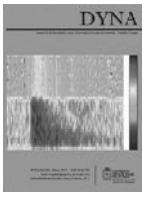
UNIVERSIDAD NACIONAL DE COLOMBIA
SEDE MEDELLÍN
FACULTAD DE MINAS



DYNA
http://dyna.medellin.unal.edu.co/

A comparison of trajectory granular based algorithms for the location-routing problem with heterogeneous fleet (LRPH)

José Alfonso Bernal-Moyano [a], John Willmer Escobar [b], Cesar Marín-Moreno [c], Rodrigo Linfati [d], & Gustavo Gatica [e]

[a] *Escuela de Ingeniería de Sistemas, Universidad del Valle, Colombia. jose.bernal@correounivalle.edu.co*
[b] *Departamento de Ingeniería Civil e Industrial, Pontificia Universidad Javeriana Cali, Colombia. jwescobar@javerianacali.edu.co*
[c] *Integra S.A., Universidad Tecnológica de Pereira, Colombia. cmarin@integra.com.co*
[d] *Departamento de Ingeniería Industrial, Universidad del Bío-Bío, Chile. rlinfati@ubiobio.cl*
[e] *Facultad de Ingeniería, Universidad Andres Bello, Chile. ggatica@unab.cl*

Received: February 1rd, 2016. Received in revised form: June 25th, 2016. Accepted: September 12th, 2016.

Abstract
We consider the Location-Routing Problem with Heterogeneous Fleet (LRPH) in which the goal is to determine the depots to be opened, the customers to be assigned to each open depot, and the corresponding routes fulfilling the demand of the customers and by considering a heterogeneous fleet. We propose a comparison of granular approaches of Simulated Annealing (GSA), of Variable Neighborhood Search (GVNS) and of a probabilistic Tabu Search (pGTS) for the LRPH. Thus, the proposed approaches consider a subset of the search space in which non-favorable movements are discarded regarding a granularity factor. The proposed algorithms are experimentally compared for the solution of the LRPH, by taking into account the CPU time and the quality of the solutions obtained on the instances adapted from the literature. The computational results show that algorithm GSA is able to obtain high quality solutions within short CPU times, improving the results obtained by the other proposed approaches.

Keywords: Location-routing problem; heterogeneous fleet; simulated annealing; variable neighborhood search; probabilistic tabu search; metaheuristic algorithms.

Una comparación de algoritmos basados en trayectoria granular para el problema de localización y ruteo con flota heterogénea (LRPH)

Resumen
Nosotros consideramos el problema de localización y ruteo de vehículos con flota heterogénea (LRPH) en el cual la meta es determinar los depósitos a ser abiertos, los clientes asignados a cada deposito, y las rutas que satisfagan la demanda de los clientes considerando una flota heterogénea. Nosotros proponemos una comparación de algoritmos granulares de Recocido Simulado (GSA), Búsqueda de Vecindario Variable (GVNS) y Tabú Search probabilístico (pGTS) para el LRPH. De esta manera, los algoritmos propuestos consideran un subconjunto del espacio en el cual los movimientos menos favorables son descartados según un factor de granularidad. Los algoritmos propuestos son comparados experimentalmente para la solución del LRPH, considerando el tiempo de CPU y la calidad de la solución obtenida en instancias adaptadas de la literatura. Los resultados computacionales muestran que el algoritmos GSA es capaz de obtener buenas soluciones en tiempos computacionales reducidos, mejorando los resultados obtenidos por los otros algoritmos propuestos.

Palabras clave: Problema de localización y ruteo; flota heterogénea; recocido simulado; búsqueda de vecindario variable; búsqueda tabú probabilística; algoritmos metaheurísticos.

1. Introduction

Long and short-term logistic operations consider Combinatorial Optimization Problems (COP). One of the most well studied decisions within the COP is the Traveling Salesman Problem (TSP), in which an agent should visit

How to cite: Bernal-Moyano, J.A., Escobar, J.W., Marín-Moreno, C., Linfati, R. and Gatica, R., A comparison of trajectory granular based algorithms for the location-routing problem with heterogeneous fleet (LRPH)., DYNA 84(200), pp. 193-201, 2017.

DYNA 84 (200), pp. 193-201, Marzo, 2017. Medellín. ISSN 0012-7353 Printed, ISSN 2346-2183 Online
DOI: http://dx.doi.org/10.15446/dyna.v84n200.55533

several cities and go back to its initial position in an optimal way minimizing the travelled distance and by considering that each city must visited only once. The problem could be represented by an undirected graph; where the vertex correspond to the cities and the arcs to the distances to be traveled satisfying the corresponding demand of cities [1]. The TSP is classified as NP-Hard problem, being computationally intractable with the number of nodes is increased [2,3].

The well-known Vehicle Routing Problem (VRP), is based on the principles of the TSP. Different variants of the Vehicle Routing Problem have been studied deeply in the fields of the Operations Research and Combinatorial Optimization, during the last fifty years. The variants include divided demand, time windows, different fleet, etc. The VRP could be depicted as a problem of designing routes from one depot to a set of customers by satisfying constrains of demand and of capacity of the vehicles. A variant of the VRP by considering several depots, from which is possible to satisfy the demand of the customers, each one of them having different fixed cost and different capacity, is a prominent research area [4,5]. This extension of the VRP is called Location-Routing Problems (LRP). The Location-Routing Problem considers two types of problems, which have been studied independently (the Multi Depot Vehicle Routing Problem – MDVRP and the Facility Location Problem – FLP). However, recent research results showed that the integration of both decisions allows obtaining a major efficiency of the design of the supply chain [6-8]. The FLP is associated with decisions of opening depots and assigning customers to the open depots. On the other hand, the MDVRP corresponds to the development of a set of routes to be performed by minimizing the total travelled distance. The LRPH is considered NP-hard, since it is a generalization of two well-known sub-problems: The Facility Location Problem (FLP) and the Multi-Depot Vehicle Routing Problem with Heterogeneous Fleet (MDHVRP) [4].

Exact algorithms for the LRP are proposed in [9]. In this work, Bender's decomposition for splitting a LRP problem into two sub-problems (location-assignment and routing) is proposed. The drawback of linear programming approaches appears when dealing with real-life problems due to the cardinality of the scenario. Therefore, heuristic approaches are commonly used.

The heuristics algorithms for the LRP can be classified [6,7] into groups (sequential, iterative, nested) regarding to the way to deal with the FLP and the MDVRP. In the sequential approach, an algorithm for the MDVRP is performed after the FLP has been solved and, hence, there is no feedback between the two approaches. The iterative algorithm tries to take this drawback into account by iterating on the process. The nested algorithms compute the MDVRP for each solution obtained in the FLP. In general terms, sequential approaches tend to be faster than iterative and nested ones regarding the number of iterations to be performed.

