Multi-objective MDVRP solution considering route balance and cost using the ILS metaheuristic

Luis Fernando Galindres-Guancha[a], Eliana Mirledy Toro-Ocampo[b*] and Ramón Alfonso Gallego- Rendón[c]

[a]*Ph.D Student, Faculty of Engineering, Technological University of Pereira, Risaralda, Colombia*
[b]*Ph.D Professor, Faculty of Industrial Engineering, Technological University of Pereira, Risaralda, Colombia*
[c]*Ph.D Professor, Faculty of Electrical Engineering, Technological University of Pereira, Risaralda, Colombia*

C H R O N I C L E	A B S T R A C T
	The multi-objective problem of multi-depot vehicle routing (MOMDVRP) is proposed by considering the minimization of the traveled arc costs and the balance of routes. Seven mathematical models were reviewed to determine the route balance equation and the best-performing model is selected for this purpose. The solution methodology consists of three stages; in the first one, beginning solutions are built up by means of a constructive heuristic. In the second stage, fronts are constructed from each starting solution using the iterated local search multi-objective metaheuristics (ILSMO). In the third stage, we obtain a single front by using concepts of dominance, taking as a base the fronts of the previous stage. Thus, the first two fronts are taken and a single front is formed that corresponds to the current solution of the problem; next the third front is added to the current Pareto front of the problem, the procedure is repeated until exhaustion of the list of the fronts initially obtained. The resulting front is the solution to the problem. To validate the methodology we use instances from the specialized literature, which have been used for the multi-depot routing problem (MDVRP). The results obtained provide very good quality. Finally, decision criteria are used to select the most appropriate solution for the front, both from the point of view of the balance and the route cost.

1. Introduction

The Multi-Depot Vehicle Routing Problem (MDVRP) is a variant of the classical Vehicle Routing Problem (VRP), which consists of designing a set of routes with a set of clients which consume a determined demand. A fleet of vehicles attends each client´s demands with a capacity already defined from a depot.

The objective is to minimize the total distance traveled (Toth, 2014). The MDVRP considers several depots from which a set of vehicles attends a number of clients; once the tour is completed, they return to the same depot. Both the MDVRP and the VRP are NP-hard combinatorial problems (Cordeau et al., 1997; Ho et al., 2008).

Keskinturk and Yildirim (2011) propose that each driver´s workload is defined according to the length of each route, the volume carried during a time frame (including charging and discharging times) and the number of clients that need to be visited.

Schwarze and Voß (2013) propose six different types of objective functions related to the workload balancing, and they take into account three types of indicators included in the objective function. First, the route length, which refers to the distance, time or cost to carry out a route; second, the time invested in the discharging operation in each client and third, the demand of each client that implies the transportation volume.

To solve the MDVRP, several exact and metaheuristic techniques have been suggested. In the case of the exact-technique approach, the MDVRP is formulated as a Mixed-Integer Linear Programming (MILP) problem, as described by Kulkarni and Bhave (1985) and Montoya et al. (2015). However, these techniques converge into optimal solutions for small-size problems (less than 50 clients). On the other hand, the metaheuristic techniques have been widely used to solve efficiently both the mono-objective MDVRP and the Multi-Objective Multi-Depot Vehicle Routing Problem (MOMDVRP).

Regarding the MOMDVRP, very little has been researched as only about 12% of the papers reviewed address the MOMDVRP, and only about 4% take into account the workload balancing, as shown by Montoya et al. (2015).

Geiger (2008) proposes a concept denominated as Pareto Iterated Local Search (PILS), that combines intensification and diversification in one algorithm to generate a set of solutions traditionally called *population*, which starts from an initial solution x_1, from which an approximated Pareto set is obtained applying VNS iterative searches. From this set, the non-dominated solutions (Pareto front) are computed, from which a unique solution is selected x_2 and from which diversification is applied by using a perturbation operator obtaining x_3 as a perturbed solution. This procedure is performed to solve the Multi-Objective Flow Shop Scheduling problem.

The concept of Multi-Objective Local Search based on the dominance concept (DMLS) is explained by Liefooghe et al. (2012); besides, the following strategies are described in detail: dominance relation, selection of current set of solutions, neighborhood exploration and stopping criteria. The strategy is tested in two combinatorial optimization problems with several objectives: the Flow Shop Problem (FPS) and the Traveling Salesman Problem (TSP) from which, a DMLS model is proposed and a comparative study of different strategies for the DMLS to solve FSP and TSP variants is presented.

On the other hand, in Duarte et al. (2015), the VNS metaheuristic adaptation is explored along with its extensions to solve multi-objective combinatorial problems. To achieve the objective, the solution concept is redefined and adapted to the multi-objective context, where a set of solutions called approximated set of efficient solutions is taken. This new definition also allows redefining the meaning of improvement *i.e.* an improvement is given when a new solution is added to the approximated set of efficient solutions. Under these considerations, a procedure is developed for solving multi-objective combinatorial optimization problems, considering that this approach may require a high computational effort.

In both the MDVRP and the VRP, it is generally aimed to minimize the operation cost; that is, the total distance traveled by the vehicle fleet without exceeding its load capacity; however, there are other objectives that might be optimized such as the environmental and social impact. In this regard, the social matter is approached by balancing the drivers' workload through the minimization of the objective function, which is calculated from all the length-routes standard deviation. However, since the calculation of this objective function requires an important computational effort, it offers better results and no additional parameters in the objective function are required.

