A probabilistic granular tabu search for the distance constrained capacitated vehicle routing problem

Jose Bernal

Escuela de Ingeniería de Sistemas,
Universidad del Valle,
Cali 76001000, Valle del Cauca, Colombia
Email: jose.bernal@correounivalle.edu.co

John Willmer Escobar* and Juan Camilo Paz

Department of Civil and Industrial Engineering,
Pontificia Universidad Javeriana Cali,
Cali 76001000, Valle del Cauca, Colombia
Email: jwescobar@javerianacali.edu.co
Email: juan_camilo_paz_roa@javerianacali.edu.co
*Corresponding author

Rodrigo Linfati

Departamento de Ingeniería Industrial,
Universidad del Bío-Bío,
Concepción 4051381, Chile
Email: rlinfati@ubiobio.cl

Gustavo Gatica

Facultad de Ingeniería,
Universidad Andres Bello,
Santiago de Chile 8320000, Chile
Email: ggatica@unab.cl

Abstract: We address the well-known distance constrained capacitated vehicle routing problem (DCVRP) by considering Euclidean distances, in which the aim is to determine the routes to be performed to fulfil the demand of the customers by using a homogeneous fleet. The objective is to minimise the sum of the variable costs associated with the distance travelled by the performed routes. In this paper, we propose a metaheuristic algorithm based on a probabilistic granular tabu search (pGTS) by considering different neighbourhoods. In particular, the proposed algorithm selects a neighbourhood by using a probabilistic discrete function, which is modified dynamically during the search by favouring the moves that have improved the best solution found so far. A shaking procedure is applied whenever the best solution found so far is not improved for a given number of iterations. Computational experiments on benchmark instances taken from the literature show that the proposed approach is able to obtain high quality solutions, within short computing times.

Copyright © 2018 Inderscience Enterprises Ltd.

Keywords: probabilistic granular tabu search; pGTS; distance constrained capacitated vehicle routing problem; DCVRP; metaheuristic algorithms; vehicle routing problems; VRPs.

Reference to this paper should be made as follows: Bernal, J., Escobar, J.W., Paz, J.C., Linfati, R. and Gatica, G. (2018) 'A probabilistic granular tabu search for the distance constrained capacitated vehicle routing problem', *Int. J. Industrial and Systems Engineering*, Vol. 29, No. 4, pp.453–477.

Biographical notes: Jose Bernal is a student researcher of System Engineering from University of Valle, Cali Colombia. Currently, he is studying his Master degree on Heriot-Watt University Erasmus Mundus Master in Vision and Robotics.

John Willmer Escobar is a full-time Professor of Pontificia Universidad Javeriana Cali, Colombia. He has a PhD in Operations Research from University of Bologna, Italy. Currently, he is working on combinatorial and stochastic problems seeking applications in logistic and financial issues.

Juan Camilo Paz is an Industrial Engineer from University of Valle, Cali Colombia. Currently, he is studying his Master degree on Engineering in Pontificia Universidad Javeriana Cali. He is a part-time Professor of Pontificia Universidad Javeriana Cali.

Rodrigo Linfati is an Assistant Professor of University of Bio-Bio, Chile. He has a PhD in Operations Research from University of Bologna, Italy. Currently, he is working on combinatorial and stochastic problems seeking applications in logistic and financial issues. Currently, he is the main researcher of project FONDECYT 11150370.

Gustavo Gatica is a full-time Professor of Universidad Andres Bello, Chile. He has a PhD in Engineering from Universidad Santiago de Chile (USACH), Chile. Currently, he is working on combinatorial and stochastic problems seeking applications in logistic and financial issues.

1 Introduction

The vehicle routing problem (VRP) has been studied intensively in the disciplines of the operations research and combinatorial optimisation. The VRP can be described as the problem of designing routes, from one depot to a set of customers by satisfying constrains of demand, of vehicle capacity and duration of routes. The VRP has several applications in real world logistic systems by studying the distribution of products or services to different types of customers (Toth and Vigo, 2014; Golden et al., 2008; Clarke and Wright, 1964). This paper considers the variant of the VRP with capacity and length of route (duration) constraints [distance constrained capacitated vehicle routing problem (DCVRP)] and Euclidean distances.