One example of a sequential algorithm is the two-phase algorithm introduced in [10]. In the first phase, the FLP is solved by considering the length of the tours (routes) as a variable cost. In the second phase, a multilevel heuristic is used to solve the corresponding MDVRP. The proposed method in [11] is similar to the approach proposed by [10]. The main difference is that using a combined approach between a tabu search approach and a simulated annealing scheme solves the two phases. Another sequential algorithm based on four levels is considered in [12]. In this work, a combined problem by considering inventory decisions with a heterogeneous fleet is solved exactly by a mathematical model. Finally, a problem of transfer products from hubs to customers by considering a heterogeneous fleet is introduced in [13].

Typically, the LRP problems have considered an unlimited fleet of homogeneous vehicles. Although some considerations of heterogeneous fleet for the LRPH are proposed by [9-13], the literature regarding to location routing problems by considering heterogeneous fleet is scarce. Thus, this work is focused on the comparison of granular-based heuristic algorithm for solving the Location Routing Problem with Heterogeneous Fleet (LRPH). Indeed, the only work related to the LRPH has been introduced by Linfati et al [4]. In this paper, a modified granular tabu search for the LRPH has been proposed. The proposed algorithm is executed on instances adapted from the literature and the results are compared with a relaxed version of the proposed mathematical model for the LRPH. A remarkable fact of this algorithm is the use of the granularity concept introduced by [14], which is, in fact, a key for reducing the computational cost of the tabu search while conserving the quality of the solution. This paper aims to apply this idea to other heuristic algorithms in order to reduce the search space.

A work related to a similar problem of LRPH is presented in [15]. This paper considers a mathematical formulation and a Variable Neighborhood scheme for the Multi-Depot Vehicle Routing Problem with Heterogeneous Fleet (MDHVRP). Unlike the LRPH, the MDHVRP not considers decisions related to the depots to be opened.

In this work, we perform a computational comparison of three trajectory heuristics by using the granular concept introduced in [14] for the LRPH. The former algorithms consider the same initial solution obtained by a hybrid procedure and the same neighborhood structures. The first algorithm, called Granular Simulated Annealing approach (GSA), considers a Simulated Annealing (SA) method, with a "granular" search space, to improve the initial solution S_0 [5]. The second algorithm, called Granular Variable Neighborhood Search (GVNS), considers a Variable Neighborhood Search (VNS) procedure to enhance the quality of solution S_0. Finally, the third algorithm, called Granular Tabu Search (pGTS), consider a Probabilistic Granular Tabu Search scheme for the LRPH.

The main contribution of the paper is the comparison of effective algorithms for the solution of the LRPH. The proposed algorithms are novel metaheuristic approach, which combine different local based procedures with a granular search space for getting good results within short computing times. The paper is structured as follows. Section 2 introduces the general framework used by the considered algorithms. A detailed description of the three approaches is given in Section 3. The comparative study on adapted benchmark instances from the literature is provided in Section 4. Finally, Section 5 contains concluding remarks.

2. General framework

2.1. Mathematical formulation

The LRPH could be represented by the following undirected graph problem: Let $G = (V, E)$ be a complete undirected graph, where $V = \{1, \ldots, M + N\}$ is the set of vertices representing customers and potential depots, and $E = \{(v_i, v_j): i \neq j\}$ is the set of edges. Vertices $J = \{1 \ldots, M\}$ correspond to set of customers each with known positive demand d_j, and vertices $I = \{M + 1, \ldots, M + N\}$ correspond to the potential depots, each with capacity w_i and opening costs o_i. With each edge $(i, j) \in E$ is associated a non-negative traveling cost $\boldsymbol{c_{ij}}$. At each potential depot $i \in I$, a set of K heterogeneous vehicles is located, each with a vehicle capacityq_k. Furthermore, any vehicle that performs a route generates a fixed cosv_kt. The goal of LRPH is to determine the depots to be opened, the customers to be assigned to each open depot, and the routes to be performed to fulfill the demand of the customers by considering a heterogeneous fleet. Indeed, the LRPH integrates a strategic decision (depots to be used) and an operational decision (the routes to be developed by considering a heterogeneous fleet).

The objective function of LRPH considers the minimization of the sum of the fixed cost of the depots, of the costs of the used vehicles, and of the costs related with the distance traveled by the vehicles [4]. Any LRPH solution is feasible, if the following constraints are satisfied: (i) Each route starts and finishes at the same depot, (ii) each customer must visited exactly once by one vehicle, (iii) the sum of the demand of the customers served by a vehicle $k \in K$ must not exceed its corresponding vehicle capacity q_k, (iv) the sum of the demand of the customers assigned to an open depot $i \in I$ must not exceed its corresponding depot capacity w_i, and (v) the flow of products between depots is not allowed.

2.2. Concept of granular search space

The granular search conception (Toth and Vigo [14]), is predicated on the employment of a sparse graph (incomplete graph) containing the edges incident to the depots, the edges belonging to the best feasible solutions found so far, and the edges whose cost is smaller than a granularity threshold $\vartheta = \beta \bar{z}$, where $\bar{z} = \frac{z}{n+r}$ is the average cost of the edges belonging to the best solution found so far, and β is a *dynamic sparsification factor* which is updated during the search ([4-5] and [14]). In particular, the search starts by initializing β to a small value β_0. After $N_s \times n$ iterations the value of β is increased to the value β_n, and $N_r \times n$ additional iterations are performed by considering as current solution the best feasible solution found so far [7]. Conclusively, the *sparsification factor* β is reset to its previous value β_0 and the search continues. β_0, β_n, N_s, and N_r are given parameters. The main idea of the granularity is to obtain high quality solutions in based-trajectory heuristics within short computing times [6,7].

2.3. Neighborhood structures

The former heuristics use well-known intra-route (moves performed in a performed route) and well-known inter-route (moves performed between two routes assigned to the same depot or to different depots) moves corresponding to five neighborhood structures N_k $(k = 1, \ldots, 5)$: Insertion, Swap, Two-opt, Double-Insertion, and Double-Swap [4-7]. A move is performed only if all the new inserted edges in the solution belonging to the sparse graph (granular search space).

Some of the proposed approaches allow infeasible solutions respect to the depot and vehicle capacity constraints. For any feasible solution S, we calculate its objective function value $F_1(S)$ as the sum of the fixed cost of the depots, of the used vehicles and of the cost of the edges traveled by the performed routes. On the other hand, if the solution S is infeasible, we calculate its objective function value $F_2(S) = F_1(S) + P_q(S) + P_d(S)$, where $P_q(S)$ is a penalty term obtained by multiplying the global over vehicle capacity of the solution S times a dynamically updated penalty factor $\alpha_v = \rho_v \times F_1(S_0)$, and $P_d(S)$ is a penalty term obtained by multiplying the global over depot capacity S times a dynamically updated penalty factor $\alpha_d = \rho_d \times F_1(S_0)$. In particular, ρ_v and ρ_d are adjusted parameters during the search, and $F_1(S_0)$ is the objective function value of the initial solution S_0. If infeasible solutions with respect to the depot capacity have been not found over N_{fact} iterations, then the value of ρ_d is set to $max\{\rho_{min}, \rho_d \times \partial_{red}\}$, where $\partial_{red} < 1$. On the other hand, if feasible solutions have been not found during N_{fact} iterations, then the value of ρ_d is set to $min\{\rho_{max}, \rho_d \times \partial_{inc}\}$, where $\partial_{inc} > 1$. A similar procedure is applied to update the value of ρ_v. Note if the current solution S is feasible, $F_1(S) = F_2(S)$ (for further details see [6-7]).

