Comparative analysis of granular neighborhoods in a Tabu Search for the vehicle routing problem with heterogeneous fleet and variable costs (HFVRP)

Análisis comparativo de vecindarios granulares en una búsqueda tabú para el problema de ruteo de vehículos con flota heterogénea y costos variables (HFVRP)

Titulo portugues

John Willmer Escobar*
Wilson Adarme-Jaimes**
Nicolás Clavijo-Buriticá***

Fecha de recepción: 25 de agosto de 2016
Fecha de aceptación: 2 de junio de 2017

Abstract

In the vehicle routing problem with heterogeneous fleet and variable costs (HFVRP), the group of routes to be developed to satisfy the demand of the customer must be determined, considering the minimization of the total costs of the travelled distance. Heuristic algorithms based on local searches use simple movements (neighborhoods) to generate feasible solutions to problems related to route design. In this article, we conduct a comparative analysis of granular neighborhoods in a Tabu Search for the HFVRP, in terms of the quality of the obtained solution. The computational experiments, performed on instances of benchmarking for the HFVRP, showed the efficiency and effectiveness of implementing some neighborhoods in metaheuristic algorithms of path, such as the Tabu Search.

Key words: Granular neighborhoods; Heterogeneous fleet; Tabu search; Vehicle routing problems.

Resumen

En el problema de ruteo de vehículos con flota heterogénea y costos variables (HFVRP) se debe determinar el conjunto de rutas que se han de desarrollar para satisfacer las demandas de los clientes, teniendo en cuenta la minimización de la suma de los costos totales de la distancia recorrida. Algoritmos heurísticos basados en búsquedas locales utilizan comúnmente movimientos simples (vecindarios) para generar soluciones factibles en problemas relacionados con diseños de rutas. En este artículo se realiza un análisis comparativo de vecindarios granulares en una búsqueda tabú para el HFVRP. La comparación se ha realizado en términos de la calidad de la solución encontrada. Los experimentos computacionales, realizados sobre instancias de benchmarking para el HFVRP, muestran la eficiencia y efectividad de la implementación de algunos vecindarios en algoritmos metaheurísticos de trayectoria, como es la Búsqueda Tabú.

* Ph. D. Universidad del Valle (Cali-Valle del Cauca, Colombia). john.wilmer.escobar@correounivalle.edu.co. ORCID: 0000-0001-6175-9553.

** Ph. D. Universidad Nacional de Colombia (Bogotá-Distrito Capital, Colombia). wadarmej@unal.edu.co. ORCID: 0000-0001-7401-223X.

*** M. Sc. Pontificia Universidad Javeriana Cali (Cali-Valle del Cauca, Colombia). nicolas.clavijo@javerianacali.edu.co. ORCID: 0000-0001-5871-9390.

Palabras clave: Búsqueda tabú; Flota heterogénea; Problema de ruteo de vehículos; Vecindarios granulares

I. Introduction

The transport system is one of the crucial elements of the supply chain, and its management is one of the key activities of logistics. According to [1], the logistical costs can represent between 5 % and 18 % of the organization's sales volume. Among these costs, transport can represent between a third and two thirds of the total; therefore, one of the most important decisions regarding transport is the planning and programming of routes.

The decisions over programming distribution routes in their basic context have been mathematically modeled, using different approximations of the vehicle routing problem with capacities or VRP (Vehicle Routing Problem), which is considered a NP-hard problem [2]. Some variants of the VRP include restrictions such as simultaneous delivery and collection, time windows, routes length, multiple warehouses, heterogeneous fleet, different objective functions, and stochastic elements, among others. All these variations give rise to multiple research problems in the area of transport in logistics. In particular, this article considers the variant of the VRP with heterogeneous fleet (HFVRP), in which each vehicle has assigned a load capacity, a maximum traveling time, a variable travelling cost, and a fixed acquisition cost. The main objective of the HFVRP is to determine the group of developed routes that minimize the total costs of the travelled distance.

The HFVRP considers that each customer must be attended by a unique vehicle that provides all their demands; that the sum of the route customers' demands must not exceed the capacity of the assigned vehicle; that the length of a route, contemplating the travel and service time, should not exceed the maximum time set per trip; that the available vehicles are limited; and finally, that each route begins and finishes in the warehouse. The study of the HFVRP is of big interest for the scientific community, because it allows solving a lot of real cases of programming routes for different companies [3].

This paper compares different granular neighborhoods (called granular because the search is restricted only to a part of the complete graph (G)) inside a Tabu Search algorithm (TS) for the vehicle routing problem with heterogeneous fleet (HFVRP). In particular, this study aimed at evaluating different operators to determine the best design of routes, considering the quality of the solution as a way of comparison. The main contribution of this article is the comparative analysis of the neighborhoods in algorithms of local search, efficient for solving route design problems such as the HFVRP. In addition to the studies by [4] and [5], the articles that compare the use of granular neighborhoods for vehicle routing problems are limited.

Section 2 reviews the literature on HFVRP, whereas section 3 details the general diagram of the algorithm and the neighborhoods used; finally, section 4 describes the computational results, and sections 5 lists the conclusions of the analysis.