This paper proposes a new multi-objective methodology for solving the workload balance and cost in the MDVRP, where the metaheuristic is used based on the trajectory called Iterated Local Search (ILS) that

includes the Variable Neighborhood Search (VNS). The ILS is composed of two stages that keep the diversification and intensification in the search space. The diversification stage is performed through a perturbation mechanism that allows exploring promissory regions in the solution space. On the other hand, the intensification stage is implemented with the VNS, which consists of specialized operators responsible for reducing the solution space by making the search in nearby surroundings (neighborhoods) the current solution. In the present work, the VNS is implemented by using two types of operators: The Inter-route operators that look for a better solution between two routes and the Intra-route operators that look for a better solution into an only one route. Both operators are based on shift, swap and 2-opt strategies.

The proposed methodology includes multi-objective optimization, with which the approximated Pareto front is obtained, based on the non-dominated solutions generated by the ILS. The methodology is validated with instances from the literature taken from Cordeau et al. (1997). The results obtained are of good quality and allow concluding about the relationship between cost minimization and route balance, which is of interest for the academic community.

Finally, the rest of the article is organized as follows: Section 2 presents the model for the multi-objective multi-depot vehicle rout problem (MOMDVRP), where two objectives are defined: the solution cost and the standard deviation of the total distance traveled in each route. In Section 3, the new methodology proposed is described to solve the MOMDVRP using the ILS-VNS. In Section 4, the results are analyzed comparing them with some existing instances for the MDVRP. Section 5 presents the conclusions, considerations and guidelines for future works.

2. MOMDVRP Proposed Model

The MDVRP is an extension of the VRP that determines a set of routes traveled by specific vehicles. (i) every vehicle starts and ends its trip in the same depot, (ii) every client is attended by a single one vehicle once only, (iii) the total demand of every route does not exceed the vehicle capacity and (iv) the routes traveled are minimized (Montoya et al., 2015). Kulkarni and Bhave (1985) propose a three-index mathematical model that requires the definition of a binary decision variable x_{ijk} that takes the value of "1" when two nodes i and j are in the vehicle route k and take the "0" value otherwise. The model is formulated as a generalized TSP problem.

2.1. Objective function for route balancing

To define the objective function whose purpose is balancing the routes in Halvorsen and Savelsbergh (2016) and Schwarze and Voß (2013), different approaches available in the literature are describe that include load-balance VRP modifications. In all the cases, l_r, l_t and l_u are the lengths of the routes r, t and u, that belong to the set of routes T, being |T| the number of routes in the solution and l the average route length.

In Eq. (1), the maximum route length is minimized.

$$\min \max_{u \in T} l_u \tag{1}$$

In Eq. (2), the difference between the maximum and minimum route length is minimized.

$$\min(\max_{u \in T} l_u - \min_{u \in T} l_u) \tag{2}$$

In Eq. (3), the accumulated difference between each route length and the shortest one of these is minimized.

$$\min \sum_{t \in T} (l_t - \min_{u \in T} l_u) \tag{3}$$

In Eq. (4), the variance of the route length is minimized.

$$\min\left(\frac{\sum_{t\in T}l_t^2}{|T|}-\left(\frac{\sum_{t\in T}l_t}{|T|}\right)^2\right)$$ (4)

In Eq. (5), the relative deviation of the lengths, regarding the maximum length is minimized.

$$\min\frac{1}{|T|}\sum_{t\in T}\left(\frac{\max_{u\in T}l_u-l_t}{\max_{u\in T}l_u}\right)$$ (5)

In Eq. (6) the summation of the absolute deviation of the length is minimized from an average already known in advanced (parameter).

$$\min\sum_{t\in T}|l_t-\bar{l}|$$ (6)

Eq. (7) minimizes the summation of the absolute deviation of the length from an average already known in advanced.

$$\min\sqrt{\sum_{r\in T}(l_r-\mu)^2\Big/|T|}.$$ (7)

The above equations show an approximated value of the route-balance measurement; however, the objective function (7), which measures the standard deviation for the length of the routes, is the most accurate to observe the route balancing even though it implies a greater computational effort.

The objective function (7) is chosen, since the standard deviation is the measure of better behavior around the average value (Ribeiro & Ramalhinho, 2001). The objective function (1) is the easiest implementation; however, the results obtained are of low quality. In (2), (3) and (5) the length of the shortest route is subtracted, presenting undesirable behavior (Schwarze & Voß, 2013). In objective function (6), a predefined average value must be assumed which makes difficult the calculation. Objective functions above display an approximated value of the route-balance measurement. Objective functions (4) and (7), which measure the variance and standard deviation respectively for the length of the routes, are the most accurate to observe the route balancing even though they are quadratic functions and greater computational effort is required. Finally, the objective function (7) is chosen, since the standard deviation is the measure of better behavior around the average value (Ribeiro, Ramalhinho, 2001).

2.2. Mathematical model

The equations of the model are shown below

Nomenclature

Sets
C   Set of clients C = {$1,\ldots,n$}.
D   Set of depots D = {$n+1,\ldots,n+m$}.
V   Set of vertices V = $C\cup D$.

Parameters

D_{ij} Distance between nodes i and j.
C_{ij} Cost associated with the trajectory between nodes i and j.
D_j Quantity of the product to deliver to every client $j\in C$.

Variables

x_{ijk} Binary variable which indicates whether the path between clients $i,j \in V$ is traveled, both belonging to depot k.

y_{ik} Auxiliary binary variable which indicates whether the path between clients $i,j \in V$ is traveled, both belonging to depot k.

s_{ij} Quantity of merchandise carried between nodes i and j.