The DCVRP could be represented as the following graph problem: Let $G = (V, E)$ be a complete undirected graph, where $V = \{v_0, v_1,\dots,v_n\}$ is the set of vertices representing customers with the depot located at vertex v_0, and $E = \{(v_i, v_j): i \neq j\}$ is the set of edges.

With each edge $(v_i, v_j) \in E$ is associated a non-negative travelling cost c_{ij}. In particular, we consider the symmetric version of the DCVRP ($c_{ij} = c_{ji}$). Each customer $j \in V \setminus \{v_0\}$ has a non-negative demand d_j, which must be satisfied from the depot. A homogeneous set of vehicles, each one having a capacity Q, is available at the depot v_0. The goal of the DCVRP is determine the routes to be performed for satisfying the demand of the customers by considering the minimum total travelling cost. The following constrains are imposed:

- each route must start and end at the depot
- each customer $j \in V \setminus \{v_0\}$ is visited exactly once by exactly one vehicle
- the sum of the demand of customers belonging to each route must not exceed the vehicle capacity Q
- the total duration of each route (given by the sum of the travelling costs of the traversed edges and of the service times of the visited customers) must not exceed a given value D
- the number of performed routes must not exceed the number of vehicles.
- every customer $j \in V \setminus \{v_0\}$ requires a service time $\delta_i (\delta_{v_0} = 0)$.

Several approaches for the DCVRP have been proposed during the last decades (Toth and Vigo, 2014; Golden et al., 2008). These approaches consider optimisation schemes for solving small and medium size instances, and several heuristic and metaheuristic algorithms for large-size instances of the DCVRP. One of the most cited algorithms for the capacitated vehicle routing problem (CVRP), which is a variant of the DCVRP without the consideration of duration of route constraints, has been proposed by Clarke and Wright (1964). Other approaches, based on the idea introduced in Clarke and Wright (1964), have been proposed by Holmes and Parker (1976), Beasley (1981), Dror and Trudeau (1986) and Paessens (1988). In Holmes and Parker (1976), a procedure based on the Clark and Wright (CW) approach, with a perturbation procedure scheme that is applied when the algorithm remains in a local optimum for a given number of iterations, is introduced. In Beasley (1981), an adaptation of the CW is proposed to solve the CVRP with travel times inter customers. A probabilistic version of the CW algorithm for the stochastic CVRP is presented in Dror and Trudeau (1986). Finally, a review of the main strengths of CW algorithms and their performances for VRP problems is proposed in Paessens (1988).

Other well-known constructive approach for the CVRP has been proposed by Gillett and Miller (1974). A GRASP approaches to solve the DCVRP have been proposed in Feo and Resende (1989, 1995). A tabu search metaheuristic for the DCVRP has been proposed by Gendreau et al. (1994). In this work, a tabu route algorithm with different local search procedures is proposed. An improved version of the tabu search algorithm is introduced in Toth and Vigo (2003). This version considers a granular search space, which allows reducing the computing time of a traditional tabu search but keeping the same quality result.

Evolutionary algorithms for the CVRP have been proposed by Alba and Dorronsoro (2004), Berger and Barkaoui (2003) and Prins (2004). A metaheuristic algorithm for the DCVRP has been proposed by Franceschi et al. (2005). In this work, an approach for TSP is extended to solve the DCVRP. The proposed method involves the extraction of nodes in order to generate and relocate new sequences by using integer linear programming (ILP). More recently, several approaches (Kara, 2011; Zhou et al., 2013; Bouzid et al., 2016; Nagarajan and Ravi, 2011) and new benchmark instances (Uchoa et al., 2014) have been proposed for the DCVRP.

The main body of the proposed algorithm considers two parts:

1 the construction of an initial solution by using a hybrid approach

2 a probabilistic granular tabu search (pGTS) procedure.

A similar granular tabu search (GTS) for the CVRP has been proposed by Toth and Vigo (2003). The main differences of the proposed algorithm with respect to the work presented in Toth and Vigo (2003) for the DCVRP are

1 the procedure for the construction of the initial solution

2 the penalty diversification scheme of the infeasible solutions during the search

3 the intensification and diversification process during the search

4 the criteria used to select and to accept a move

5 the shaking procedure used to avoid the algorithm remains in a local optimum for a given number of iterations

6 the benchmarking sets considered for the computational experiments.