2.3. Initial Solution

Utilizing a hybrid heuristic based on a cluster approach, a good feasible initial solution S_0 is generated within short computing times. The following steps are executed until all the customers are assigned to one route and one depot:

Firstly, a giant TSP tour containing all the customers (without the consideration of the depots) is constructed by using the well-known Lin-Kernighan heuristic (LKH) ([16] and [17]). Secondly, starting from any initial customer $j^{\star}$, divide the built giant TSP tour into several clusters composed of consecutive customers so that, for each cluster the vehicle capacity constraint is satisfied. In particular, we have assigned the large vehicles firstly.

Thirdly, for each depot i and each cluster g, a TSP tour is obtained using procedure LKH to evaluate the traveling cost of the route performed starting from i and visiting all the customers belonging to g (keeping the sequence obtained by the giant TSP). Finally, the depots are assigned to the clusters by solving an ILP model for the Single Source Capacitated Facility Location problem. This step determines the depots to be opened and the clusters to be assigned to the open depots (for further details see [18] and [19]).

3. Description of the considered algorithms

3.1. The Granular Simulated Annealing heuristic algorithm (GSA)

The former algorithm considers a standard implementation of the Simulated Annealing metaheuristic

(SA) by using a granular search space (reduced graph). We have assumed the same notation used on similar problems such as LRP [4]. Let $S^\star$ be the best feasible solution found so far, S the current solution (feasible or infeasible), S' a random solution obtained from the neighborhood solutions of the current solution S, α the cooling factor, and T the current temperature. Initially, we set $S^\star := S_0$, and $S := S_0$. In addition, we set the initial temperature T_0 and the number of iterations $it := 0$. The proposed algorithm performs the following steps until the number of iterations is reached:

- Decrease the current temperature every N_{cool} iterations (where N_{cool} is a given parameter) by using the following function $T_i = \propto T_{i-1}$, where $0 < \alpha < 1$; and also increase the number of iterations $i := i + 1$.
- Construct a random solution S' obtained by considering all the neighborhood structures Nk $(k = 1, \ldots, 5)$ from the current solution S described in section 2.3.
- Compute $\sigma = F_2(S') - F_2(S)$
- Generate a random number r in the range [0, 1];
- If $\sigma \leq 0$ set $S := S'$;
- If $\sigma > 0$ do If $r < exp(-\sigma/T)$, set $S := S'$; otherwise, keep S.

Finally, the best feasible solution found so far $S^\star$ is kept. The pseudocode algorithm of GSA is presented as follows:

```
Input: Initial solution S0, initial temperature T0, and max number of iterations maxIter
Output: Final solution S*
S ← S* ← S0
T ← T0
iter ← 0
While iter < maxIter do
    S' ← Nk // Generate random solution from the neighborhood
    If F2(S') < F2(S) Then
        S ← S'
        If F2(S') < F2(S*) Then S* ← S'
    Else
        r ← random(0,1) //Generate a random number
        If r < e^(−(F2(S') − F2(S))/T) Then
            S ← S'
    T ← α × T //decrease the current temperature T
    iter ← iter + 1
Return S*
```

In particular, the GSA approach requires less computing effort performing more iterations respect to the other proposed approaches.

3.2. The Granular Variable Neighborhood Search approach (GVNS)

The GVNS algorithm brings together the potentiality of the systematic changes of neighborhood structures proposed by the well-known Variable Neighborhood Search (VNS) [20], and the efficient Granular Search Space introduced by [14] and improved by [5-7]. According to [20], the VNS applies a search strategy based on the systematic change of the neighborhoods structures to elude local optima. Three main concepts are applied on a VNS: (1) All the local minimum obtained by different neighborhood structures are not necessarily equals; (2) The best local minimum (respect to the objective function) obtained from all possible neighborhood structures (described in section 2.3) is called global minimum; (3) The different local minima obtained from the neighborhood structures should be relatively close each other.

Once the initial solution is performed (S_0), the VNS approach iterates through different neighborhood structures to amend the best feasible solution ($S^\star$) found so far, until a the number of iterations is reached. The algorithm starts by setting $S^\star = S = S_0$, where S is the current solution.

The Variable Neighborhood Search considers two steps: (1) selecting a random solution from the first neighborhood and (2) applying a Granular Search Space by the same exchange operator until there is no more improvement. Then, the algorithm selects another neighborhood and the search continues.

The pseudocode algorithm of GVNS is presented as follows:

```
Input: Initial solution S0, number of neighborhoods N
Output: Final solution S*
S ← S* ← S0
iter ← 0
While iter < N do
    S' ← N_iter(S) // Generate a random solution
    S' ← LocalSearch(S') // Refine solution S'
    If z(S') < z(S*)Then
        S ← S* ← S'
        iter ← 0
    Else
        iter ← iter + 1
Return S*
```

Finally, the best feasible solution found so far $S^\star$ is kept.

3.3. A probabilistic Granular Tabu Search heuristic algorithm (pGTS)

The proposed algorithm is an extension of the proposed idea by [21] for the DCVRP. After the construction of the initial solution S_o, the pGTS algorithm iterates through different neighborhood structures (described in Section 2.3) by using a discrete probabilistic function to improve the best feasible solution (S^*) found so far, until the number of iterations is reached. The algorithm starts by setting $S^* = \bar{S} = \hat{S} = S_0$, where $\bar{S}$ is the current solution (feasible or infeasible), and $\hat{S}$ is the current feasible solution. The following steps then are repeated sequently. First, the former algorithm selects a neighborhood from the neighborhoods structures N_k $(k = 1, \ldots, 5)$ described in Section 2.3 by using the following function of probability $f(N_k) = \frac{1}{u}$, where u is the total number of neighborhoods. Second, we apply a granular tabu search (GTS) based on the idea proposed by [14] in the selected neighborhood $N_k(\bar{S})$ until a local minimum S' is found. Depending on the solution, the following choices are possible:

- Increase the probability of selecting the current neighborhood (N_k) by a given factor P_{inc} as follow $max\{f(N_k) + P_{inc}, 1\}$, only if any of three cases occurs: (1) S' is infeasible and $F_2(S') \leq F_2(\bar{S})$, (2) S' is feasible and $F_1(S') \leq F_1(\hat{S})$, and (3) S' is feasible and $F_1(S') \leq F_1(\bar{S})$. If (1) is performed, then $\bar{S} := S'$, if (2) is found, set $\hat{S} := S'$, $\bar{S} := S'$. Finally, if (3) occurs, set $\bar{S} := S'$.
- Otherwise, decrease the probability of selecting the current neighborhood (N_k) by a factor P_{dec} as follow: $min\{0.01, f(N_k) - P_{dec}\}$.