Equations of the mathematical model are shown as follows,

$$\Psi_1 = min \sum_{i \in V} \sum_{j \in V} \sum_{k \in D} c_{ij} x_{ijk} \tag{8}$$

$$\Psi_2 = min \sqrt{\sum_{r \in T} (l_r - \mu)^2 \Big/ |T|} \tag{9}$$

subject to

$$\sum_{k \in D} \sum_{i \in V} x_{ijk} = 1 \qquad \forall j \in C \tag{10}$$

$$\sum_{k \in D} \sum_{j \in V} x_{ijk} = 1 \qquad \forall i \in C \tag{11}$$

$$\sum_{i \in C} x_{kik} - \sum_{i \in C} x_{ikk} = 0 \qquad \forall k \in D \tag{12}$$

$$\sum_{k \in D} y_{ik} = 1 \qquad \forall i \in C \tag{13}$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = y_{ik} \qquad \forall k \in D \wedge \forall i \in C \tag{14}$$

$$x_{kjk} \le y_{jk} \qquad \forall j \in C \tag{15}$$

$$x_{jkk} \le y_{jk} \qquad \forall j \in C \tag{16}$$

$$\sum_{i \in C} s_{ki} = \sum_{i \in C} y_{ki} d_i \qquad \forall k \tag{17}$$

$$s_{ik} = 0 \qquad \forall i \in C \wedge \forall k \in D \tag{18}$$

$$\sum_{k} \sum_{i \in V} x_{ijk} s_{ij} = d_j + \sum_{k} \sum_{l \in V} s_{jl} x_{jlk} \qquad \forall i \in C \tag{19}$$

$$0 \le s_{ij} \le \sum_{k} x_{ijk} M \qquad \forall i,j \tag{20}$$

$$s_{kj} \le Q y_{jk} \qquad \forall j \in C \wedge k \in D \tag{21}$$

$$x_{ijk} \in \{0,1\} \tag{22}$$

$$y_{ik} \in \{0,1\} \tag{23}$$

$$S_{ij} \in R \tag{24}$$

The multi-objective model has two objective functions; the objective function (8) minimizes the total distance traveled by the vehicles from the k depots.

The objective function (9) is formulated considering (Halvorsen, Savelsbergh, 2016) and the minimization of the standard deviation of the distance traveled by every route in the solution, where μ is the average distance of every route in the solution and l_r is the length of every route. A greater

computational effort is required to calculate the mean due to the need of knowing the length of all the routes in the solution; however, better results are obtained.

The constraints (10) and (11) guarantee that all the arcs arriving to a node and leaving a node must be equal to one. The constraint (12) guarantees that the number of vehicles arriving and leaving a depot is the same. A client i assigned to a unique depot k is assured by constraint (13), thus sets of clients assigned to determined depots are obtained. The arrival and departure of a single arc to node i assigned to node k is guaranteed in constraint (14). The connection between a node and its respective depot is guaranteed by constraints (15) and (16). The constraints (17) and (18) show the flow equations in which the demand for each client is guaranteed and that the demand in the depot is equals to 0.

Constraint (19) guarantees that the amount of resources leaving node i, is equal to the difference between the amount of resources entering node i and the resources delivered to node i. The restrictions (20) guarantees that the flow s_{ij} between nodes i and j is considered if and only if arc x_{ijk} is active. Depot k capacity is restricted in (21). The type of variables used in the mathematical model are shown in constraints (22), (23) and (24).

3. Methodology

In general, the multi-objective problems have been solved using metaheuristics based on sets of solutions called population, and evolutionary algorithms such as the NSGA-II (Non-Dominated Sorting Generic Algorithm), where a genetic algorithm is used for generating a dominance- based population and ordering of solutions. The NSGA-II, uses selection and mutation operators to create half of the population following the selection of the best solutions (according to the function and the diversity adaptation). For most problems, the results show that NSGA-II is capable of finding diverse solutions and good convergence close to the optimal Pareto front, in comparison to the multi-objective evolutionary algorithms (MOEAs) (Deb et al., 2002).

Given the above, Geiger (2008), explains a metaheuristic for solving multi-objective optimization problems denominated as Pareto Iterated Local Search (PILS). PILS combines proper characteristics of how metaheuristic algorithms operate, whose development is based on two stages: intensification and diversification. The intensification is done by applying the VNS explained by Mladenovic and Hansen (1997). On the other hand, the diversification is performed by applying a perturbation which uses operators to avoid getting stuck in local optima.

An adaptation of the method previously explained is presented in this work, where a front of non-dominated solutions of constant size F is created. On each solution, s that belongs to the front, a local search and a perturbation is made. The new s_0 solutions are evaluated and selected according to their non-dominance front F. The local search is performed by using a modified VNS to evaluate the two objectives of the problem. The procedure is explained in Algorithm 1 named MOILS (Muti-Objective ILS).

The procedure starts by obtaining an initial solution s_0 using the algorithm cited by Paessens (1988) (step 2 of Algorithm 1). From this solution, a search in the neighborhood of the initial solution s_0 using inter-route operators denominates as *Inter_Ruta* is performed. Initially, the operators list is enable to be used during the iterative process (steps 6 and 7). As long as there are non-explored neighborhoods, a neighborhood v is randomly selected (steps 8 to 10). Then, from the randomly selected neighborhood v, the set of non-dominated solutions for every solution s of the front F is searched. The new set of non-dominated solutions is stored into F' (steps 11 and 12). The sets F and F' are blended and the front is updated (step 14). If during the former process, there was at least one non—dominated solution that became part of the front F, the search is performed over a set of neighborhoods with only modified routes *Intra-Ruta* operations (steps 16 to 20).