The paper is organised as follows. In Section 2, we give the detailed description of the proposed algorithm. Although some procedures are similar to the corresponding ones presented in Toth and Vigo (2003), for the sake of clarity we prefer to give all the details of the procedures used in the proposed algorithm. Experimental results on the benchmark instances from the literature are reported in Section 3. Finally, conclusions and future research are presented in Section 4.

2 General framework of the proposed algorithm

The proposed algorithm (pGTS) consists of two parts

1 the construction of an initial solution by using a modified procedure proposed in Clarke and Wright (1964)

2 the pGTS approach.

In the following sections, the two parts are detailed.

2.1 Initial solution

The initial solution S_o is built by using a modified Clark and Wright procedure introduced in Clarke and Wright (1964) and modified by Doyuran and Catay (2010). This approach is used to obtain good initial solutions within short computing times. Different of the proposed in Clarke and Wright (1964), the approach uses the following function to calculate the value of saving between a pair of customers (i, j):

$$S_{ij} = \frac{c_{i0} + c_{i0} - \lambda c_{ij}}{c_{\max}} + \mu \cos\left(\theta_{ij}\right) \frac{\left|c_{\max} - \left(c_{i0} - c_{0j}\right)/2\right|}{c_{\max}} + v \frac{\left|\overline{d} - \left(d_i + d_j\right)/2\right|}{d_{\max}} \tag{1}$$

where c_{ij} is the distance between a pair of nodes i and j (note that 0 is the depot node), $c_{\max}$ is the maximum distances between two nodes of the complete graph, θ_{ij} is the angle between the two rays determined from the depot to nodes i and j, d_i is the demand of node i, $\overline{d}$ is the average demand, and $d_{\max}$ is the maximum demand of a node of the complete graph. λ, μ and v are given parameters, and their values could change regarding to the specific conditions of each set of benchmarking.

The procedure considers a reallocation of the vehicles to groups of customers according to the total demand, durations of the routes and service time constraints. In particular, once the savings between each pair of customers is calculated by equation (1), they are stored in a list and sorted out decreasingly. Then, each saving is evaluated in terms of demand, duration and service time constraints. If all the constraints are satisfied, the saving is to be considered. Otherwise, it is rejected. The procedure is carried out until all savings have been taken into account. Algorithm 1 shows the pseudocode for obtaining an initial solution by the modified Clarke and Wright procedure.

Algorithm 1 Modified Clarke and Wright algorithm

```
Procedure MCW (customers, distances, vehicles)
  Calculate saving list by (1)
  Order saving list decreasing
  For each saving in saving_list do
    Evaluate demand, total route length and service time if saving is applied
  If all constraints are fulfilled then
      Apply saving
      Calculate demand, total route length and service time
  Else
      Ignore saving
```

The initial solution could be classified as greedy approach because all routes are built through the continuous search of local minima. The initial solution approach allows obtaining more routes than available vehicles at depot, i.e., the constraint related to the maximum number of routes is relaxed. The extra routes are eliminated during the probabilistic granular tabu phase (pGTS).

2.2 Granular search space

The proposed algorithm (pGTS) uses the idea of granular search space introduced by Toth and Vigo (2003), which is based on the utilisation of a sparse graph instead of the complete graph. The sparse graph contains the edges incident to the depots, the edges belonging to the best solutions found during the search and the edges for which the travelling cost is less than a granular threshold value $\vartheta = \beta \overline{z}$; where $\overline{z} = \frac{z}{n + r}$ is the average cost of the best feasible solution found, z is the objective function value, n is the number of customers, r is the number of routes, and β is a sparsification parameter. β is dynamically updated during the search. The modification of the value of β allows to the algorithm alternate between intensification stages (high values of β) and diversification stages (small values of β). The main goal of the GTS is to find high quality solutions by keeping the main characteristics of the original tabu search within short computing times. Successful algorithms based on the idea of granularity for solving different variations of the VRP have been proposed by Escobar et al. (2013, 2014a, 2014b), Escobar (2013), Linfati et al. (2014a, 2014b), Puenayán et al. (2014) and Escobar and Linfati (2012).