In order to preserve the probability function properties after decreasing or increasing a certain neighborhood, the values of the probability to select other neighborhoods $N_{k'}$ $(k' = 1, \ldots, 5)$, where $k' \neq k$, must be adjusted [21]. If the probability to select the neighborhood N_k *is* decreased in a P_{dec} value, the remaining value $1 - [f(N_k) - P_{dec}]$ is distributed for the remaining neighborhoods according to their current (probability) [21]. Therefore, the new probability $[f(N_{k'})]$ to select the remaining neighborhoods $(N_{k'})$ is calculated as follows:

$$f(N_{k'}) = f'(N_{k'}) * \left[\frac{1 - [f(N_k) - P_{dec}]}{1 - f(N_k)}\right] \quad (1)$$

where $f'(N_{k'})$ is the previous probability of the corresponding remaining neighborhood $(k' \neq k)$. If the probability to select the neighborhood N_k is increased in a P_{inc} value, the remaining value $1 - [f(N_k) + P_{inc}]$ is distributed for the remaining neighborhoods according to their current (probability). Therefore, the new probability $[f(N_{k'})]$ to select the remaining neighborhoods $(N_{k'})$ is calculated as:

$$f(N_{k'}) = f'(N_{k'}) * \left[\frac{1 - [f(N_k) + P_{inc}]}{1 - f(N_k)}\right] \quad (2)$$

Finally, the best feasible solution found so far S^* is kept. The algorithm explores the solution space by moving at each iteration, from a solution $\bar{S}$ to the best solution in the neighborhood $N_k(\bar{S})$, even if it is infeasible. The selected move is declared as *tabu*. The *tabu tenure* is defined as a random integer value in the range $[t_{min}, t_{max}]$, where t_{min} and t_{max} are given parameters [21].

The pseudocode algorithm of pGTS is the following:

Procedure pGTS (S_0, IT_{max})
  $\hat{S} \leftarrow S_0$
  $'S \leftarrow \hat{S}$
  $ops \leftarrow \{2opt, shift, swap, 2shift, 2swap\}$
  $props \leftarrow \{0.2, 0.2, 0.2, 0.2, 0.2\}$
  $blacklist \leftarrow \{\}$
  $op \leftarrow choose(props, ops)$
  $iterate \leftarrow true$
  While $iterate$ **do**
    $S' \leftarrow GTS('S, op, IT_{max})$
    $increase? \leftarrow false$
    If $notfeasible(S') \wedge F_2(S') < F_2('S)$ **then**
      $'S \leftarrow S'$
    If $feasible(S')$ **then**
      If $F_1(S') < F_1(\hat{S})$ **then**
        $'S \leftarrow S'$
        $\hat{S} \leftarrow' S$
        $increase? \leftarrow true$
      If $F_1(S') \leftarrow F_1('S)$ **then**
        $'S \leftarrow S'$
        $increase? \leftarrow true$
    If $increase?$ **then**
      $increase(props[op], P_{up})$
      $adjust(props)$
      $blacklist \leftarrow \{\}$
    Else
      $decrease(props[op], P_{down})$
      $adjust(props)$
      $blacklist \leftarrow blacklist \cup \{op\}$
      $op \leftarrow choose(props, ops)$
      While $op \in blacklist$ **do**
        $op \leftarrow choose(props, ops)$
    If size(blacklist) == size(ops) **then**
      $iterate \leftarrow false$
  Return $\hat{S}$

The pseudocode of the pGTS shows a brief summary of the performance of the proposed algorithm according to the discrete probability function and the replacement of the solutions depending of the characteristics of the solution found by the neighborhood S'.

4. Computational experiments

Fixing a maximum CPU time as stopping criterion has performed the comparison of the effects of the initial solution on the performance of the algorithms GSA, GVNS and pGTS. Finally, the best performance of the algorithms has been considered by executing $N_{stop} \times n$ iterations (where N_{stop} is a given parameter) for each instance. For each considered instance, the proposed approaches have been executed five times with different random generator seeds. The results reported in Tables 2 to 5 correspond for each instance to the best solution value obtained over the five runs with its corresponding total running time and the average results found within its corresponding computing time.

The three former algorithms have been coded in C++, and the computational experiments have been performed on an Intel Core Duo (only one core is used) CPU (2.00 GHz) under Linux Ubuntu 12.1 with 2 GB of memory RAM. The proposed algorithms have been tested on four benchmarking sets of instances adapted from the literature. The set of instances are available in https://github.com/maxbernal/LRPH. In all the sets, points in the plane represent the customers and the depot. Therefore, the traveling cost for an edge is calculated as the Euclidian distance between vertices.

The first three sets of instances are adapted from benchmarking instances for the CLRP proposed by [23], [24] and [25] for the CLRP respectively. In particular, for these sets of instances, the characteristics of the vehicles (fixed cost and capacities) have been modified in order to consider heterogeneous fleet for a location-routing problem. The first data subset was adapted from [23], and contains 36 instances with uncapacitated depots. The number of customers for each instance is n = 100, 150 or 200. The number of potential depots is either 10 or 20. The second data subset was adapted from [24], and considers 30 instances with capacity constraints on routes and depots. The number of customers for each instance is n = 20, 50, 100 or 200. The number of potential depots is either 5 or 10. Finally, the third data subset was adapted from [25], and considers 13 instances also with capacity constraints on depots and routes. The number of customers ranges from 21 to 150, and the number of potential depots from 5 to 10.

The fourth set is adapted from [26]. Originally, the instances from [26] are proposed for the HFVRP. In the HFVRP, all the depots are considered as opened in order to

Table 1.
Parameters used by the different algorithms

Parameter	Value	Algorithm using the parameter
T_0	100	GSA
α	0.97	GSA
ρ_{min}	1	pGTS, GSA
ρ_{max}	100	pGTS, GSA
∂_{inc}	1.1	pGTS, GSA
∂_{red}	$1 / \partial_{inc}$	pGTS, GSA
P_{inc}	0.1	pGTS, GSA
P_{dec}	0.1	pGTS, GSA
t_{min}	7	pGTS
t_{max}	49	pGTS
β_0	1.50	GSA, GVNS, pGTS
β_n	3.00	GSA, GVNS, pGTS
N_s	1	GSA, GVNS, pGTS
N_r	1	GSA, GVNS, pGTS
IT_{max}	10	GSA, GVNS, pGTS
N_{fact}	10	pGTS, GSA

Source: Owner

determine the routes to be performed. On the other hand, for the LRPH all the depots are considered as "potential". Therefore, it is mandatory to select the depots to be opened and the customers to be assigned to each open depot.

4.1. Setting of parameters

A suitable set of parameters, whose values are based on extensive computational tests on the benchmark instances, was selected for each algorithm. The parameters and values are presented in Table 1.

These values have been utilized for the comparison of the solutions obtained by the described algorithms.

The calibration of the value of each parameter was performed by a multi-objective optimization approach (considering the minimization of the objective function and the computing time). Values coming from previous works with similar algorithms ([4,5-7,21]) are used. The next step is to select a parameter and search for the given value the best result. This search is executed applying 1D function minimization. The process is carried out for each considered value. Since this process is iterative, it can be refined using more repetitions.