1:      **Algorithm 1:** MOILS
2:          $s_0 \leftarrow$ *initialSolution()*
3:          $F = \{s_0\}$
4:          *iter* $\leftarrow 1$
5:          **while** *iter* $<=$ *iterMax*
          do
6:                  **for** $v \in Inter\_Route$
                  do
7:                          *v.used* $\leftarrow false$
8:                  **while** $\exists v \in Inter\_Ruta \mid v.used = \textbf{false}$ **do**
9:                          *improveInter* $\leftarrow 0$
10:                         $v = random(Inter\_Ruta)$
11:                         **for** $s \in F$ **do**
12:                                 *improvInter* $\leftarrow v(s)$
13:                         **if** *improveInter* **then**
14:                                 $F \leftarrow F \cup F'$
15:
16:                         **for** $s \in F$ **do**
17:                                 **if** *s.modified* $==$ *true* **then**
18:   *improveIntra* $\leftarrow 0$
19:   **for** $w \in Intra\_Ruta$ **do**
20:   *improveaIntra* $\leftarrow$ w(s)
21:                                 $F \leftarrow F \cup F'$
22:                         **else**
23:                                 *v.used* $\leftarrow true$
24:                 **for** $s \in F$ **do**
25:                         *s'= Perturbation(s')*
26:                         *F'.add(s')*
27:                 $F \leftarrow F \cup F'$
28:                 *iter* $\leftarrow$ *iter* $+ 1$

The non-dominated solutions obtained during this process are stored in *F'* and after this; *F* is updated (step 21). In the case that no non-dominated solution in the neighborhood v is found, it is excluded from the list (step 23). On the current front *F*, a perturbation operation is performed on each of the front solutions, the new solutions are stored in *F'* (steps 24 to 26). Finally, the front is updated and a new iteration starts (steps 27 and 28).

1:      **Algorithm 2**: V(*S*)
2:          **for** $r_x \in S$ **do**
3:              **for** $r_y \in S$ **do**
4:                  **if** $r_i 6= r_j$ **then**
5:                      **for** $i \in r_x$ **do**
6:                          **for** $j \in r_y$ **do**
7:                              $S' \leftarrow set(S,i,j)$
8:                              *dominated* $\leftarrow false$
9:                              **for** $S \in F$ **do**
10:                                 **if** *dominated(S',S)* **then**
11:                                     *dominated* $\leftarrow true$
12:                                     **Break**
13:                             **if** $\neg dominated$ **then**
14:                                 $F' \leftarrow F' \cup S'$

Every one of the neighborhoods *v* in *Inter-Route* and *w* in Intra-Route operates in a general way, as described in Algorithm 2.

The search in the neighborhood *S* is performed exhaustively (steps 2 to 6). For every route r_x and r_y, different movements of clients *i* and *j* are made. According to the selected neighborhood, the objective functions are evaluated and a new solution *s'* is created (step 7). If the solution *s'* is not dominated by the current front *F*, it is added to the front *F'* (steps 9 to 14). In Table 1 and 2 a list of inter-route and intra-route neighborhoods is shown.

Table 1
List of inter-route neighborhoods

I	V(S)
1	swap(1,1)
2	swap(2,1)
3	swap(2,2)
4	insertion()
5	shift(2,0)
6	shift(3,0)
7	two-opt_paralell()
8	two-_opt_cross()

Table 2
List of intra-route neighborhoods

i	V(S)
1	swap(1,1)
2	insertion()
3	shift(2,0)
4	shift(3,0)
5	two_opt()

The perturbation is the mechanism that allows escaping from local optima. In this work, the perturbation mechanism is applied on the current front of solutions F, which consists in randomly applying a neighborhood operator considering the objective function of costs. To define the solution size belonging to the front *F*, the crowding distance explained by Deb et al. (2000) is used.

4. Computational Results

For the analysis, benchmark instances from Cordeau et al. (1997) were used. For each instance the Pareto front was obtained and from every front, three solutions were selected for later analysis. Two of them correspond to the solutions placed in the extremes of the front; the third one corresponds to the solution obtained by using the min-max metric, detailed by López et al. (2011). This metric normalizes the solutions for objective function values regarding to the extreme points; from these, the minimum value is selected and finally, the maximum value among the values previously chosen corresponding to a solution of equilibrium between both objectives is selected. The results were obtained starting from a solution with which a front is generated; a second front is generated from another initial solution, these solutions generate a Pareto front using the non-dominance concept. A third front is generated from another initial solution, which uses again the current Pareto front, the process continues until 10 fronts have been generated by updating the Pareto front in each iteration. The time presented in Table IV corresponds to the average time = total Time//10. The algorithm was run in an Intel Core i3, 3.3 GHz and 4 GB of RAM memory and implemented in C++. In Table 3, three different solutions of the Pareto front from the instance P01 are shown: two solutions correspond to the extreme points and the third one is obtained using the *min-max* metric.