II. Literature review

The vehicle routing problem with heterogeneous fleet has diverse variants, depending on whether the fleet is fixed or unlimited, and whether the costs are variable or fixed [6–7]. Due to the computational complexity of the HFVRP, most of the technical proposals are framed within the heuristic and metaheuristics [8]; nevertheless, some authors have proposed exact algorithms to resolve the HFVRP [9-11].

Heuristic algorithms based on Tabu Search (TS) for the HFVRP have been proposed in the literature [12-14]. Brandão [12] introduced an algorithm based on TS that included strategic oscillation, perturbation procedure, and memory based on frequencies. Previously, the same author proposed a TS deterministic that used different heuristic procedures to generate the initial solution [13]. Finally, an algorithm based on TS that included mutations and different procedures of local search was proposed by [14].

Diagrams of solution based on evolution strategies (ES) for the HFVRP have been developed by [15] and [16]: [15] proposed a hybrid heuristic that combines genetic algorithms (AG) with a scatter search (SS); and [16] developed an algorithm designated approximated for the solution of the HFVRP. Additionally, hybrid algorithms to solve the HFVR have been proposed by [4, 17-22]: [17] proposed an algorithm of iterated local search (ILS), combined with a search of variable neighborhoods decline (VND) and a randomly neighborhood ordering (RVND); [18], an algorithm based on a variable neighborhood search (VNS) with several neighborhoods in the phase of local search; [19], an algorithm based on the development of a series of classical heuristics for the traditional VRP, followed by a local search (SDLS) and a TS; [20], an algorithm based on a procedure of adaptive memory with multi-start (multi-start AMP) and a modification of the traditional TS; [21], a method called threshold of acceptance (Threshold Accepting Approach), which adapts the procedure of simulated annealing (SA); and finally, [22] proposed a combination of the algorithm record-to-record with the method of the threshold of acceptance to solve the two variants of the HFVRP.

The comparative analysis developed in this study complements the state of the art, in terms of the efficiency of using granular neighborhoods within metaheuristics algorithms based on path. In the revised literature, with the exception of the work proposed by [3], there are no algorithms that apply the granular idea introduced by [4] for the HFVRP.

III. General diagram of the algorithm based in Tabu Search

In this section, we detail the principal aspects related to the proposed algorithm and the neighborhoods used in the comparative analysis for the HFVRP.

A. Granular space search

The concept of granularity, which was originally introduced by [4], consists in reducing the computation time when exploring neighborhoods, keeping solutions of high quality for heuristic algorithms based on local searches. In particular, the goal is to obtain "promising neighborhoods" (neighboring solutions of high quality) by means of using a list of candidates (incomplete graph G'), which contains the "short" arches of the complete graph G, the arches incident to the warehouses, and the arches that belong to the best solutions found during the search [4]. In this way, the local search is intelligently conducted, generating neighborhoods with the pertaining arches to the graph G'. The "short" arches are defined base on a threshold of granularity ϑ. An arch can be defined as "short" if its distance is lower than $\vartheta = \beta . \frac{z'}{(n+k)'}$; where n is the number of customers, k is the number of routes obtained in the initial solution of the algorithm, z' is the value of the objective function of the initial solution, and β is a dynamic sparsification parameter that is adjusted during the search.

In procedures of intensification of the algorithm, β is small and near to zero; whereas in stages of diversification, β takes higher values. Initially, the sparsification factor β is adjusted to a small value β_0, and the resultant arches of the graph G' are stored in a matrix. If the best feasible solution has not improved after $Iter_\beta$ iterations, the sparsification factor β is increased to a value β_f. Afterwards, the graph G' is recalculated and stored in a new structure. In this way, the search begins again by $Change_\beta$ iterations, beginning from the best feasible solution. Finally, the sparsification factor takes again its original value β_0 and the search continues. β_0, $Iter_\beta$, β_f and $Change_\beta$ are given parameters. Successful applications of the idea of granularity to different vehicle routing problems can be consulted in [23-30].

B. Algorithm based on Tabu Search

The proposed algorithm includes the generation of an initial solution through a heuristic procedure, and the improvement of such solution through TS, considering neighborhoods that follow the granularity idea mentioned above. Particularly, we considered four operators that generate granular neighborhoods, which will be detailed in section 3.3.

The pseudocode of the proposed algorithm is the following:

Pseudocode 1. Algorithm proposed.