2.3 Neighbourhood structures

Algorithm pGTS allows infeasible solutions respect to the capacity of the vehicles and the duration of the routes. Given a feasible solution S composed by a set of z routes $(r_1,\ldots,r_z)$ with each route r_l, where $l \in \{1,\ldots,z\}$, is denoted by $(v_0, v_1, v_2,\ldots, v_0)$. In addition, let us denote with $v \in r_l$ with a customer v belonging to route r_l, and with $(u, v) \in r_l$ an edge such that u and v are two consecutive vertices of route r_l. We assign to S an objective function value $F_1(s) = \sum_{l=1}^{z} \sum_{(u,v)\in r_l} c_{uv}$. If the solution S is infeasible, we assign S an 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 α_q, and $P_d(S)$ is a penalty term obtained by multiplying the global over duration of the routes performed in S times a dynamically updated penalty factor α_r. In particular, $\alpha_q = \rho_q \times F_1(S_0)$ and $\alpha_d = \rho_d \times F_1(S_0)$, where ρ_q and ρ_d are calculated parameters during the search. Note that the current solution (S) is feasible, $F_1(S) = F_2(S)$.

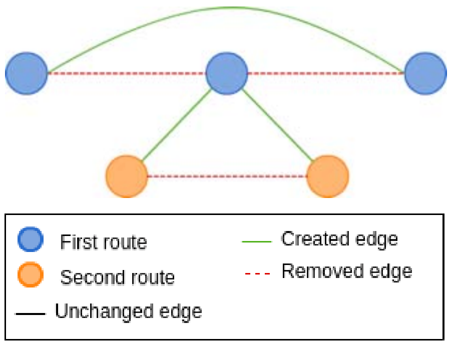
If during the search, we have found infeasible solutions respect to the vehicle capacity for a given N_{fact} iterations, the value of ρ_q is set to $\min\{\rho_{\max}, \rho_d \times \delta_{inc}\}$, where $\delta_{inc} > 1$ Otherwise, if any feasible solution has been found for a given N_{fact} iterations, the value of ρ_q is set to $\max\{\rho_{\min}, \rho_d \times \delta_{red}\}$, where $\delta_{red} < 1$. A similar approach is used to calculate the value of ρ_d during the search. N_{fact}, $\rho_{\min}$, $\rho_{\max}$, δ_{inc} and δ_{red} are given parameters.

Note that it is useful to relate the value of α_q and α_d to the value of $F_1(S_0)$ because the order of magnitude of these values by considering different set of benchmarking instances. Indeed, one of the main differences of the proposed algorithm with respect to that proposed in Toth and Vigo (2003), is that we consider dynamically updated penalty factors related with the objective function value of S_o instead of using penalty factors defined within a fixed interval. We have experimentally proved that the utilisation of this penalty scheme generally produces excellent solutions for determined benchmarking sets within short computing times.

The proposed algorithm uses intra-routes and inter-routes moves corresponding to the following neighbourhoods:

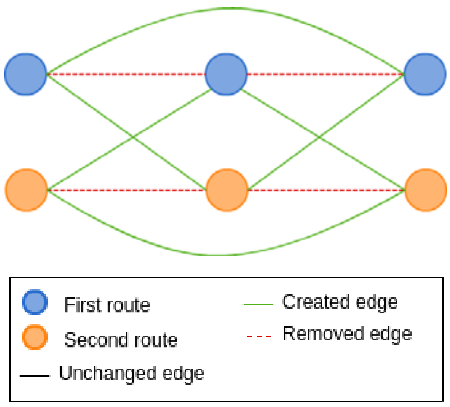
- *Shift:* A customer is removed from its current position and inserted either in a different position of the same route or in a different route. Figure 1 shows the shift neighbourhood.