Table 3
Solutions selected for the instance P01

	fo_1	fo_2	Length	
			max	Min
1	576.86	18.01	81.3	23.49
2	746.59	10.17	81.87	38.47
3	922.57	2.02	72.76	64.65

Fig. 1 presents an extreme point of the front, whose value for fo_1 corresponds to the optimal operation cost and fo_2 is the standard deviation measured among the solution routes. For this case, the maximum and the minimum route length is of 81.39 and 23.49 units of distance respectively.
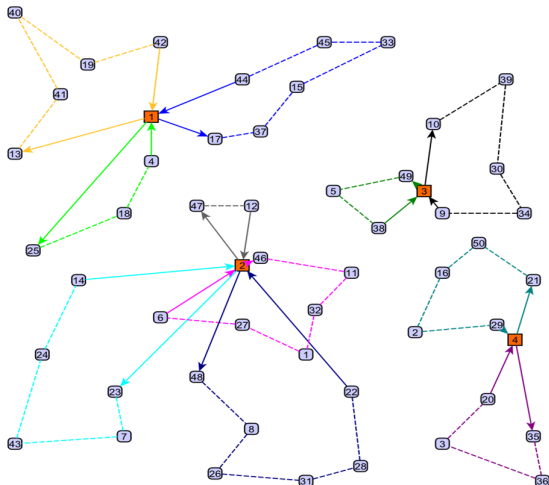
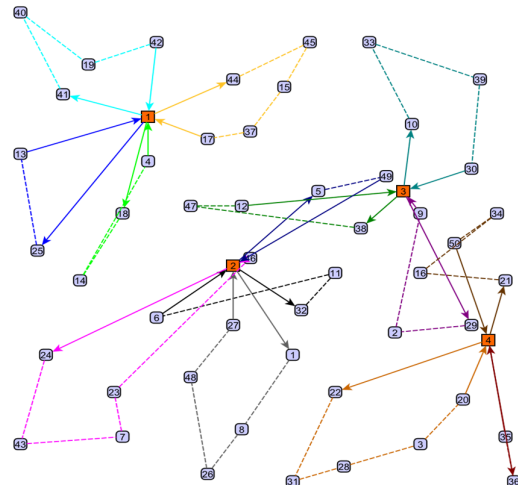**Fig. 1.** Solution for the instance P01 min fo_1

Fig. 2. Solution for the instance P01 min-max

A second solution is shown in Fig. 2, which was selected according to the *min-max* criterion where intermediate values for both objective functions were chosen. fo_1 is worse than the solution mentioned earlier, and fo_2 corresponds to a solution of better quality with maximum and minimum length for the routes of 81.87 and 38.47.
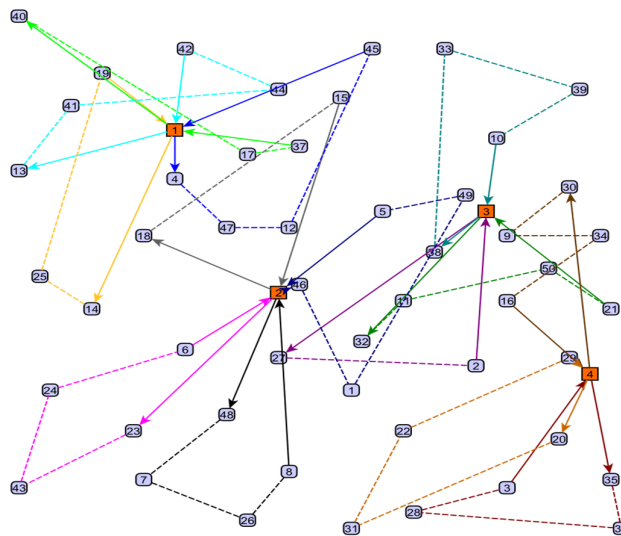

Fig. 3. Solution for the instance P01 min fo_2

The third solution is shown in Fig. 3, with a value for fo_2 equivalent to the standard deviation of 2.02 with a route length between 72.76 and 64.65 showing an appropriate route balancing. As opposed to point 1, the fo_1 presents a high value, apparently moving away from the optimal value. In brief, the optimal operation cost value is presented in Fig. 1 and the optimal route balance value is presented in Fig. 3. The intermediate value between the two optimal values is shown in Fig. 2; that is, one objective function is deteriorated to benefit the other one until a position of equilibrium is achieved. In all the cases studied, a conflict between both objectives was observed. The results using the benchmark instances proposed by (Cordeau et al., 1997), excluding those with a length constraint are shown in Table 4. In fo_2, low values indicate proper route balances that would have similar length in the case under study. However, this condition in the fo_1 expresses elevated operation costs that would make the implementation infeasible. Thus, it is necessary to select a point of equilibrium for both the economic and the social, for which the min-max concept is applied in this study. Nonetheless, this decision may be considered according to the decision-making criterion.