Read instance to execute
Read parameters of GTS and CW
Calculate matrix of distances
Generate initial solution S_0 by through algorithm CW
Generate graph of S_0
Define Tabu Tenure
Define $S^* = S_0$
$Iterations = 0$
Select operator op
Initiate the procedure of tabu search
While do not reach the stopping criterion ($Iteracciones = I$) do

Create list of Candidates according to the parameter of sparsification β
Review possible feasible neighborhoods
If the arches of the neighborhoods belong to List of Candidates and are not in the Tabu list **then**
Calculate the neighborhood's cost
Calculate the feasibility of the new neighborhood in relation to the capacity of the vehicle and the length of the route
End If
Review the relocation of customers in the current routes
Select the neighborhood with lower cost
Save arches of the new neighborhood in the Tabu list
Modify the routes of the current solution S'
Calculate cost of the new solution S'
If S' is better than S^*

$$S^* = S'$$

End If
Update *Tabu tenure* in accordance with the behaviour of the process of search
End
Generate final solution S^* of the algorithm GTS
Create graph of the final solution S^*

In the pseudo code 1, we can observe that after generating the initial solution S_0, a TS is applied for improving the routes, defining the size of the tabu list (*magic*) and the movements to be used (*op*). The *magic* parameter is randomly defined in a rank $[t_{min}, t_{max}]$. The evaluation of neighborhoods begins with the generation of the list of candidates. Each granular neighborhood is obtained through the execution of simple movements (*op*) applied on the current solution (S'). Finally, during the search, the algorithm permanently updates both S^* and the tabu list.

The initial solution is generated through a heuristic procedure (CW) based on the savings algorithm. The pseudo code of the procedure of initial solution (CW) is shown below:

Pseudocode 2. Procedure of Initial Solution CW.

Build vector of savings
Order vector of savings of elder to minor
Order vector of trucks of elder to minor
Create n routes of customers (where n is the number of customers)
While there are available trucks **do**
Assign new truck to route of greater demand
While there are arches in vector of savings **do**
Review following arch in the vector of savings
If total route demand < *Captruck* then
Build new route
Update demand and length of route
End if
End
End
Create and assign new empty truck
End

The algorithm of initial solution allows the constant update of routes and allocation of trucks, attaining a balance between demand and capacity. The algorithm is an extension of the original procedure of the savings algorithm, applied to a vehicle routing problem with heterogeneous fleet, assigning trucks of elder to lower capacity.

C. Granular neighborhoods

Granular neighborhoods can be defined as enclosed feasible solutions generated by simple and promising movements with the arches that belong to the graph G' (See Section 3.1.). The computational time is drastically reduced, because the granular neighborhoods can be evaluated in less time than with an exhaustive local search, and without affecting the quality of the solution. The TS implemented is inclined to the use of these neighborhoods. In particular, a granular neighborhood (S'') is generated through $r - intercambios$ of arches of the solution S'. In other words, to generate S'', r arches of the current solution S' are stirred up, replacing them with other r new arches that must be in the list of candidates. In this case, it is said that S'' is a granular neighborhood of S'. According to [11], as the value of r increases, the computational time required to examine the neighborhoods increases as well. This characteristic is called neighborhood cardinality and can be measured as $O(n^r)$. In this study, we have considered small values of r $(r \leq 4)$. The four operations considered to generate granular neighborhoods were insertion, double insertion, exchange, and double exchange [24, 25].

IV. Computational results

A. Instances and resources used to execute the algorithm

The proposed algorithm was executed on a group of instances of benchmarking obtained from the literature. Table 1 summarizes the main characteristics of the considered instances, taking into account n number of customers and n_k number of available vehicles in the warehouse.

Table 1

Subset of used instances

Name of the instance in the literature	Source	n	n_k	Q_k Available Vehicles Capacity																					
				50	100	150	200	250	300	400	450	500	750	800	900	1000	1500	2000	2500	2700	3000	4000	4500	8000	15000
D023-03g	[31]	23	3													1							1	1	
D030-03g	[31]	30	3																				2	1	
D033-04g	[31]	33	4																				1	2	1
D051-06c	[32]	51	6	2	2			1			1														
D076-11c	[32]	76	11	2	5			3			1														
D101-09c	[32]	101	9	3				3			3														
D101-11c	[32]	101	11	2	2		5				2														
D121-11c	[32]	121	11	2	2		5				2														
D151-14b	[32]	151	14	2	3		6			3															
D151-14c	[32]	151	14	6	1		3				4														
D200-18b	[32]	200	18	2	1	10			3			2													
D200-18c	[32]	200	18	2	6		5		3			2													
D201-05k	[32]	201	5								2				2			1							
D241-10k	[33]	241	10		1		2			3				1		3									
D281-08k	[33]	281	8				1					4				2					1				
D321-10k	[33]	321	10				5					2				1		2							
D361-09k	[33]	361	9				1			2				2		3			1						
D401-10k	[33]	401	10					2				2	2			1	2	1							
D441-11k	[33]	441	11				6					2				1				1		1			
D481-12k	[33]	481	12									7				3			1		1				

The algorithm was programmed in C++ using the compiler Dev–C++ version 4.9.9.2, in an environment Windows 7 Home Premium. The computational experiments were executed in an Intel Core i5 (2.3 GHz) with 4 GB of RAM.

B. Parameterization

The success of the proposed algorithm directly depends on the precision in the parameter estimation; this is why extensive computational proofs were executed on the group of instances. Such experimentation determined that the critical parameter that directly affects the quality of the solution found by the granular neighborhoods is the value of β_f. In a first instance, to establish a rank of potential values of β_f, we used suggested values for similar problems [4]. In this way, values of β_f were considered from 0.5 to 2.5 with increases of 0.25. According to [4], the quality of the solutions found for the routing problems is not directly proportional to the increase of the sparsification parameter β, which is associated to a greater computational effort. We would expect that roughly selecting between 10 % and 20 % of the arches of the complete graph G, we could obtain solutions of high quality in a reduced computation time [4].