Figure 1 Example of shift move (see online version for colours)



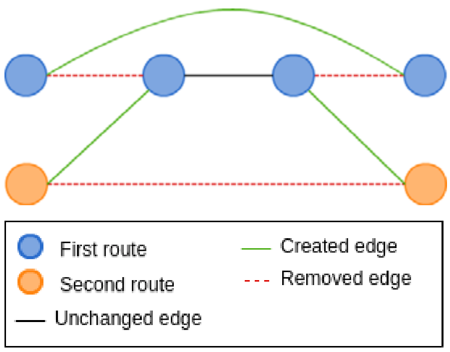
- *Swap:* Two customers (in the same route or in different routes) exchange their position. Figure 2 shows the swap neighbourhood.

Figure 2 Example of swap move (see online version for colours)



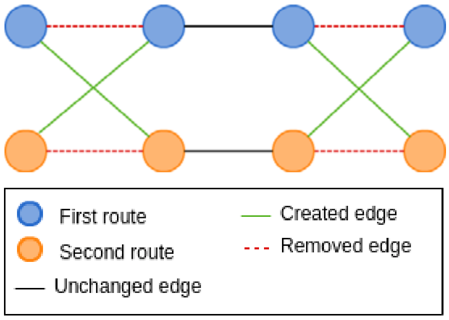
- *Double-insertion:* Two consecutive customers are removed from their current position and inserted in the same route or in a different route by keeping the edge connecting them. Figure 3 shows the double insertion neighbourhood.

Figure 3 Example of double-insertion move (see online version for colours)



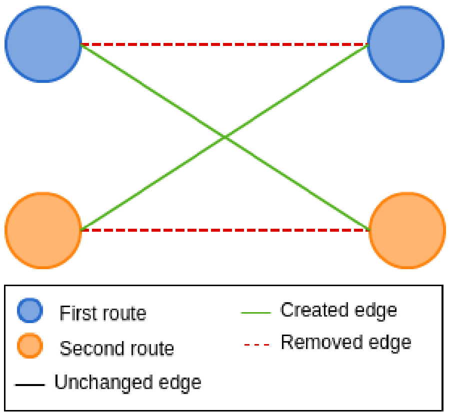
- *Double-swap:* This move is an extension of the swap move obtained by considering two pairs of consecutive customers. The edge connecting each pair of customers is kept. The move is performed between pairs of customers in two different routes.

Figure 4 Example of double-swap move (see online version for colours)



- *Two-opt:* we use the well-known two-opt move used for the CVRP (intra-route and inter-route moves).

Figure 5 Example of two-opt move (see online version for colours)



A move is performed only if the new edges to be inserted in the current solution belonging to the granular search space.

2.4 *Description of the proposed algorithm (pGTS)*

This section presents the discussion of the proposed algorithm for the DCVRP. In particular, after the construction of an initial solution S_o (by the procedure described in Section 2.1, pGTS iterates through different neighbourhood structures by using a discrete probabilistic function to improve the best feasible solution (S^*) found so far, until a stopping criterion is reached (number of iterations or computing time). 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 until a stopping criterion ($IT_{\max}$ iterations) is reached:

1 Select a neighbourhood from the neighbourhoods structures N_k ($k = 1,\ldots,5$) described in Section 2.3 by using a discrete function of probability $f(N_k) = \frac{1}{t}$, where t is the total number of neighbourhoods.

2 Apply a GTS in the selected neighbourhood $N_k(\overline{S})$ until a local minimum S' is found.

3 If S' is infeasible and $F_2(S') \leq F_2(\overline{S})$, set $\overline{S} := S'$ and increase the probability of selecting the current neighbourhood (N_k) by a factor P_{up}. Therefore, the probability to select the current neighbourhood is calculated as $\max\{f(N_k) + P_{up}, 1\}$.

4 If S' is feasible and $F_1(S') \leq F_1(\hat{S})$, set $\hat{S} := S', \overline{S} := S'$ and increase the probability of selecting the current neighbourhood by a factor P_{up}. Therefore, the probability to select the current neighbourhood is calculated as $\max\{f(N_k) + P_{up}, 1\}$.

5 If S' is feasible and $F_1(S') \leq F_1(\overline{S})$, set $\overline{S} := S'$ and increase the probability of selecting the current neighbourhood by a factor P_{up}. Therefore, the probability to select the current neighbourhood is calculated as $\max\{f(N_k) + P_{up}, 1\}$.

6 Otherwise, decrease the probability of selecting the current neighbourhood by a factor P_{down}. Therefore, the probability to select the current neighbourhood is calculated as $\min\{0.01, f(N_k) - P_{down}\}$.