Table 4
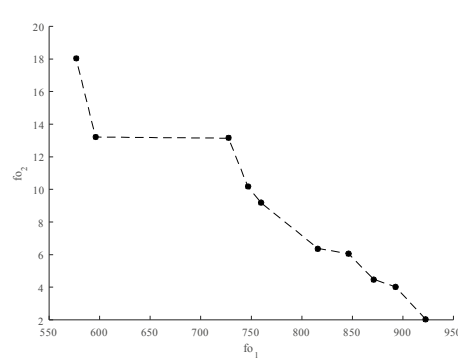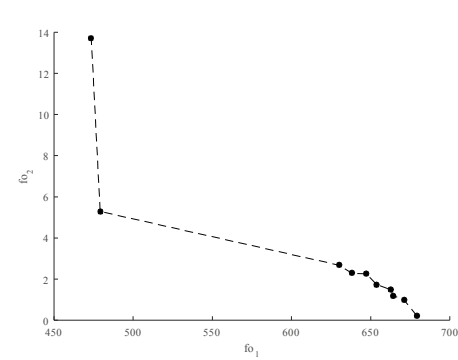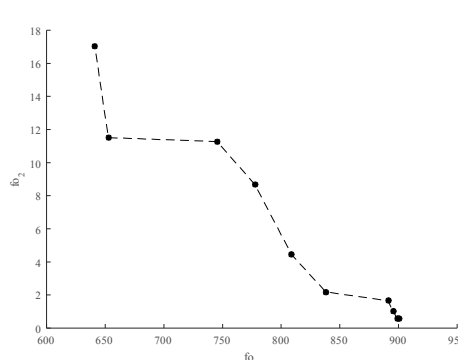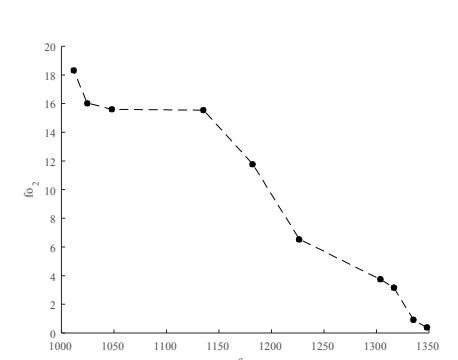Computational results for instances used by Cordeau, et al. (1997)

Instance	N of routes	f_{o1}	f_{o2}	length min	max	time (s)	Pareto's Front
p01	14	922.577	2.02595	64.6	72.7	20.9	
	14	892.817	4.01729				
	14	871.067	4.48472				
	14	846.301	6.05546				
	14	816.088	6.37164				
	14	759.654	9.18066				
	14	746.594	10.1763	38.4	81.8		
	14	727.509	13.1408				
	11	595.89	13.2128				
	11	**576.866**	18.0128	23.5	81.4		
p02	7	679.093	0.22556	96.7	97.4	20	
	7	671.008	0.98959				
	7	664.298	1.18972				
	7	662.8	1.48891				
	7	653.841	1.73567				
	7	647.207	2.2578				
	7	638.214	2.28202				
	7	630.14	2.67438				
	5	479.161	5.28649	87	101.9		
	5	**473.533**	13.719	74.2	115		
p03	13	900.248	0.54712	68.6	70.5	54.4	
	13	899.061	0.58397				
	13	895.291	1.02138				
	13	891.116	1.65167				
	13	838.114	2.18347				
	13	808.936	4.42969				
	13	777.556	8.68057	42.9	80.4		
	13	745.386	11.2673				
	11	652.649	11.5113				
	11	**641.186**	17.0489	27.8	87.8		
p04	16	1347.88	0.36727	83.89	84.81	71.9	
	16	1335.14	0.93156				
	16	1316.44	3.17538				
	16	1303.93	3.7323				
	16	1226.44	6.54645				
	16	1182.22	11.7783	47.9	92.8		
	16	1135.46	15.5395				
	16	1047.78	15.6012				
	16	1024.62	16.0456				
	16	1011.42	18.3273	34.6	98.1		