The value of the sparsification parameter β_f was adjusted in the analysis of every operator that generates the neighborhoods. In this way, a comparative analysis in terms of efficiency and quality of the solution was carried out. The other parameters were adjusted through the implementation of extensive computational proofs, fixing the operator that generates the solutions of best quality. Table 2 shows the parameter values obtained without considering the parameter value β_f, which is analyzed in the following section.

TABLE 2

PARAMETER VALUES

β_0	0.5
$Iter_\beta$	$2n$
$Change_\beta$	n
t_{min}	5
t_{max}	15
$Iteracciones$	$20n$

C. Results

Several tests (180 runs) were performed to analyze each neighborhood. Indeed, 20 instances multiplied by 9 times the value of sparsification parameter β_f were executed. In this way, it was possible to determine the quality of the solutions found during the search and the most efficient operator in terms of the solution's quality. In the subsequent tables, the shaded values reflect an improvement of the objective function in relation to the initial solution S_0.

1) Neighborhood insertion: Table 3 shows the results obtained by the operator insertion for each possible value of β_f.

TABLE 3

VALUES OF SOLUTION OBTAINED BY THE NEIGHBORHOOD INSERTION

Name	Values of the objective function S^* for the different values of β_f								
	0.50	**0.75**	**1.00**	**1.25**	**1.50**	**1.75**	**2.00**	**2.25**	**2.50**
D023-03g	674.06	674.33	648.40	611.11	823.47	823.47	823.47	823.47	823.47
D030-03g	438.23	433.82	433.82	442.75	442.75	442.75	522.75	522.75	522.75
D033-04g	694.76	694.76	694.76	694.76	716.86	716.86	713.90	712.09	853.35
D051-06c	504.73	506.05	538.98	565.72	564.05	589.21	656.14	639.75	656.14
D076-11c	763.07	755.16	773.80	744.06	830.15	830.15	820.23	830.15	830.15
D101-09c	874.03	824.39	826.57	994.15	994.15	994.15	994.15	994.15	994.15
D101-11c	887.22	877.43	990.22	812.61	1015.57	785.11	1025.44	1045.14	1099.92
D121-11c	1012.00	1043.53	1049.64	1039.87	1050.92	1059.52	1037.43	1037.43	1069.43
D151-14b	1045.41	1017.54	1044.14	1050.45	1056.46	1072.05	1081.05	1081.05	1081.05
D151-14c	1135.51	1135.51	1135.51	1135.51	1135.51	1135.51	1135.51	1135.51	1135.51
D200-18b	1349.20	1349.20	1349.20	1349.20	1349.20	1349.20	1349.20	1349.20	1349.20
D200-18c	1371.25	1371.25	1371.25	1371.25	1371.25	1371.25	1371.25	1371.25	1371.25
D201-05k	7243.70	7243.70	7243.70	7243.70	7243.70	7243.70	7243.70	7243.70	7243.70
D241-10k	6003.64	6003.64	6003.64	6003.64	6003.64	6003.64	6003.64	6003.64	6003.64
D281-08k	9317.24	9443.79	9707.26	9782.39	9914.26	9914.26	9914.26	9914.26	9914.26
D321-10k	9403.01	9403.01	9403.01	9403.01	9403.01	9403.01	9403.01	9403.01	9403.01
D361-09k	12612.22	12612.22	12612.22	12612.22	12377.12	12612.22	12612.22	12612.22	12612.22
D401-10k	12741.26	12534.16	12534.16	12534.16	12708.89	12741.26	12741.26	12741.26	12741.26
D441-11k	13080.49	13097.91	13641.02	13701.24	13816.67	14042.75	14086.72	14223.77	14254.48
D481-12k	17234.28	17234.28	17234.28	17234.28	17234.28	17234.28	17234.28	17234.28	17234.28

The greatest number of improvements was observed for $\beta_f = 0.75$ and $\beta_f = 1.25$ (Table 3). In particular, about 10 % and 20 % of the edges of the complete graph G were selected by considering these values of the parameter beta. The value of β_f was fixed to 0.75, based on the percentage of improvements in relation to the initial solution. By applying the neighborhood insertion, we obtained 12 improvements in the value of the objective function of the initial solution S_0. Of this number of improvements, the greatest proportion was found in the two first subsets of instances.

2) Neighborhood double insertion: Table 4 shows the results obtained by the operator double insertion for each possible value of β_f.