4.2. Comparison of the three described algorithms

For each instance, the proposed algorithms are executed 5 times due to their random calculations. Tables 2 - 5 provide the detailed results of the three proposed algorithms on the four data sets respectively. The algorithms are compared based on their best and average objective function for each instance; and the CPU for obtaining the best result and the CPU time to process the five runs of each instance. As expected, the GSA algorithm is faster, in the most of the cases, than the other algorithms since a solution is selected randomly and then evaluated; while GVNS performs local search on the selected solution, and pGTS explores the granular search space. A remarkable fact of the three algorithms is that although using random numbers, the objective function value tends to converge (see Tables 2 – 5).

The results clearly show that algorithm GSA outperforms the other two algorithms for what concerns both the CPU time and the quality of the answers found. Indeed, for all the data sets, the average costs, and the values of the best results for GSA are better than the corresponding values of algorithms GVNS and pGTS. Therefore algorithm GSA is the best performing of the three described algorithms, and, it could be compared with the most effective heuristics will publish in the literature.

5. Final remarks and future research

In this paper, a comparison of trajectory granular based algorithms for the Location Routing Problem with Heterogeneous Fleet (LRPH) is performed. All the proposed approaches use a granular search space based on the idea of using a sparse graph instead of the complete graph. Three algorithms have been proposed: Granular Simulated Annealing (GSA), Granular Variable Neighborhood Search (GVNS) and a probabilistic Granular Tabu Search (pGTS).

The computational experiments show that algorithm GSA generally obtains better results in terms of average and best results than those obtained by algorithms GVNS and pGTS. The results emphasize the importance of the granular search approach for the proposed algorithms, by showing that it significantly improves the performance and the computing time of the proposed approaches. We have compared the proposed approaches for the LRPH on four set of benchmarking instances adapted from the literature. The results show the effectiveness of GSA, imposing several best-known results within short computing times.

Acknowledgments

This work has been partially supported by Pontificia Universidad Javeriana, Cali, Integra S.A., Universidad Tecnologíca de Pereira, Universidad del Valle, and by Universidad del Bío-Bío and Universidad Andrés Bello from Chile, and FondeCYT with code 11150370. This support is gratefully acknowledged.

Table 2
Results for Tuzun-Burke Instances

Instance	BKS	INITIAL SOLUTION		GSA				GVNS				pGTS			
		Cost	CPU	Avg Cost	Avg	Best	Total	Avg Cost	Avg	Best	Total	Avg Cost	Avg	Best	Total
Tuzun_3_1	**2317.93**	2400.76	36	2334.08	43	2317.93	217	2399.93	31	2396.95	156	2392.98	70	2392.98	348
Tuzun_3_2	**2989.72**	3233.17	65	3000.94	69	2989.72	344	3188.08	48	3156.49	240	3213.96	128	3213.96	641
Tuzun_3_3	**2311.83**	2334.33	36	2312.29	45	2311.83	227	2334.21	30	2334.04	151	2334.33	50	2334.33	249
Tuzun_3_4	**2365.86**	2425.82	72	2374.32	76	2365.86	382	2417.02	55	2408.64	276	2425.82	91	2425.82	455
Tuzun_3_5	**2069.15**	2078.18	40	2070.21	45	2069.15	223	2078.02	31	2077.71	157	2073.54	67	2073.54	334
Tuzun_3_6	**2330.57**	2467.55	76	2335.94	85	2330.57	427	2441.61	56	2414.87	282	2455.48	136	2454.80	682
Tuzun_3_7	**1652.17**	1686.85	47	1658.07	57	1652.17	283	1685.29	38	1683.47	191	1672.67	93	1672.02	463
Tuzun_3_8	**1922.34**	2147.48	60	2001.41	96	1922.34	478	2123.69	63	2093.07	315	2135.57	124	2135.01	619
Tuzun_3_9	**2252.62**	2360.95	49	2258.03	54	2252.62	270	2360.26	39	2359.77	194	2360.95	69	2360.95	345
Tuzun_3_10	**2257.68**	2713.23	79	2364.05	87	2257.68	434	2654.87	60	2633.78	301	2705.17	107	2703.72	536
Tuzun_3_11	**1852.03**	1944.76	50	1873.20	56	1852.03	281	1944.43	38	1943.55	191	1940.11	77	1940.06	387
Tuzun_3_12	**2371.29**	2562.77	85	2433.94	88	2371.29	439	2558.97	62	2555.11	312	2560.53	100	2560.53	499
Tuzun_3_13	**6176.56**	6456.06	157	6256.08	179	6176.56	893	6455.98	120	6455.67	598	6442.71	968	6442.38	4840
Tuzun_3_14	**6385.58**	6536.73	299	6391.86	324	6385.58	1618	6525.14	219	6504.62	1094	6482.75	1231	6482.69	6155
Tuzun_3_15	**6214.05**	6372.13	154	6234.95	178	6214.05	891	6372.13	116	6372.13	578	6342.42	705	6340.09	3525
Tuzun_3_16	**6554.11**	6880.01	313	6624.20	341	6554.11	1705	6865.46	231	6855.05	1153	6856.53	962	6853.14	4810
Tuzun_3_17	**5947.54**	6157.28	183	5995.29	208	5947.54	1038	6156.84	133	6156.00	665	6150.17	679	6148.53	3396
Tuzun_3_18	**6114.50**	6209.83	521	6125.97	566	6114.50	2832	6200.27	372	6188.53	1861	6201.01	931	6200.70	4654
Tuzun_3_19	**5440.51**	5716.49	219	5502.74	265	5440.51	1323	5715.94	165	5715.13	827	5708.90	709	5708.33	3545
Tuzun_3_20	**6207.16**	6754.28	598	6409.92	649	6207.16	3246	6748.07	422	6742.73	2110	6748.55	869	6748.55	4347
Tuzun_3_21	**5787.07**	6081.60	186	5876.29	226	5787.07	1130	6081.42	141	6081.23	704	6071.82	833	6071.76	4164
Tuzun_3_22	**6248.68**	6521.57	419	6266.54	452	6248.68	2259	6489.98	297	6468.44	1485	6512.23	757	6512.23	3785
Tuzun_3_23	**5736.73**	5801.56	274	5764.78	319	5736.73	1593	5799.63	203	5797.53	1014	5792.12	756	5791.69	3782
Tuzun_3_24	**6732.60**	7394.55	466	6920.36	494	6732.60	2468	7341.25	333	7313.80	1663	7380.87	1253	7379.16	6266
Tuzun_3_25	**3523.67**	3801.64	76	3536.97	94	3523.67	471	3800.67	59	3798.37	293	3749.18	207	3725.38	1034
Tuzun_3_26	**3565.55**	4080.75	164	3709.04	168	3565.55	842	4052.42	122	4032.13	608	4080.75	231	4080.75	1153
Tuzun_3_27	**3514.89**	3681.68	81	3525.46	96	3514.89	479	3681.39	65	3680.22	323	3616.72	391	3616.66	1955
Tuzun_3_28	**3784.44**	4048.16	160	3854.08	166	3784.44	828	4031.95	120	4022.89	601	4026.38	277	4026.38	1384
Tuzun_3_29	**3207.38**	3271.55	99	3245.34	118	3207.38	588	3270.55	80	3269.77	399	3269.32	212	3269.32	1062
Tuzun_3_30	**3208.90**	3749.59	207	3322.80	225	3208.90	1125	3710.75	149	3674.92	744	3737.02	335	3737.02	1674
Tuzun_3_31	**2790.62**	2918.91	117	2791.99	141	2790.62	707	2917.61	89	2916.42	443	2912.39	250	2911.17	1250
Tuzun_3_32	**3006.20**	3296.47	215	3081.10	234	3006.20	1170	3284.45	157	3271.75	783	3284.83	536	3283.72	2679
Tuzun_3_33	**3208.38**	3377.91	103	3229.38	124	3208.38	618	3376.86	80	3375.62	398	3375.51	197	3375.51	987
Tuzun_3_34	**3505.24**	4448.21	190	3770.37	199	3505.24	995	4434.86	136	4420.16	678	4422.68	342	4422.68	1710
Tuzun_3_35	**2876.83**	2996.11	105	2880.62	118	2876.83	590	2994.95	81	2993.46	406	2960.24	296	2951.67	1479
Tuzun_3_36	**3739.83**	3895.81	204	3741.84	218	3739.83	1092	3857.93	149	3842.49	744	3894.80	299	3894.80	1493
Average		**4133.46**	**172**	**3946.51**	**192**	**3893.62**	**959**	**4120.91**	**127**	**4111.29**	**637**	**4119.19**	**426**	**4117.83**	**2130**

