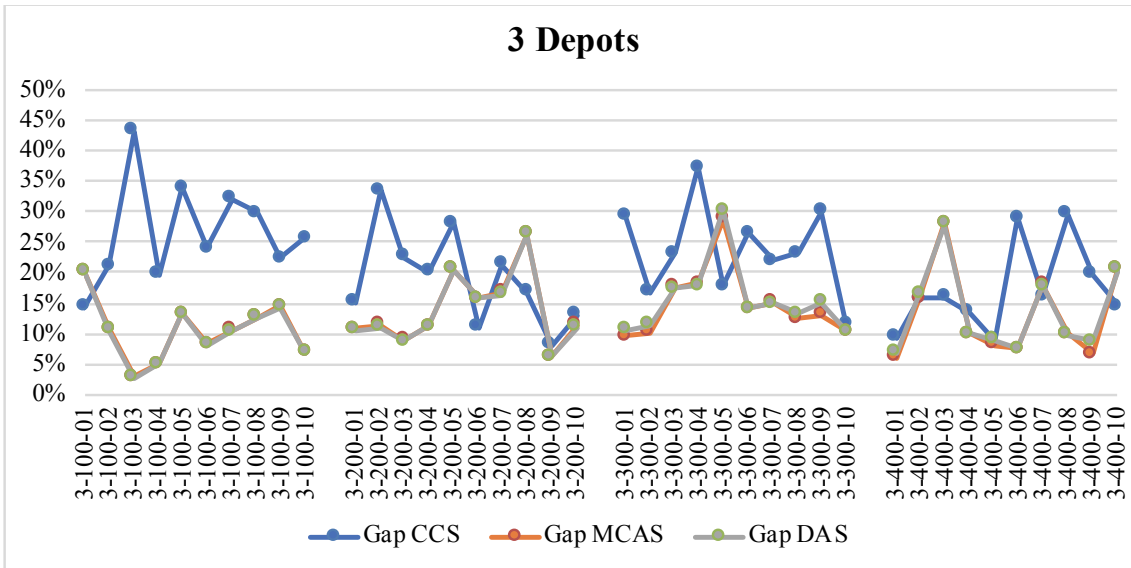


Fig. 12. Gap constructive procedures vs. optimal solution (3 Depots)

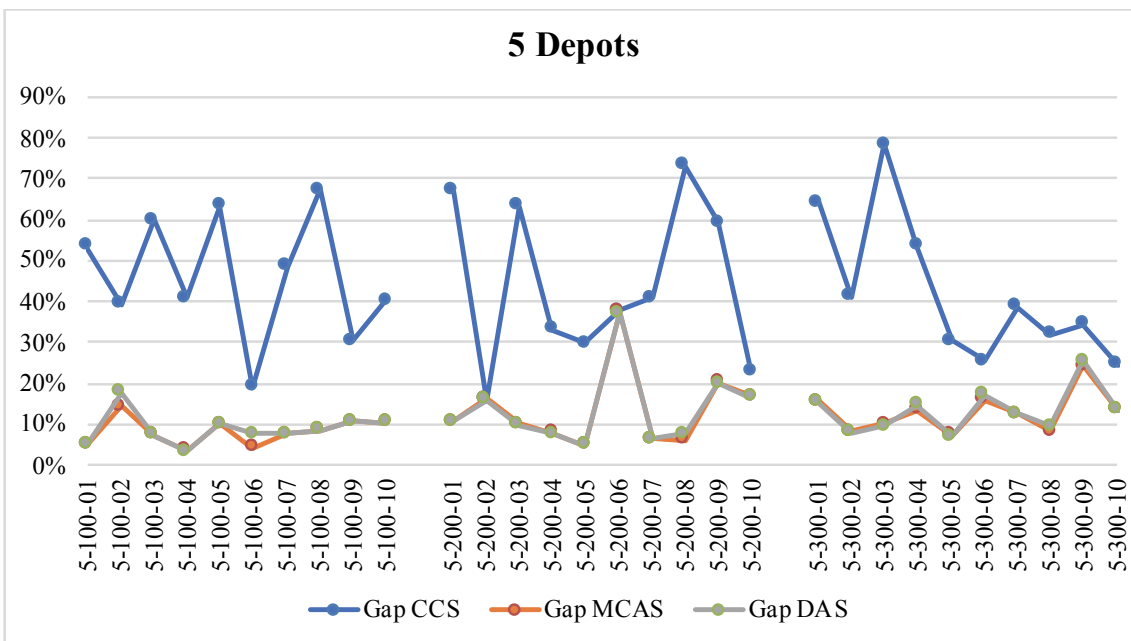


Fig. 13. Gap constructive procedures vs. optimal solution (5 Depots)