TABLE 4
VALUES OF THE SOLUTION OBTAINED BY THE NEIGHBORHOOD DOUBLE INSERTION

Name	Values of the objective function S^* for the different values of β_f								
	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25	2.50
D023-03g	823.47	823.47	823.47	823.47	823.47	823.47	823.47	823.47	823.47
D030-03g	454.30	454.30	461.31	454.30	454.30	438.83	458.23	458.23	520.00
D033-04g	751.80	751.80	751.80	761.64	761.64	761.64	761.64	756.35	719.95
D051-06c	602.84	608.83	574.40	605.33	582.02	656.14	656.14	656.14	656.14
D076-11c	754.34	783.73	804.36	782.58	800.18	776.81	830.15	830.15	830.15
D101-09c	876.91	850.99	994.15	994.15	994.15	994.15	994.15	994.15	994.15
D101-11c	1023.52	993.14	1099.92	1064.90	1099.92	1099.92	1099.92	1099.92	1099.92
D121-11c	992.59	980.70	1041.05	981.52	1045.61	1008.73	982.76	1107.83	1046.16
D151-14b	1081.05	1069.80	1081.05	1070.35	1081.05	1081.05	1080.45	1081.05	1081.05
D151-14c	1113.90	1135.51	1135.51	1135.51	1135.51	1135.51	1135.51	1135.51	1135.51
D200-18b	1349.20	1349.20	1349.20	1349.20	1349.20	1349.20	1349.20	1349.20	1349.20
D200-18c	1371.25	1371.25	1371.25	1371.25	1371.25	1371.25	1371.25	1371.25	1371.25
D201-05k	7243.70	7243.70	7243.70	7243.70	7243.70	7243.70	7243.70	7243.70	7243.70
D241-10k	6003.64	6003.64	6003.64	6003.64	6003.64	6003.64	6003.64	6003.64	6003.64
D281-08k	9914.26	9914.26	9914.26	9914.26	9914.26	9914.26	9914.26	9914.26	9914.26
D321-10k	9403.01	9347.91	9403.01	9369.23	9403.01	9403.01	9403.01	9403.01	9403.01
D361-09k	12612.22	12612.22	12612.22	12612.22	12377.12	12612.22	12612.22	12612.22	12612.22
D401-10k	12741.26	12741.26	12741.26	12741.26	12741.26	12741.26	12741.26	12741.26	12741.26
D441-11k	13469.79	13219.21	13481.99	13420.35	14162.84	13751.97	14492.00	14135.82	14104.72
D481-12k	17234.28	17234.28	17234.28	17234.28	17234.28	17234.28	17234.28	17234.28	17234.28

The greatest number of improvements of the initial solution was obtained implementing the algorithm with $\beta_f = 0.75$. However, neighborhood insertion obtained better solutions than the double insertion (Table 6).

3) Neighborhood exchange: Table 5 shows the results obtained by the neighborhood exchange for each possible value of β_f.

Table 5

Values of the solution obtained by the neighborhood exchange

Name	Values of the objective function S^* for the different values of β_f								
	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25	2.50
D023-03g	823.47	823.47	823.47	823.47	823.47	823.47	823.47	823.47	823.47
D030-03g	534.45	534.45	534.45	534.45	534.45	534.45	534.45	534.45	534.45
D033-04g	823.12	822.53	823.12	846.64	846.64	870.94	870.94	902.04	902.04
D051-06c	656.14	656.14	656.14	656.14	656.14	656.14	656.14	656.14	656.14
D076-11c	830.15	830.15	830.15	830.15	830.15	830.15	830.15	830.15	830.15
D101-09c	994.15	994.15	994.15	994.15	994.15	994.15	994.15	994.15	994.15
D101-11c	1099.92	1099.92	1099.92	1099.92	1099.92	1099.92	1099.92	1099.92	1099.92
D121-11c	1157.01	1157.01	1157.01	1157.01	1157.01	967.43	967.43	967.43	967.43
D151-14b	1081.05	1081.05	1081.05	1081.05	1081.05	1081.05	1081.05	1081.05	1081.05
D151-14c	1135.51	1135.51	1135.51	1135.51	1135.51	1135.51	1135.51	1135.51	1135.51
D200-18b	1349.20	1349.20	1349.20	1349.20	1349.20	1349.20	1349.20	1349.20	1349.20
D200-18c	1371.25	1371.25	1371.25	1371.25	1371.25	1371.25	1371.25	1371.25	1371.25
D201-05k	7243.70	7243.70	7243.70	7243.70	7243.70	7243.70	7243.70	7243.70	7243.70
D241-10k	6003.64	6003.64	6003.64	6003.64	6003.64	6003.64	6003.64	6003.64	6003.64
D281-08k	9914.26	9914.26	9914.26	9914.26	9914.26	9914.26	9831.77	9724.86	9892.62
D321-10k	9403.01	9403.01	9403.01	9403.01	9403.01	9403.01	9403.01	9403.01	9403.01
D361-09k	12612.22	12612.22	12612.22	12612.22	12612.22	12612.22	12612.22	12612.22	12612.22
D401-10k	12741.26	12741.26	12741.26	12741.26	12741.26	12741.26	12741.26	12741.26	12741.26
D441-11k	14667.15	14667.15	14667.15	14667.15	14667.15	14667.15	14667.15	14667.15	14667.15
D481-12k	17234.28	17234.28	17234.28	17234.28	17234.28	17234.28	17234.28	17234.28	17234.28

In the case of the exchange operator, the number of improvements of the initial solution was minimum. The greatest number of improvements was seen when $\beta_f = 2.0$. Furthermore, the average improvement of the final solution was minimum in comparison with the results obtained by the neighborhood previously described (Table 5).