Source: Owner

Table 3.
Results for Prodhon Instances

Instance	BKS	INITIAL SOLUTION		GSA				GVNS				pGTS			
		Cost	CPU time	Avg Cost	Avg time	Best Cost	Total time	Avg Cost	Avg time	Best Cost	Total time	Avg Cost	Avg time	Best Cost	Total time
Prodhon_2_1	**22028.47**	22077.75	5	22042.60	4	22028.47	21	22076.60	7	22071.98	33	22066.16	8	22066.16	40
Prodhon_2_2	**16318.64**	16344.61	6	16322.88	5	16318.64	23	16344.61	7	16344.61	36	16331.84	7	16331.84	37
Prodhon_2_3	**23511.03**	23553.12	5	23511.03	5	23511.03	26	23552.42	6	23549.64	30	23544.35	7	23544.35	36
Prodhon_2_4	**15912.53**	15912.53	5	15912.53	5	15912.53	26	15912.53	6	15912.53	30	15912.53	8	15912.53	41
Prodhon_2_5	**16946.94**	16980.95	14	16949.43	13	16946.94	65	16980.89	15	16980.81	75	16979.26	23	16979.26	115
Prodhon_2_6	**16946.94**	16980.95	13	16949.87	14	16946.94	68	16980.93	15	16980.81	75	16979.26	22	16979.26	109
Prodhon_2_7	**31215.64**	31521.80	14	31230.69	15	31215.64	75	31521.62	16	31521.33	79	31447.33	25	31445.47	127
Prodhon_2_8	**31238.55**	31521.80	14	31250.30	15	31238.55	74	31521.75	16	31521.57	82	31447.85	26	31446.22	129
Prodhon_2_9	**18535.68**	18898.72	15	18682.36	15	18535.68	75	18898.14	16	18895.84	80	18852.91	29	18840.51	146
Prodhon_2_10	**19614.68**	19951.12	13	19745.87	14	19614.68	68	19950.87	17	19950.31	87	19931.37	23	19931.37	116
Prodhon_2_11	**12766.26**	12847.36	16	12769.76	12	12766.26	60	12847.36	17	12847.36	87	12839.82	23	12839.82	114
Prodhon_2_12	**12761.67**	12847.36	16	12766.18	12	12761.67	60	12847.36	18	12847.36	89	12839.82	22	12839.82	111
Prodhon_2_13	**143839.64**	143873.66	35	143842.28	33	143839.64	163	143868.54	40	143857.30	201	143873.66	79	143873.66	397
Prodhon_2_14	**143841.83**	143873.66	37	143843.60	32	143841.83	161	143869.93	40	143860.20	199	143873.66	79	143873.66	395
Prodhon_2_15	**99095.66**	99095.66	37	99095.66	31	99095.66	153	99095.66	41	99095.66	206	99095.66	92	99095.66	458
Prodhon_2_16	**99095.66**	99095.66	35	99095.66	30	99095.66	151	99095.66	39	99095.66	196	99095.66	90	99095.66	451
Prodhon_2_17	**96656.46**	96659.28	36	96656.46	31	96656.46	155	96658.96	39	96658.50	197	96659.28	81	96659.28	405
Prodhon_2_18	**96656.46**	96659.28	36	96656.46	31	96656.46	156	96659.04	39	96658.45	194	96659.28	86	96659.28	431
Prodhon_2_19	**163846.85**	163846.85	214	163846.85	272	163846.85	1358	163846.85	218	163846.85	1088	163846.85	329	163846.85	1647
Prodhon_2_20	**163846.85**	163846.85	217	163846.85	270	163846.85	1352	163846.85	218	163846.85	1092	163846.85	329	163846.85	1645
Prodhon_2_21	**161256.16**	161256.16	72	161256.16	81	161256.16	407	161256.16	71	161256.16	356	161256.16	142	161256.16	708
Prodhon_2_22	**161256.16**	161256.16	71	161256.16	82	161256.16	409	161256.16	71	161256.16	356	161256.16	140	161256.16	698
Prodhon_2_23	**151783.10**	151783.10	121	151783.10	146	151783.10	730	151783.10	123	151783.10	616	151783.10	205	151783.10	1027
Prodhon_2_24	**151783.10**	151783.10	117	151783.10	148	151783.10	739	151783.10	118	151783.10	589	151783.10	198	151783.10	992
Prodhon_2_25	**243999.72**	244038.02	182	244004.73	221	243999.72	1105	244036.97	190	244036.46	952	244025.39	1399	244023.25	6994
Prodhon_2_26	**243981.01**	244038.02	182	243995.08	220	243981.01	1102	244036.07	188	244032.89	941	244024.32	1351	244023.25	6753
Prodhon_2_27	**283998.66**	284018.77	223	284002.94	287	283998.66	1437	284018.46	227	284017.82	1134	284018.77	701	284018.77	3506
Prodhon_2_28	**284002.20**	284018.77	226	284004.51	289	284002.20	1444	284018.38	227	284017.94	1134	284018.77	746	284018.77	3728
Prodhon_2_29	**320250.59**	320250.59	175	320250.59	218	320250.59	1089	320250.59	180	320250.59	899	320250.59	715	320250.59	3573
Prodhon_2_30	**320250.59**	320250.59	194	320250.59	218	320250.59	1089	320250.59	179	320250.59	897	320250.59	775	320250.59	3877
Average		**118969.41**	**78**	**118920.14**	**92**	**118907.92**	**461**	**118968.87**	**80**	**118967.61**	**401**	**118959.68**	**259**	**118959.04**	**1293**