It is important to mention that during the execution of the tests, it was observed that the constructive procedures did not appear to dominate each other for the distinct alternatives. In fact, they are complementary to obtain good quality results when considering the different cases of the existing benchmark instances.

4.2 Genetic Algorithm (GA)

After building several initial solutions with the constructive procedures, a population-based metaheuristic could be an interesting strategy to benefit from the generated schedules. Once the initial population is created (Fig. 14), one applies a procedure that seeks to quickly obtain an UB from the fleet required to meet the services. This procedure simply consists in sorting all the services T_i in ascending order according to their start time, and constructing itineraries consecutively with this sequence.

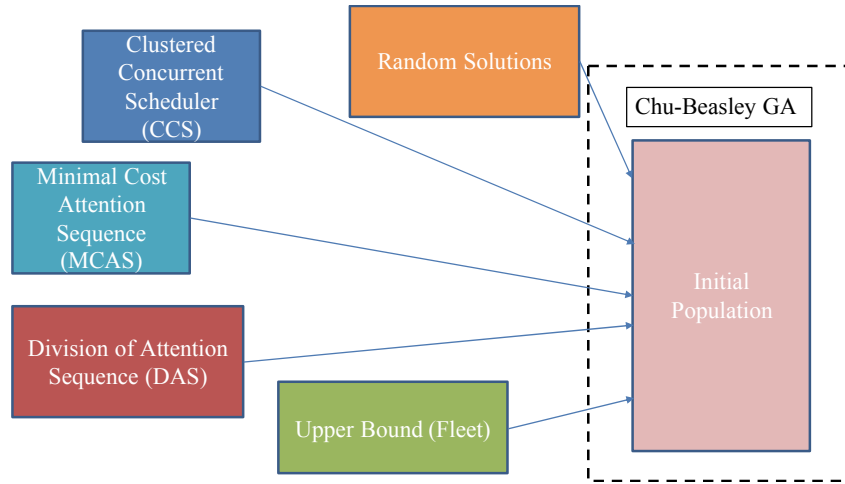


Fig. 14. Generating the initial population.

In our case, we chose to implement a Chu-Beasley-based Genetic Algorithm as a population-based metaheuristic. An elitist acceptance criterion is used, and when it is possible to update the population, only the worst solution is modified.

The coding of the solutions that represent the population is composed of two vectors. The first corresponds to the order in which the services are added to the itineraries, and the second indicates the vehicles to which the services of the first vector are assigned, as proposed in Park (2001). The criterion adopted for assigning trips to itineraries is improved using a trip insertion operator to each of the itineraries under construction.

Coding the solutions in vectors facilitates the application of recombinations. In this work, the five forms of recombination presented in Leung et al. (2001) are used.

4.3 Set Partitioning (SP)

After the GA terminates, a SP-based procedure is applied as an attempt to further improve the incumbent solution. In this phase, a solution is built from the best itineraries of the final population obtained by the GA. The strategy consists of partitioning each solution (each one is a set of itineraries) and performing the optimal reconstruction of a solution for the MDVSP. In Subramanian et al. (2013) a compact model of SP is proposed, oriented to vehicle routing problems with a homogeneous fleet. In this work, such model is applied to the MDVSP as described next. Let R be the set of all possible itineraries assigned to the different depots and $R_i \subseteq R$ the set of itineraries that contains the trip or service. Define y_j as the binary variable associated with the itinerary $j \in R$ with associated cost c_j . Moreover, let $R_l \subseteq R$ be the set of itineraries assigned to depot $l \in D$ and let d_l be the number of buses or available vehicles at depot l . The formulation can be written as follows:

$$\min \sum_{j \in R} c_j y_j \quad (12)$$

subject to

$$\sum_{j \in R_i} y_j = 1 \quad \forall i \in I \quad (13)$$

$$\sum_{j \in R_l} y_j \leq d_l \quad \forall l \in D \quad (14)$$

$$y_j \in \{0,1\}, \quad \forall j \in R \quad (15)$$

Objective function (12) minimizes the sum of the costs of choosing the best combination of itineraries. Constraints (13) guarantee that service i is served exactly by one itinerary j . Constraints (14) ensure that the selected itineraries do not exceed the maximum number of vehicles available at each depot. Enumerating all elements of set R would result in a prohibitively large number of itineraries. Therefore, we decided to limit the size of R by only considering the set of solutions associated with the final population obtained by the GA procedure. We will thus obtain a compact model that will contain about thousands of itineraries for the largest cases. As a result, the model can be easily solved with the well-known MIP solver CPLEX® Version 12.7.

4.4 Initial primal bound and MIP-based solution

The best solution found during the SP procedure is provided as an initial primal solution to the the MIP model proposed by Mesquita and Paixão (1992), which in turn is the last step of our methodology. It was observed that providing an initial solution substantially helped improving the efficiency of the model. However, as it still may be hard to determine the optimal solution, especially for large-size instances, a time limit was imposed to avoid prohibitively long runs.

5. Computational Results

The proposed algorithm was implemented in C++® running Ubuntu Linux 16.04.03 of 64-bits LTS operating system and CPLEX® 12.7 as a MIP solver. The experiments were performed on an Intel® Core™ i5-4570 CPU of 3.20 GHz \times 4, and 8 GB of RAM. A time limit of 500 seconds was imposed to the MIP model presented by Mesquita and Paixão (1992). In order to evaluate the performance of the developed approach, we made use of the benchmark instances introduced by Fischetti et al. (1999). These test cases are composed of 10 instances for each set. The sets combine 100, 200, 300, 400 and 500 services with 2 depots. For 3 and 5 depots they handle a maximum of 400 and 300 services, respectively. In Tables 1, 2 and 3, the 10 instances of each benchmark set are grouped and the corresponding averages are presented for each set (OPT, optimal value of the objective function). Also, we compare the average number of vehicles (NV) as well as the CPU times in seconds (T (s)) and the Gaps found by the different stages of the proposed methodology.

Table 1

Consolidated (average) of instances groups with 2 depots

Instances	Fischetti et al. (1999)			GA+SP+MIP			NV	T(s)
	OPT	NV	TS	Gap GA	Gap SP	Gap MIP		
2-100	309258,4	27,9	0,6	2,46%	1,21%	0,00%	27,9	2,345
2-200	606028,1	54,8	23,3	3,12%	1,13%	0,00%	54,8	30,014
2-300	833318,7	75,5	84,16	3,35%	1,77%	0,00%	75,5	130,189
2-400	1115927,9	101,4	297,94	3,11%	1,98%	0,00%	101,4	361,886
2-500	1373243,4	125,1	1160,73	3,94%	1,89%	0,19%	125,1	481,911

Table 2

Aggregate results for the benchmark instances with 3 depots

Instances	Fischetti et al. (1999)			GA+SP+MIP			NV	T(s)
	OPT	NV	TS	Gap GA	Gap SP	Gap MIP		
3-100	309642,3	28,2	2,8	2,65%	1,19%	0,00%	28	39,051
3-200	584726	53,6	72,74	3,49%	1,36%	0,01%	53,6	233,718
3-300	831986,7	76,5	339,84	3,91%	1,99%	0,02%	76,5	472,109
3-400	1084336,6	99,9	3177,63	3,54%	2,13%	0,40%	99,9	847,763

Table 3

Aggregate results for the benchmark instances with 5 depots

Instances	Fischetti et al. (1999)			GA+SP+MIP			NV	T(s)
	OPT	NV	TS	Gap GA	Gap SP	Gap MIP		
5-100	320728,9	29,5	22,49	2,95%	1,40%	0,00%	29,5	249,597
5-200	587420,1	54,3	341,43	3,66%	1,76%	0,08%	54,3	461,199
5-300	827447	76,8	3130,49	4,28%	2,20%	0,61%	76,8	499,16

In addition to achieving optimality in most of the test cases, the proposed algorithm also finds the optimal number of vehicles required in all instances, thus showing the efficiency of our approach for the benchmark instances considered. We therefore believe that the algorithm can be extended to a real life instance in the context of mass public transportation.