4) Double operator exchange: This neighborhood obtained the worst results for the problem and set of considered instances. In fact, the granular neighborhoods generated by this operator did not improve the initial solution. Therefore, this operator was not included in the comparative analysis of the different neighborhoods shown in Table 6.

D. Analysis of the results

Table 6 summarizes the best results per operator, for the set of instances. Column "Best" shows the best results of the value of the objective function S^*, obtained by the total of operators for each instance; whereas column "Gap" shows the percentage variation of the results found by each operator in relation to the value in column "Best". When the operator was able to find the best result (Best), the result is in bold and underlined.

The worst result for the set of selected HFVRP instances was obtained from the granular neighborhoods generated by the exchange operator (Table 6). However, this operator was able to find the best solution in 8 of 20 instances, and was the only operator able to find the best solution for the instance D101-11c.

Table 6
Consolidated results for the neighborhoods

	Valor S_0	Best	Insertion	Gap	Double Inserción	Gap	Exchange	Gap
D023-03g	823.47	674.33	**674.33**	0.00%	823.47	22.12%	823.47	22.12%
D030-03g	534.45	433.82	**433.82**	0.00%	454.3	4.72%	534.45	23.20%
D033-04g	902.04	694.76	**694.76**	0.00%	751.8	8.21%	870.94	25.36%
D051-06c	656.14	506.05	**506.05**	0.00%	608.83	20.31%	656.14	29.66%
D076-11c	830.15	755.16	**755.16**	0.00%	783.73	3.78%	830.15	9.93%
D101-09c	994.15	824.39	**824.39**	0.00%	850.99	3.23%	994.15	20.59%
D101-11c	1099.92	877.43	**877.43**	0.00%	993.14	13.19%	1099.92	25.36%
D121-11c	1157.01	967.43	1043.53	7.87%	980.7	1.37%	**967.43**	0.00%
D151-14b	1081.05	1017.54	**1017.54**	0.00%	1069.8	5.14%	1081.05	6.24%
D151-14c	1135.51	1135.51	**1135.51**	0.00%	**1135.51**	0.00%	**1135.51**	0.00%
D200-18b	1349.20	1349.2	**1349.2**	0.00%	**1349.2**	0.00%	**1349.2**	0.00%
D200-18c	1371.25	1371.25	**1371.25**	0.00%	**1371.25**	0.00%	**1371.25**	0.00%
D201-05k	7243.70	7243.7	**7243.7**	0.00%	**7243.7**	0.00%	**7243.7**	0.00%
D241-10k	6003.64	6003.64	**6003.64**	0.00%	**6003.64**	0.00%	**6003.64**	0.00%
D281-08k	9914.26	9443.79	**9443.79**	0.00%	9914.26	4.98%	9831.77	4.11%
D321-10k	9403.01	9347.91	9403.01	0.59%	**9347.91**	0.00%	9403.01	0.59%
D361-09k	12612.22	12612.22	**12612.22**	0.00%	**12612.22**	0.00%	**12612.22**	0.00%
D401-10k	12741.26	12534.16	**12534.16**	0.00%	12741.26	1.65%	12741.26	1.65%
D441-11k	14667.15	13097.91	**13097.91**	0.00%	13219.21	0.93%	14667.15	11.98%
D481-12k	17234.28	17234.28	**17234.28**	0.00%	**17234.28**	0.00%	**17234.28**	0.00%

The granular neighborhoods obtained through the double insertion operator allowed to obtain good results. In fact, the algorithm was able to find the best results in 8 of 20 instances. Likewise, the best result for the instance D321-10k was obtained through this operator (Table 6).

Finally, insertion was the neighborhood that allowed to reach the best results in 18 of 20 instances considered for the HFVRP. The reason for this could be that the insertion operator allows to easily readjust routes with a simple movement. On the other hand, the double insertion operator, when keeping the arch that connects to the two nodes that go to transfer to another route, generated too many infeasible granular neighborhoods that were not included by the algorithm. Finally, in the exchange neighborhood, four new edges must belong to the candidates list, which causes that a large quantity of neighborhoods are excluded by considering the granularity criteria. Obviously, it is important to mention that the efficiency of the operators in the generation of neighborhoods is determined by the quality of the initial solution S_0, the value of the considered parameters, and the implementation form in the programming language.

V. Conclusions and future investigations

In this article, we compared granular neighborhoods inside a Tabu Search (TS) for the vehicle routing problem with heterogeneous fleet and variable costs (HFVRP). In particular, the proposed algorithm is inclined to using granular neighborhoods and strategies of diversification and intensification based on the change of a dynamic sparsification parameter that is updated during the search. The TS proposal accepts only feasible solutions that are obtained through the different neighborhoods.

The comparative analysis was carried out considering a group of instances of benchmarking for the HFVRP. The computational experiments showed the efficiency and effectiveness of implementing the insertion neighborhood in the proposed algorithm.