Source: Owner

Table 4 Results for Barreto Instances

Instance	BKS	INITIAL SOLUTION		GSA				GVNS				pGTS			
		Cost	CPU time	Avg Cost	Avg time	Best Cost	Total time	Avg Cost	Avg time	Best Cost	Total time	Avg Cost	Avg time	Best Cost	Total time
Prodhon_2_1	**22028.47**	22077.75	5	22042.60	4	22028.47	21	22076.60	7	22071.98	33	22066.16	8	22066.16	40
Prodhon_2_2	**16318.64**	16344.61	6	16322.88	5	16318.64	23	16344.61	7	16344.61	36	16331.84	7	16331.84	37
Prodhon_2_3	**23511.03**	23553.12	5	23511.03	5	23511.03	26	23552.42	6	23549.64	30	23544.35	7	23544.35	36
Prodhon_2_4	**15912.53**	15912.53	5	15912.53	5	15912.53	26	15912.53	6	15912.53	30	15912.53	8	15912.53	41
Prodhon_2_5	**16946.94**	16980.95	14	16949.43	13	16946.94	65	16980.89	15	16980.81	75	16979.26	23	16979.26	115
Prodhon_2_6	**16946.94**	16980.95	13	16949.87	14	16946.94	68	16980.93	15	16980.81	75	16979.26	22	16979.26	109
Prodhon_2_7	**31215.64**	31521.80	14	31230.69	15	31215.64	75	31521.62	16	31521.33	79	31447.33	25	31445.47	127
Prodhon_2_8	**31238.55**	31521.80	14	31250.30	15	31238.55	74	31521.75	16	31521.57	82	31447.85	26	31446.22	129
Prodhon_2_9	**18535.68**	18898.72	15	18682.36	15	18535.68	75	18898.14	16	18895.84	80	18852.91	29	18840.51	146
Prodhon_2_10	**19614.68**	19951.12	13	19745.87	14	19614.68	68	19950.87	17	19950.31	87	19931.37	23	19931.37	116
Prodhon_2_11	**12766.26**	12847.36	16	12769.76	12	12766.26	60	12847.36	17	12847.36	87	12839.82	23	12839.82	114
Prodhon_2_12	**12761.67**	12847.36	16	12766.18	12	12761.67	60	12847.36	18	12847.36	89	12839.82	22	12839.82	111
Prodhon_2_13	**143839.64**	143873.66	35	143842.28	33	143839.64	163	143868.54	40	143857.30	201	143873.66	79	143873.66	397
Prodhon_2_14	**143841.83**	143873.66	37	143843.60	32	143841.83	161	143869.93	40	143860.20	199	143873.66	79	143873.66	395
Prodhon_2_15	**99095.66**	99095.66	37	99095.66	31	99095.66	153	99095.66	41	99095.66	206	99095.66	92	99095.66	458
Prodhon_2_16	**99095.66**	99095.66	35	99095.66	30	99095.66	151	99095.66	39	99095.66	196	99095.66	90	99095.66	451
Prodhon_2_17	**96656.46**	96659.28	36	96656.46	31	96656.46	155	96658.96	39	96658.50	197	96659.28	81	96659.28	405
Prodhon_2_18	**96656.46**	96659.28	36	96656.46	31	96656.46	156	96659.04	39	96658.45	194	96659.28	86	96659.28	431
Prodhon_2_19	**163846.85**	163846.85	214	163846.85	272	163846.85	1358	163846.85	218	163846.85	1088	163846.85	329	163846.85	1647
Prodhon_2_20	**163846.85**	163846.85	217	163846.85	270	163846.85	1352	163846.85	218	163846.85	1092	163846.85	329	163846.85	1645
Prodhon_2_21	**161256.16**	161256.16	72	161256.16	81	161256.16	407	161256.16	71	161256.16	356	161256.16	142	161256.16	708
Prodhon_2_22	**161256.16**	161256.16	71	161256.16	82	161256.16	409	161256.16	71	161256.16	356	161256.16	140	161256.16	698
Prodhon_2_23	**151783.10**	151783.10	121	151783.10	146	151783.10	730	151783.10	123	151783.10	616	151783.10	205	151783.10	1027
Prodhon_2_24	**151783.10**	151783.10	117	151783.10	148	151783.10	739	151783.10	118	151783.10	589	151783.10	198	151783.10	992
Prodhon_2_25	**243999.72**	244038.02	182	244004.73	221	243999.72	1105	244036.97	190	244036.46	952	244025.39	1399	244023.25	6994
Prodhon_2_26	**243981.01**	244038.02	182	243995.08	220	243981.01	1102	244036.07	188	244032.89	941	244024.32	1351	244023.25	6753
Prodhon_2_27	**283998.66**	284018.77	223	284002.94	287	283998.66	1437	284018.46	227	284017.82	1134	284018.77	701	284018.77	3506
Prodhon_2_28	**284002.20**	284018.77	226	284004.51	289	284002.20	1444	284018.38	227	284017.94	1134	284018.77	746	284018.77	3728
Prodhon_2_29	**320250.59**	320250.59	175	320250.59	218	320250.59	1089	320250.59	180	320250.59	899	320250.59	715	320250.59	3573
Prodhon_2_30	**320250.59**	320250.59	194	320250.59	218	320250.59	1089	320250.59	179	320250.59	897	320250.59	775	320250.59	3877
Average		**118969.41**	**78**	**118920.14**	**92**	**118907.92**	**461**	**118968.87**	**80**	**118967.61**	**401**	**118959.68**	**259**	**118959.04**	**1293**

Source: Owner

Table 5
Results for Christofides Instances

Instance	BKS	INITIAL SOLUTION		GSA				GVNS				pGTS			
		Cost	CPU time	Avg Cost	Avg time	Best Cost	Total time	Avg Cost	Avg time	Best Cost	Total time	Avg Cost	Avg time	Best Cost	Total time
Christofides_13	**2153.21**	2505.20	15	2203.56	12	2153.21	59	2504.25	17	2500.43	87	2449.13	27	2446.08	134
Christofides_14	**7685.83**	7686.94	15	7685.83	12	7685.83	60	7686.73	18	7686.07	90	7686.94	22	7686.94	109
Christofides_15	**2822.21**	2829.77	14	2822.21	12	2822.21	60	2829.43	17	2828.91	85	2829.77	20	2829.77	100
Christofides_16	**2922.67**	2930.19	11	2922.67	11	2922.67	53	2930.02	14	2929.33	69	2923.53	19	2923.53	95
Christofides_17	**1902.57**	1920.70	17	1905.67	23	1902.57	113	1920.70	22	1920.70	111	1920.70	30	1920.70	152
Christofides_18	**2825.50**	2839.99	18	2826.69	22	2825.50	111	2839.87	22	2839.70	112	2827.94	42	2827.94	210
Christofides_19	**10218.28**	10767.52	32	10618.81	40	10218.28	200	10764.51	40	10757.15	201	10738.69	146	10733.56	730
Christofides_20	**4351.23**	4370.17	33	4351.23	40	4351.23	201	4370.17	40	4370.17	202	4361.55	133	4361.55	664
Average		**4481.31**	**19**	**4417.09**	**21**	**4360.19**	**107**	**4480.71**	**24**	**4479.06**	**120**	**4467.28**	**55**	**4466.26**	**274**

Source: Owner

References

[1] Applegate, D.L., Bixby, R.E., Chvatal, V. and Cook, W.J., The traveling salesman problem: A computational study (Princeton series in applied mathematics), Princeton, NJ, USA, Princeton University Press, 2007.