5.1 Application in a Real Case

The results for the real world scenario obtained by both the constructive procedures and the GA+SP+MIP algorithm are shown in Table 4 and Fig. 15(a) and Fig. 15(b). The real test case derived from operations that arise in a business day is composed of two depots and 719 services that must be served by the available fleet of 37 vehicles (available at <https://www.researchgate.net/publication/327288601> Study Case MDVSP Integra SA Pereira Colombia).

Table 4
Comparative table constructives real case, business day

Real Case 719 Trips – 2 Depots	NV			
	Manual Scheduling	CCS	MCAS	DAS
	36	66	35	66

As illustrated in Table 4, by only applying the constructive procedures, we were capable of obtaining a reduction of one vehicle on the fleet necessary to meet all services on a business day. In later stages, a decrease in the number of vehicles was not achieved, given that the demand is very high. However, the GA+SP+MIP algorithm decreases deadhead trips, finding a uniform distribution on the allocation of services to buses. In Fig. 15 and Fig. 16, where the vehicle scheduling is presented using 35 vehicles; the x-axis represents the time of the day, the y-axis contains each schedule, and the numbers in the lines correspond to the “ids” of the trips or services assigned to the vehicles. Each service has a starting time and a completion time, which are the points between the ids. The deadhead trips or idle times for the vehicles are highlighted in green, showing how the methodology achieves a more balanced solution (with cost 16208, Fig. 16) when compared with the initial solution in Fig. 15 (with cost 20240).

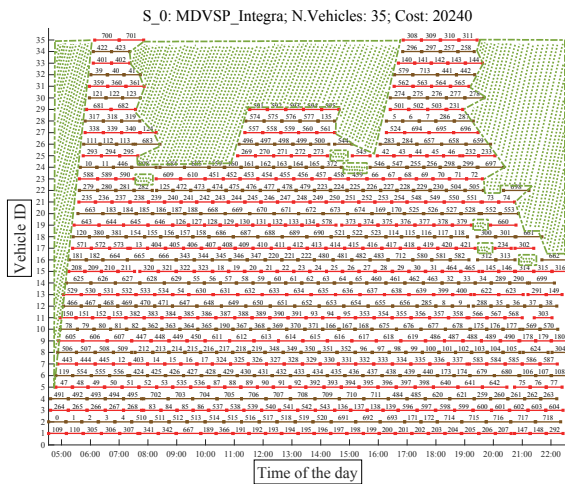


Fig. 15. Initial solution

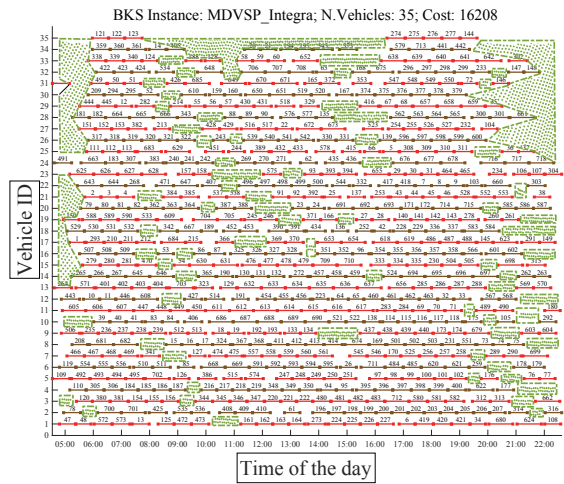


Fig. 16. Algorithm (GA+SP+MIP) impact in the resulting assignment

Therefore, it is possible to conclude that the minimum number of vehicles were used and that the deadhead trips were reduced significantly.

6. Concluding Remarks

In this paper, we proposed a hybrid methodology that involves constructive algorithms, a metaheuristic, a Set Partitioning approach and a MIP-based procedure. Such methodology is able to solve test cases of considerable size both in the benchmark instances and in a real case, within reasonable CPU times. The

MDVSP approach, combined with techniques commonly used in vehicle routing problems, presents an adequate development and offers an acceptable scalability when the number of depots and trips increase. Future work may include the development of parallel algorithms so as to mitigate the excessive CPU times that may arise in complex single-thread algorithms, such as the one described in this paper.