The use of granular neighborhoods led to a substantial improvement in relation to the transport costs of the used fleet and the total distance travelled. The results obtained suggest that the implementation of the insertion neighborhood in algorithms of search based on path could be applied to other problems of similar distribution. We proposed the following future research:

- Evaluation of the proposed methodology considering different objective functions, such as the minimization of the quantity of vehicles or the minimization of the environmental impact generated by the vehicles.

- Consideration of fixed costs of using the vehicles in the objective function.

- Generation of candidates lists that take into account the quantity of near customers, given selected customers. For each customer is considered that another is neighbor if it is inside the nearest customers no matter the distance that they are.

- Inclusion of dynamic probabilistic operators that change during the search, privileging the neighborhoods that generate better solutions [34-35].

References

[1] J. Engblom, T. Solakivi, J. Töyli, and L. Ojala. *Multiple-method analysis of logistics costs*, 2012.

[2] J. K. Lenstra, and A. H. G. Kan, "Complexity of vehicle routing and scheduling problems," *Networks*, vol. 11 (2), pp. 221-227, 1981. DOI: http://doi.org/10.1002/net.3230110211.

[3] D. E. Puenayán, J.C. Londoño, J. W. Escobar, and R. Linfati. "Un algoritmo basado en búsqueda tabú granular para la solución de un problema de ruteo de vehículos considerando flota heterogénea," *Revista Ingenierías Universidad de Medellín*, vol. 13 (25), pp. 81-98, 2014.

[4] P. Toth, and D. Vigo, "The granular tabu search and its application to the vehicle-routing problem," *Informs Journal on Computing*, vol. 15 (4), pp. 333-346, 2003. DOI: http://doi.org/10.1287/ijoc.15.4.333.24890.

[5] M. Schneider, and A. Stenger, *Designing granular solution methods for routing problems with time windows.* Technical Report Darmstadt Technical University, Department of Business Administration, Economics and Law, Institute for Business Studies (BWL), 2015.

[6] É. D. Taillard, "A heuristic column generation method for the heterogeneous fleet VRP. Revue française d'automatique, d'informatique et de recherche opérationnelle," *Recherche opérationnelle*, vol. 33 (1), pp. 1-14, 1999.

[7] B. Golden, A. Assad, L. Levy, and F. Gheysens, "The fleet size and mix vehicle routing problem," *Computers & Operations Research*, vol. 11 (1), pp. 49-66, 1984. DOI: http://doi.org/10.1016/0305-0548(84)90007-8.

[8] A. Subramanian, E. Uchoa, and L. S. Ochi, "A hybrid algorithm for a class of vehicle routing problems," *Computers & Operations Research*, vol. 40 (10), pp. 2519-2531, 2013. DOI: http://doi.org/10.1016/j.cor.2013.01.013.

[9] R. Baldacci, and A. Mingozzi, "A unified exact method for solving different classes of vehicle routing problems," *Mathematical Programming*, vol. 120 (2), pp. 347-380, 2009. DOI: http://doi.org/10.1007/s10107-008-0218-9.

[10] A. Subramanian, P. H. V. Penna, E. Uchoa, and L. S. Ochi, "A hybrid algorithm for the heterogeneous fleet vehicle routing problem," *European Journal of Operational Research*, vol. 221 (2), pp. 285-295, 2012. DOI: http://doi.org/10.1016/j.ejor.2012.03.016.

[11] A. Subramanian, E. Uchoa, and L. S. Ochi, "A hybrid algorithm for a class of vehicle routing problems," *Computers & Operations Research*, vol. 40 (10), pp. 2519-2531, 2013. DOI: http://doi.org/10.1016/j.cor.2013.01.013.

[12] J. Brandão, "A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem," *Computers & Operations Research*, vol. 38 (1), pp.140-151, 2011. DOI: http://doi.org/10.1016/j.cor.2010.04.008.

[13] J. Brandão, "A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem," *European Journal of Operational Research*, vol. 195 (3), pp. 716-728, 2009. DOI: http://doi.org/10.1016/j.ejor.2007.05.059.

[14] H. Jia, Y. Li, B. Dong, and H. Ya, "An improved tabu search approach to vehicle routing problem," *Procedia-Social and Behavioral Sciences*, vol. 96, pp.1208-1217, 2013. DOI: http://doi.org/10.1016/j.sbspro.2013.08.138.

[15] L. S. Ochi, D. S. Vianna, L. M. Drummond, and A. O. Victor, "An evolutionary hybrid metaheuristic for solving the vehicle routing problem with heterogeneous fleet," *Genetic programming*, Springer Berlin Heidelberg, pp. 187-195, 1998. DOI: http://doi.org/10.1007/BFb0055938.

[16] L. S. Ochi, D. S. Vianna, L. M. Drummond, and A. O. Victor, "A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet," *Parallel and Distributed Processing*, Springer Berlin Heidelberg, pp. 216-224, 1998. DOI: http://doi.org/10.1007/3-540-64359-1_691.

[17] P. H. V. Penna, A. Subramanian, and L. S. Ochi, "An iterated local search heuristic for the heterogeneous fleet vehicle routing problem," *Journal of Heuristics*,

vol. 19 (2), pp.201-232, 2013. DOI: http://doi.org/10.1007/s10732-011-9186-y.