In order to preserve the probability function properties after decreasing or increasing a certain neighbourhood, the values of the probability to select other neighbourhoods $N_{k'}$ $(k' = 1,\ldots,5)$, where $k' \neq k$, are adjusted as follows:

- If the probability to select the neighbourhood N_k is decreased in a P_{down} value, the remaining value $1 - [f(N_k) - P_{down}]$ is distributed for the remaining neighbourhoods according to their current (probability). Therefore, the new probability $(f(N_{k'}))$ to select the remaining neighbourhoods $(N_{k'})$ is calculated as

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

- If the probability to select the neighbourhood N_k is increased in a P_{up} value, the remaining value $1 - [f(N_k) + P_{up}]$ is distributed for the remaining neighbourhoods according to their current (probability). Therefore, the new probability $(f(N_{k'}))$ to select the remaining neighbourhoods $(N_{k'})$ is calculated as

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

where $f'(N_{k'})$ is the previous probability of the corresponding remaining neighbourhood $(k' \neq k)$.

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 $\overline{S}$ to the best solution in the neighbourhood $N_k(\overline{S})$, 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 (Toth and Vigo, 2003). The algorithm 2 shows the general scheme of the proposed approach.

The diversification strategy is based on the granular idea proposed by Toth and Vigo (2003). Initially, the sparsification factor β is set to small value β_0. If $\overline{S}$ is infeasible after N_{beta} iterations, the sparsification factor β is increased to a value β_d. Then, a new graph is calculated, and N_{change} iterations are performed from the best feasible solution found (S^*). Finally, if $\overline{S}$ is feasible after N_{change} iterations, the sparsification factor is reset from its original value β_0 and the search continues. β_0, β_d, N_{beta} and N_{change} are given parameters.

Finally, whenever the proposed algorithm remains in a local minimum for N_{shake} iterations (where N_{shake} is a given parameter), we apply a shaking procedure which extends the idea of the shift move by considering three random routes at the same time (for further details, see Escobar et al., 2013, 2014a, 2014b). Algorithm 2 shows the proposed pGTS phase.

2.5 Shaking procedure

The proposed approach selects three routes. The first route ($k1$) is selected randomly. The second route ($k2$) is the nearest neighbourhood of the route $k1$, and the route ($k3$) is the nearest neighbourhood of ($k2$). The distance between routes is calculated by considering the centre of gravity.

Then, the procedure selects randomly a customer $i1$ from the route $k1$, a customer $i2$ from the route $k2$ and edge $(h2, j2)$ from the route $k2$ (with $h2 \neq i2$ and $j2 \neq i2$), and an edge $(h3, j3)$ from the route $k3$. Therefore, the new solution S is obtained by considering the following moves:

1. remove customer $i1$ from the route $k1$ and insert it between vertices $h2$ and $j2$ in the route $k2$
2. remove customer $i2$ from the route $k2$ and insert it between vertices $h3$ and $j3$ in the route $k3$.

The perturbation procedure allows exploring new regions of the search space. Algorithm 3 shows the complete algorithm.

Algorithm 2 GTS algorithm

Procedure GTS (S_0, operator, $IT_{\max}$)

$S' \leftarrow S_0$

$TL \leftarrow \{\}$

$it \leftarrow 0$

While $it < IT_{\max}$ **do**

$S_i \leftarrow S_{i-1}$

$z' \leftarrow \infty$

For all $S \in N(S_{i-1}, operator)$ **do**

If $F_s(S) < z'$ AND S $\notin$ TL **then**

$S_i \notin S$

$z' \leftarrow F_s(S)$

$TL \leftarrow TL \cup \{S_i\}$

If $z' \leftarrow F_2(S')$ **then**

$S' \leftarrow S_i$

$it \leftarrow it + 1$

If it % $N_{shake} == 0$ **then**

$shake(S_i)$

Return S'

Algorithm 3 pGTS algorithm

Procedure pGTS (S_0, $IT_{\max}$)