Acknowledgments

The authors thank the support of Integra S.A., COLCIENCIAS, SENA, the Master Program in Electrical Engineering and the Doctoral Program in Engineering of the Universidad Tecnológica de Pereira.

References

- Ceder, A. (2007). *Public transit planning and operation: Modeling, practice and behavior*. CRC press.
- Chu, P. C., & Beasley, J. E. (1998). A genetic algorithm for the multidimensional knapsack problem. *Journal of heuristics*, 4(1), 63-86.
- Dell'Amico, M., Fischetti, M., & Toth, P. (1993). Heuristic algorithms for the multiple depot vehicle scheduling problem. *Management Science*, 39(1), 115-125.
- Fischetti, M., Lodi, A., & Toth, P. (1999). A branch-and-cut algorithm for the multiple depot vehicle scheduling problem. *Dipartimento di Elettronica e Informatica, Università di Padova, Italy*.
- Gintner, V., Kliewer, N., & Suhl, L. (2005). Solving large multiple-depot multiple-vehicle-type bus scheduling problems in practice. *OR Spectrum*, 27(4), 507-523.
- Guedes, P. C., & Borenstein, D. (2015). Column generation based heuristic framework for the multiple-depot vehicle type scheduling problem. *Computers & Industrial Engineering*, 90, 361-370.
- Hadjar, A., Marcotte, O., & Soumis, F. (2006). A branch-and-cut algorithm for the multiple depot vehicle scheduling problem. *Operations Research*, 54(1), 130-149.
- Hassold, S., & Ceder, A. A. (2014). Public transport vehicle scheduling featuring multiple vehicle types. *Transportation Research Part B: Methodological*, 67, 129-143.
- Huisman, D., Freling, R., & Wagelmans, A. P. (2004). A robust solution approach to the dynamic vehicle scheduling problem. *Transportation Science*, 38(4), 447-458.
- 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.
- Jungnickel, D. (2007). *Graphs, networks and algorithms* (Vol. 5). Springer Science & Business Media.
- Laurent, B., & Hao, J. K. (2009). Iterated local search for the multiple depot vehicle scheduling problem. *Computers & Industrial Engineering*, 57(1), 277-286.
- Leung, T. W., Yung, C. H., & Troutt, M. D. (2001). Applications of genetic search and simulated annealing to the two-dimensional non-guillotine cutting stock problem. *Computers & industrial engineering*, 40(3), 201-214.
- Liu, S., Huang, W., & Ma, H. (2009). An effective genetic algorithm for the fleet size and mix vehicle routing problems. *Transportation Research Part E: Logistics and Transportation Review*, 45(3), 434-445.
- Mesquita, M., & Paixão, J. (1992). Multiple depot vehicle scheduling problem: A new heuristic based on quasi-assignment algorithms. In *Computer-Aided Transit Scheduling* (pp. 167-180). Springer, Berlin, Heidelberg.
- Munoz, J. C., & Paget-Seekins, L. (Eds.). (2016). *Restructuring public transport through Bus Rapid Transit: An international and interdisciplinary perspective*. Policy Press.
- Park, Y. B. (2001). A hybrid genetic algorithm for the vehicle scheduling problem with due times and time deadlines. *International Journal of Production Economics*, 73(2), 175-188.
- Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, 31(12), 1985-2002.
- Schöbel, A. (2017). An eigenmodel for iterative line planning, timetabling and vehicle scheduling in public transportation. *Transportation Research Part C: Emerging Technologies*, 74, 348-365.
- Shui, X., Zuo, X., Chen, C., & Smith, A. E. (2015). A clonal selection algorithm for urban bus vehicle scheduling. *Applied Soft Computing*, 36, 36-44.
- Subramanian, A., Uchoa, E., & Ochi, L. S. (2013). A hybrid algorithm for a class of vehicle routing problems. *Computers & Operations Research*, 40(10), 2519-2531.
- Wang, H., & Shen, J. (2007). Heuristic approaches for solving transit vehicle scheduling problem with route and fueling time constraints. *Applied Mathematics and Computation*, 190(2), 1237-1249.
- Wen, M., Linde, E., Ropke, S., Mirchandani, P., & Larsen, A. (2016). An adaptive large neighborhood search heuristic for the electric vehicle scheduling problem. *Computers & Operations Research*, 76, 73-83.