[18] A. Imran, A. Salhi, and N. A. Wassan, "A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem," *European Journal of Operational Research*, vol. 197(2), pp. 509-518, 2009. DOI: http://doi.org/10.1016/j.ejor.2008.07.022.

[19] C. Prins, "Efficient heuristics for the heterogeneous fleet multitrip VRP with application to a large-scale real case," *Journal of Mathematical Modelling and Algorithms*, vol. 1 (2), pp. 135-150, 2002. DOI: http://doi.org/10.1023/A:1016516326823.

[20] X. Li, P. Tian, and Y. P. Aneja, "An adaptive memory programming metaheuristic for the heterogeneous fixed fleet vehicle routing problem," *Transportation Research Part E: Logistics and Transportation Review*, vol. 46 (6), pp.1111-1127, 2010. DOI: http://doi.org/10.1016/j.tre.2010.02.004.

[21] C. D. Tarantilis, C. T. Kiranoudis, and V. S. Vassiliadis, "A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem" *Journal of the Operational Research Society*, vol. 54 (1), pp. 65-71, 2003. DOI: http://doi.org/10.1057/palgrave.jors.2601443.

[22] F. Li, B. Golden, and E. Wasil, "A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem," *Computers & Operations Research*, vol. 34 (9), pp.2734-2742, 2007. DOI: http://doi.org/10.1016/j.cor.2005.10.015.

[23] J. W. Escobar, and R. Linfati, "Un algoritmo metaheurístico basado en recocido simulado con espacio de búsqueda granular para el problema de localización y ruteo con restricciones de capacidad," *Revista Ingenierías Universidad de Medellín*, vol. 11 (21), pp. 139-150, 2012.

[24] J. W. Escobar, R. Linfati, and P. Toth, "A two-phase hybrid heuristic algorithm for the capacitated location-routing problem," *Computers & Operations Research*, vol. 40 (1), pp.70-79, 2013. DOI: http://doi.org/10.1016/j.cor.2012.05.008.

[25] J. W. Escobar, R. Linfati, M. G. Baldoquin, and P. Toth, "A Granular Variable Tabu Neighborhood Search for the capacitated location-routing problem," *Transportation Research Part B*, vol. 67 (C), pp. 344-356, 2014. DOI: http://doi.org/10.1016/j.trb.2014.05.014.

[26] J. W. Escobar, R. Linfati, P. Toth, and M. G. Baldoquin, "A hybrid Granular Tabu Search algorithm for the Multi-Depot Vehicle Routing Problem," *Journal of Heuristics*, vol. 20 (5), pp. 483-509, 2014. DOI: http://doi.org/10.1007/s10732-014-9247-0.

[27] J. W. Escobar, "Heuristic Algorithms for the Capacitated Location-Routing Problem and the Multi-Depot Vehicle Routing Problem," *4OR - A Quarterly Journal of Operations Research*, vol. 12 (1), pp. 99-100, 2014.

[28] R. Linfati, J. W. Escobar, and G. Gatica, "Un algoritmo metaheurístico para el problema de localización y ruteo con flota heterogénea," *Ingeniería y Ciencia*, vol. 10 (19), pp.55-76, 2014. DOI: http://doi.org/10.17230/ingciencia.10.19.3.

[29] R. Linfati, J. W. Escobar, and B. Cuevas, "An algorithm based on granular tabu search for the problem of balancing public bikes by using multiple vehicles," *Dyna*, vol. 81 (186), pp.284-294, 2014. DOI: http://doi.org/10.15446/dyna.v81n186.45220.

[30] J. W. Escobar, R. Linfati, and W. Adarme, "Un algoritmo metaheurístico híbrido para el problema de localización y ruteo con restricciones de capacidad," *Revista DYNA*, vol. 82 (189), pp. 243-251, 2015.

[31] S. Eilon, C. D. T. Watson-Gandy, and N. Christofides, Distribution management. London: Griffin, 1971.

[32] N. Christofides, A. Mingozzi, and P. Toth, Loading problems. N. Christofides and al., editors, Combinatorial Optimization, pp. 339-369, 1979.

[33] B. L. Golden, G. Laporte, and É. D. Taillard, "An adaptive memory heuristic for a class of vehicle routing problems with minmax objective," *Computers & Operations Research*, vol. 24 (5), pp. 445-452, 1997. DOI: http://doi.org/10.1016/S0305-0548(96)00065-2.

[34] J. Bernal J. W. Escobar, J. C. Paz, G. Gatica, and R. Linfati, "A probabilistic Granular Tabu Search for the Distance Constrained Capacitated Vehicle Routing Problem (DCVRP)," *Journal of Industrial and Systems Engineering*, pre-print.

[35] J. Bernal, J. W. Escobar, C. Marin, R. Linfati, and G. Gatica, "A comparison of granular heuristic algorithms for the Location-Routing Problem with Heterogeneous Fleet (LRPH)," *Dyna*, vol. 84 (200), pp. 193-201, 2017. DOI: http://doi.org/10.15446/dyna.v84n200.55533.