Table 4

(Continued): Computational results for instances used by Cordeau, et al. (1997)
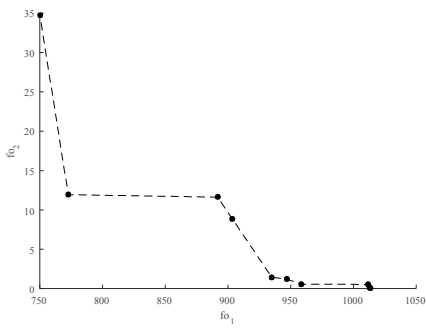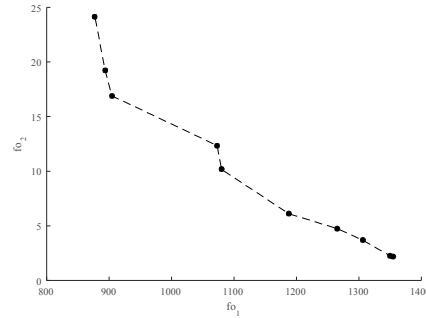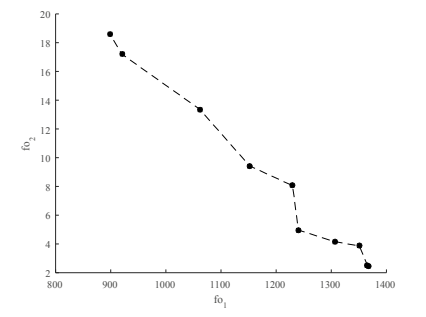
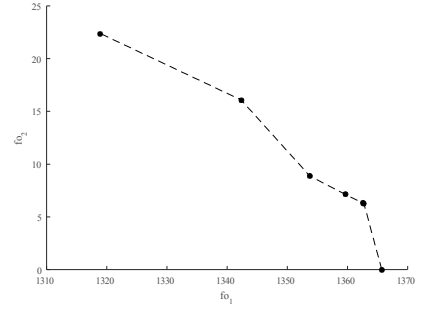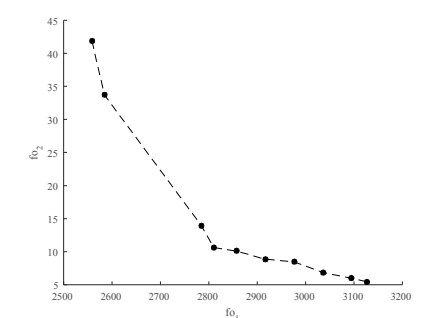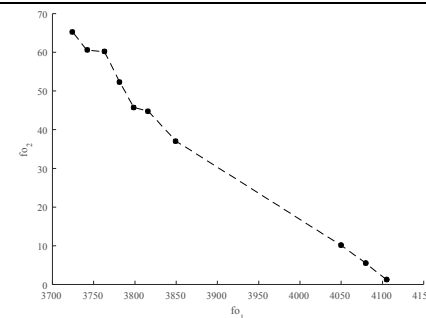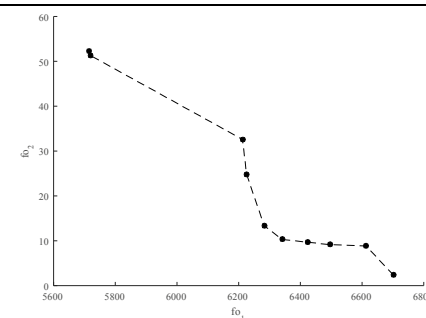| Instance	N of routes	f_{o1}	f_{o2}	length		time (s)	Pareto's Front
				min	*max*		
p05	8	1013.12	0.08316	126.5	126.8	73	
	8	1012.66	0.10618				
	8	1011.37	0.52504				
	8	958.453	0.56168				
	8	946.976	1.17268				
	8	934.755	1.44528				
	8	903.416	8.87127				
	8	891.792	11.6163				
	8	772.686	11.9385	80.6	114.3		
	8	**750.029**	34.7853	27.9	138.9		
p06	17	1355.02	2.15879	76.4	87.3	85.9	
	17	1350.37	2.24798				
	17	1306.05	3.68383				
	17	1265.44	4.74563				
	17	1187.99	6.09463				
	17	1079.87	10.1689	47.7	88.6		
	17	1072.93	12.3476				
	16	904.582	16.8998				
	16	893.432	19.2495				
	16	877.339	24.1354	4.5	106.5		
p07	18	1368.13	2.46672	71.9	82.3	101.9	
	18	1365.3	2.52425				
	18	1351.03	3.86859				
	18	1306.91	4.15166				
	18	1240.45	4.96447				
	18	1229.75	8.061				
	18	1152.27	9.41772	42.1	81.1		
	17	1061.83	13.3558				
	16	920.47	17.1919				
	16	898.77	18.5724	28	86.9		
p12	8	1365.69	0	170.7	170.7	36.1	
	8	1362.73	6.2664				
	8	1362.53	6.32003				
	8	1359.62	7.13268				
	8	1353.76	8.85959				
	8	1342.32	16.089	128.5	189.6		
	8	**1318.95**	22.3753	128.5	189.6		

Table 4

(Continued) Computational results for instances used by Cordeau, et al. (1997)

Instance	N of routes	f_{o1}	f_{o2}	length min	max	time (s)	Pareto's Front
p15	16	3125.85	5.45829	190	214.3		
	16	3093.58	5.99647				
	16	3036.33	6.85244				
	16	2977.16	8.44057				
	16	2916.44	8.86239			271.1	
	16	2857.29	10.0966				
	16	2810.63	10.6393				
	16	2784.37	13.9541	147.8	214.3		
	16	2584.2	33.7146				
	16	2558.61	41.8434	80.6	205.1		
p18	24	4104.98	1.22443	170.7	176.6		
	24	4079.57	5.53979				
	24	4049.37	10.1818				
	24	3849.52	37.0899	102.4	209.6		
	24	3816.09	44.8138			1274.6	
	23	3798.72	45.7234				
	24	3781.47	52.3456				
	24	3763.02	60.1596				
	24	3741.95	60.5524				
	24	3724.42	65.2649	48.3	233.1		
p21	36	6702.61	2.32945	183.1	204.4		
	36	6612.01	8.86635				
	36	6496.67	9.11763				
	36	6424.84	9.68818				
	36	6340.65	10.3036				
	36	6282.4	13.3801			6982.8	
	36	6225.61	24.8171	71.6	238.5		
	36	6212.38	32.5914				
	36	5718.64	51.3325				
	36	5714.56	52.2805	34.1	239.2		

This table also presents the solutions of the front of each one of the instances studied, the maximum and minimum length of the routes for the extreme points and that by using the min-max criterion. In the minimum cost point, a big difference between the minimum and the maximum route length is shown. On the contrary in the extreme point of balance, high costs and a relatively low difference between the minimum and the maximum route length are evidenced. The solution selected using the *min-max* criterion depicts a proper relation between costs and route balance. The time of the previous table corresponds to the average of the 10 cases studied with different seeds is, where a front is generated for each case. The fronts that are being generated update the incumbent front using the non-dominance criterion. The objectives behavior in the studied instances is explained through Fig. 4. In general, the routing problems have been studied so far using different models and solution methods whose objective is the cost minimization. In all the studied solutions, it is observed that a great imbalance is presented when the cost is optimized, which entails social inequality. On the other hand, if the problems were studied with an objective function for route balance, solutions socially equitable will be obtained but with operative costs excessively high, not suitable for a real-life operation. Given the conflict of interests associated, it is intended to stablish a methodology that considers a balance between both objectives. The methodology is focused on identifying a common region for both objectives. Table 5 presents the results considered as acceptable since they intersect minimum cost and route balance as presented in Fig. 4.
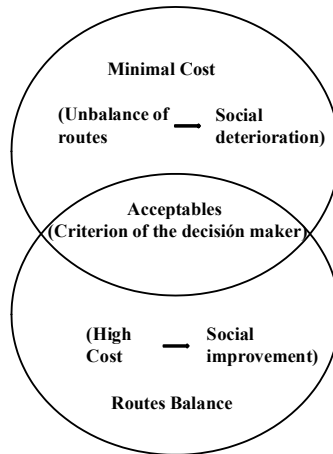
Fig. 4. Graphic analysis about obtained fronts