[2] Garey, M.R. and Johnson, D.S., Computers and Intractability; A guide to the theory of NP-Completeness, New York, NY, USA, W.H. Freeman & Co., 1990.

[3] Fortnow, L., The status of the P versus NP problem, Commun. ACM, 52(9), pp. 78-86, 2009. DOI: 10.1145/1562164.1562186

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

[5] Escobar, J.W., Heuristic algorithms for the capacitated location-routing problem and the multi-depot vehicle routing problem, 4OR, 12(1), pp. 99-100, 2014. DOI: 10.1007/s10288-013-0241-4.

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

[7] Escobar, J.W., Linfati, R., Baldoquin, M.G. and Toth, P., A granular variable tabu neighborhood search for the capacitated location-routing problem, Transportation Research Part B: Methodological, 67, pp. 344-356, 2014. DOI: 10.1016/j.trb.2014.05.014.

[8] Escobar, J.W., Linfati, R. and Adarme-Jaimes, W., A hybrid metaheuristic algorithm for the capacitated location routing problem, DYNA, 82(189), 243-251, 2015. DOI: 10.15446/dyna.v82n189.48552.

[9] Bookbinder, J.H. and Reece, K.E., Vehicle routing considerations in distribution system design, European Journal of Operational Research, 37(2), pp. 204-213, 1988. DOI: 10.1016/0377-2217(88)90330-X.

[10] Salhi, S. and Fraser, M., An integrated heuristic approach for the combined location vehicle fleet mix problem, Studies in Locational Analysis, 8, pp. 3-21, 1996.

[11] Wu, T.H., Low, C. and Bai, J.W., Heuristic solutions to multi-depot location-routing problems, Computers & Operations Research, 29(10) pp. 1393-1415, 2002. DOI: 10.1016/S0305-0548(01)00038-7.

[12] Ambrosino, D. and Grazia, M., Distribution network design: New problems and related models, European Journal of Operational Research, 165(3), pp. 610-624, 2005. DOI: 10.1016/j.ejor.2003.04.009.

[13] Gunnarsson, H., Rönnqvist, M. and Carlsson, D., A combined terminal location and ship routing problem, Journal of the Operational Research Society, 57(8), pp. 928-938, 2005. DOI: 10.1057/palgrave.jors.2602057.

[14] Toth, P. and Vigo, D., The granular tabu search and its application to the vehicle-routing problem, INFORMS Journal on Computing, 15(4), pp. 333-346, 2003. DOI: 10.1287/ijoc.15.4.333.24890.
[15] Salhi, S., Imran, A. and Wassan, N.A., The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation, Computers & Operations Research, 52(1), pp. 315-325, 2014. DOI: 10.1016/j.cor.2013.05.011.
[16] Lin, S. and Kernighan, B., An effective heuristic algorithm for the traveling-salesman problem, Operations Research, 21(2), pp. 498-516, 1973. DOI: 10.1287/opre.21.2.498.
[17] Helsgaun, K., An effective implementation of the Lin-Kernighan traveling salesman heuristic, European Journal of Operational Research, 126(1), pp. 106-130, 2000. DOI: 10.1016/S0377-2217(99)00284-2.
[18] Barcelo, J. and Casanovas, J., A heuristic Lagrangean algorithm for the capacitated plant location problem, European Journal of Operational Research, 15(2), pp. 212-226, 1984. DOI: 10.1016/0377-2217(84)90211-X.
[19] Klincewicz, J. and Luss, H., A Lagrangian relaxation heuristic for capacitated facility location with single-source constraints, Journal of the Operational Research Society, 37(5), pp. 495-500, 1986. DOI: 10.2307/2582672.
[20] Mladenović, N. and Hansen, P., Variable neighborhood search. Computers & Operations Research, 24(11), pp. 1097-1100, 1997. DOI: 10.1016/S0305-0548(97)00031-2.
[21] Bernal, J., Escobar, J.M, Paz, J.C., Linfati, R. and Gatica, G., A probabilistic granular tabu search for the distance constrained capacitated vehicle routing problem, Int. J. Industrial and Systems Engineering, In press, 2016.
[22] Gendreau, M., Hertz, A. and Laporte, G., A tabu search heuristic for the vehicle routing problem, Management Science, 40(10), pp. 1276-1290, 1994. DOI: 10.1287/mnsc.40.10.1276.
[23] Tuzun, D. and Burke, L., A two-phase tabu search approach to the location routing problem, European Journal of Operational Research, 116(1), pp. 87-99. DOI: 10.1016/S0377-2217(98)00107-6
[24] Prins, C., Prodhon, C. and Wolfler-Calvo, R., Nouveaux algorithmes pour le problème de localisation et routage sous contraintes de capacité, MOSIM, 4(2), pp. 1115-1122, 2004.
[25] Barreto, S., Análise e Modelização de Problemas de localização-distribuição. PhD Thesis, University of Aveiro, Aveiro, Portugal, 2004, 340 P.
[26] Golden, B., Assad, A., Levy, L. and Gheysens, F., The fleet size and mix vehicle routing problem, Computers & Operations Research, 11(1), pp. 49-66, 1984. DOI: 10.1016/0305-0548(84)90007-8.

J.A. Bernal-Moyano, received his BSc. Eng in Systems Engineering in 2014 from the Universidad del Valle, Cali, Colombia. Currently, he is enrolled on an Erasmus Mundus Master in Computer Vision and Robotics (VIBOT) at Heriot-Watt University.
ORCID: 0000-0003-3167-5134

J. W. Escobar, is PhD in Operations Research from University of Bologna, Italy. Currently, works as full time professor in Pontificia Universidad Javeriana Cali, Colombia.
ORCID: 0000-0001-6175-9553

C. Marín-Moreno, PhD. candidate in Engineering from Universidad Tecnologíca de Pereira, Colombia. Currently, Cesar works as manager in Integra S.A.
ORCID: 0000-0002-7354-7838

R. Linfati, is PhD. in Operations Research from University of Bologna, Italy. Currently, professor Linfati works as full time professor in Universidad del Bio-Bio, Chile.
ORCID: 0000-0002-7659-383X

G. Gatica, is PhD. in Engineering from Universidad Santiago de Chile, Chile. Currently, Professor Gatica works as full time professor in Universidad Andres Bello, Chile.
ORCID: 0000-0002-1816-6856