$\hat{S} \leftarrow S_0$

$'S \leftarrow \hat{S}$

ops $\leftarrow$ {2*opt*, *shift*, *swap*, 2*shift*, 2*swap*}

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 *not feasible* (S') *AND* $F_2(S') < F_2('S)$ **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 ← { }
Else
decrease (*props*[*op*], P_{down})
adjust (*props*)
blacklist ← *blacklist* ∪ {*op*}
op ← *choose* (*props*, *ops*)
While *op* ∈ *blacklist* **do**
op ← *choose* (*props*, *ops*)
If *size*(*blacklist*) = = *size*(*ops*) **then**
iterate ← *false*
Return $\hat{S}$

3 Computational experiments

The proposed algorithm (pGTS) has been coded in C++ and the computational experiments have been conducted on an Intel Core Duo CPU (2.00 Ghz) under Linux Ubuntu 12.1 with 2 GB of memory RAM. The proposed algorithm has been tested on four benchmarking sets of instances proposed in Uchoa et al. (2014), Christofides et al. (1979), Golden et al. (1998), Li et al. (2005) and Rochat and Taillard (1995). In all the sets, points in the plane represent the customers and the depot. Therefore, the travelling cost for an edge is calculated as the Euclidian distance between vertices. The detailed description of all the benchmarking sets is presented by Uchoa et al. (2014).

3.1 Setting of parameters

In particular, for each instance five runs of the proposed algorithm are executed. The average and the best results with their corresponding computing times are reported in Table 1. The following set of parameters have been set for all the instances: $N_{fact} = 10$, $\rho_{\min} = 1$, $\rho_{\max} = 100$, $\delta_{inc} = 1.1$, $\delta_{red} = \frac{1}{1.1}$, $P_{up} = 0.1$, $P_{down} = 0.1$, $t_{\min} = 7$, $t_{\max} = 49$, $\beta_0 = 1.50$, $\beta_d = 3$, $N_{beta} = 1$, $N_{change} = 1$, $N_{shake} = 0.20n$, and $IT_{\max} = 10n$. In addition, for the initial solution procedure, the following parameters have been considered $\lambda = 1$, $\mu = 0$ and $\nu = 1$.

3.2 Comparative analysis

The proposed algorithm has been compared with the best-known solutions reported in the literature. The results are shown as appear in the published papers. In Tables 1–5, the following notation is used:

- *Instance*: instance number
- *n*: number of customers
- *k*: maximum number of vehicles available at depot
- *D*: maximum duration of each route
- *Q*: capacity of each vehicle
- *BKS*: cost of the best-known solution found by the previous algorithms proposed for the DCVRP
- *Ref. BKS*: reference to the algorithm, which obtained for the first time the value BKS: PISM (Taillard, 1993), RT (Rochat and Taillard, 1995), TS (Gendreau et al., 1994), MB (Toklu et al., 2014), CGL (Cordeau et al., 1997), GTS (Toth and Vigo, 2003), PA (Groër et al., 2011), VCGLR (Vidal et al., 2012), CP (Jin et al., 2014), TK (Tarantilis and Kiranoudis, 2002), NB (Nagata and Bräysy, 2009), IBCP (Pecin et al., 2016), MBE (Mester and Bräysy, 2005), LGW (Li et al., 2005), MACS (Toklu et al., 2014), UPSV, ILS-SP and UHGS (Uchoa et al., 2014)
- *Cost*: cost found by the initial solution of the proposed approach
- *Avg. cost*: average solution cost found by the corresponding algorithm over the executed runs
- *Best cost*: best solution cost found by the corresponding algorithm over the executed runs
- *Gap BKS*: best percentage gap of the initial solution cost found by the corresponding algorithm over the executed runs with respect to BKS
- *Gap avg. BKS*: percentage gap of the average solution cost found by the corresponding algorithm over the executed runs with respect to BKS
- *Gap best BKS*: percentage gap of the best solution cost found by the corresponding algorithm over the executed runs with respect to BKS
- *Time*: running time of the initial solution, expressed in seconds of the CPU used by the corresponding algorithm
- *Avg. time*: average running time over the executed runs, expressed in seconds of the CPU used by the corresponding algorithm
- *Total time*: total running time over the executed runs, expressed in seconds of the CPU used by the corresponding algorithm.