Table 5
Variations with respect to the minimum cost solution

Instance	Variation fo_1 (%)	Variation fo_2 (%)	fo_1 min-max	fo_2 min-max
P01	29	77	746.594	10.17
P02	1	160	479.161	5.28
P03	21	96	777.556	8.68
P04	17	55	1182.22	11.77
P05	3	191	772.686	11.93
P06	23	137	1079.87	10.17
P07	28	97	1152.27	9.42
P12	2	39	1342.32	16.09
P15	8	199	2784.37	13.95
P18	3	76	3849.52	37.09
P21	9	111	6225.61	24.82

To calculate Table 5 the minimum cost solutions are used as reference, which are generally used in the decision making. From this value, the percetage variation is obtained according to the obtaind with the min-max criterion that corresponds to columns fo_1 and fo_2. In the previous table, it is possible to establish that with small deteorations in the operation-cost, great social improvements are obtained. The solutions presented in Table 5 are considered acceptable as it is shown in Fig. 4. These solutions may be determined by some criteria according to decision-maker considerations.

5. Conclusions and Future Work

A multi-objective methodology based on the ILS metaheuristic was proposed for solving the MOMDVRP considering two objective functions. The first one consists in minimizing the operation cost, the second one is the route balancing using the standard deviation among the routes as a criterion for achieving the aforementioned objective. It was observed that in all instances the objective functions are in conflict. In the solutions of the front extreme points, a big difference is observed not only in cost but also in the route balance, with the biggest differences found in the latter. It is also observed that with relatively small deteriorations of the cost function great benefits in the route balance are achieved. The model was tested with the instances proposed by Cordeau et al. (1997), obtaining excellent quality results. The ILS operators were efficiently adapted to the multi-objective obtaining great-quality fronts with a high-diversity degree among them. With the objective of improving the computational effort, it is necessary to define strategies in the following aspects: (i) the front size, (ii) solutions in the front to perform the VNS, (iii) efficient computation of the objective functions, (iv) multi-objective specialized operators and (v) parallel solution methods.The social objective is an issue that has been little treated in the studies that refer to load transportation. The study set forth in this article, shows the conflict between

operative cost and route balance. It was also noticed that points of equilibrium between both objectives might be obtained using relatively low deteriorations in the cost function.

References

Cordeau, J. F., Gendreau, M., & Laporte, G. (1997). A tabu search heuristic for periodic and multi-depot vehicle routing problems. *Networks, 30*(2), 105-119.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000, September). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *International Conference on Parallel Problem Solving From Nature* (pp. 849-858). Springer Berlin Heidelberg.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182-197.

Duarte, A., Pantrigo, J. J., Pardo, E. G., & Mladenovic, N. (2015). Multi-objective variable neighborhood search: an application to combinatorial optimization problems. *Journal of Global Optimization*, 63(3), 515-536.

Geiger, M. J. (2008). Foundations of the Pareto iterated local search metaheuristic. *arXiv preprint arXiv:0809.0406*.

Halvorsen-Weare, E. E., & Savelsbergh, M. W. (2016). The bi-objective mixed capacitated general routing problem with different route balance criteria. *European Journal of Operational Research*, 251(2), 451-465.

Ho, W., Ho, G. T., Ji, P., & Lau, H. C. (2008). A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, 21(4), 548-557.

Keskinturk, T., & Yildirim, M. B. (2011, June). A genetic algorithm metaheuristic for bakery distribution vehicle routing problem with load balancing. In *Innovations in Intelligent Systems and Applications (INISTA), 2011 International Symposium on* (pp. 287-291). IEEE.

Kulkarni, R. V., & Bhave, P. R. (1985). Integer programming formulations of vehicle routing problems. *European Journal of Operational Research*, 20(1), 58-67.

Liefooghe, A., Humeau, J., Mesmoudi, S., Jourdan, L., & Talbi, E. G. (2012). On dominance-based multiobjective local search: design, implementation and experimental analysis on scheduling and traveling salesman problems. *Journal of Heuristics*, 18(2), 317-352.

López, L., Hincapié, R. A., & Gallego, R. A. (2013). Planeamiento multiobjetivo de sistemas de distribución usando un algoritmo evolutivo nsga-ii (multiobjective distribution system planning using an nsga-ii evolutionary algorithm). *Revista EIA*, 8(15), 141-151.

Mladenovic, N., y Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24(11), 1097–1100.

Montoya-Torres, J. R., Franco, J. L., Isaza, S. N., Jiménez, H. F., & Herazo-Padilla, N. (2015). A literature review on the vehicle routing problem with multiple depots. *Computers & Industrial Engineering*, 79, 115-129.

Paessens, H. (1988). The savings algorithm for the vehicle routing problem. *European Journal of Operational Research*, 34(3), 336-344.

Ribeiro, R., & Ramalhinho Dias Lourenço, H. (2001). A multi-objective model for a multi-period distribution management problem.

Toth, P., & Vigo, D. (Eds.). (2014). *Vehicle routing: problems, methods, and applications*. Society for Industrial and Applied Mathematics.

Schwarze, S., & Voß, S. (2013). Improved load balancing and resource utilization for the skill vehicle routing problem. *Optimization Letters*, 7(8), 1805-1823.