Table 1 Solutions (CPU times) obtained by the pGTS algorithm on benchmarking instances proposed by Christofides et al. (1979)

Characteristics of the instances					*Previous solutions*		*Phase 1: initial solution*			*Phase 2: probabilistic granular tabu search*					
Instance	*n*	*k*	*D*	*Q*	*BKS*	*Ref BKS*	*Cost*	*Gap BKS*	*Time*	*Avg. cost*	*Gap avg. BKS*	*Avg. time*	*Best cost*	*Gap best BKS*	*Total time*
CMT1	50	5	∞	160	*524.61**	PISM	584.64	11.44	0.16	571.43	8.92	2.31	571.43	8.92	12.32
CMT2	75	10	∞	140	*835.26**	RT	907.39	8.64	0.24	901.65	7.95	4.25	901.65	7.95	22.61
CMT3	100	8	∞	200	*826.14**	TS	889.00	7.61	0.24	872.49	5.61	3.03	872.49	5.61	17.46
CMT4	150	12	∞	200	*1,028.42**	RT	1,140.42	10.89	0.45	1,136.88	10.55	5.65	1,136.88	10.55	33.74
CMT5	199	17	∞	200	*1,291.29**	MB	1,395.74	8.09	0.74	1,392.53	7.84	13.00	1,392.53	7.84	77.24
CMT6	50	6	200	160	*555.43*	PISM	618.39	11.34	0.13	618.39	11.34	0.73	618.39	11.34	4.20
CMT7	75	11	160	140	*909.68*	PISM	975.46	7.23	0.18	973.52	7.02	4.74	973.52	7.02	28.39
CMT8	100	9	230	200	*865.95*	PISM	973.94	12.47	0.25	969.90	12.00	3.23	969.90	12.00	18.08
CMT9	150	14	200	200	*1,162.55*	RT	1,287.64	10.76	0.45	1,287.64	10.76	8.85	1,287.64	10.76	45.70
CMT10	199	18	200	200	*1,395.85*	RT	1,538.66	10.23	0.72	1,504.80	7.81	7.30	1,483.99	6.31	150.54
CMT11	120	7	∞	200	*1,042.12**	RT	1,071.07	2.78	0.31	1,056.23	1.35	1.90	1,056.23	1.35	10.11
CMT12	100	10	∞	200	*819.56**	CGL	833.51	1.70	0.26	821.29	0.21	1.59	821.29	0.21	8.73
CMT13	120	11	∞	200	*1,541.14*	RT	1,596.72	3.61	0.30	1,576.52	2.30	2.89	1,576.52	2.30	16.20
CMT14	100	11	1,040	200	*866.37*	GTS	875.75	1.08	0.24	872.83	0.75	2.24	872.83	0.75	12.64
G.avg								*7.70*	*0.33*		*6.74*	*4.41*		*6.64*	*32.71*

Source: Owner

Table 2 Solutions (CPU times) obtained by the pGTS algorithm on benchmarking instances proposed by Golden et al. (1998)

Characteristics of the instances					*Previous solutions*		*Phase 1: initial solution*			*Phase 2: probabilistic granular tabu search*					
Instance	*n*	*k*	*D*	*Q*	*BKS*	*Ref. BKS*	*Cost*	*Gap BKS*	*Time*	*Avg. cost*	*Gap avg. BKS*	*Avg. time*	*Best cost*	*Gap best BKS*	*Total time*
Golden_1	240	9	650	550	*5,623.47*	PA	6,086.38	8.23	0.95	5,914.51	5.18	23.35	5,906.03	5.02	193.67
Golden_2	320	10	900	700	*8,404.61*	VCGLR	11,288.17	34.31	1.62	11,288.17	34.31	15.72	11,288.17	34.31	89.39
Golden_3	400	9	1,200	900	*11,036.23*	PA	12,449.86	12.81	2.61	12,373.97	12.12	49.12	12,373.97	12.12	309.34
Golden_4	480	10	1,600	1,000	*13,590.00*	CP	16,318.40	20.08	3.81	16,154.70	18.87	104.36	16,110.13	18.54	571.67
Golden_5	200	5	1,800	900	*6,460.98*	TK	7,274.80	12.60	0.74	7,239.55	12.05	6.12	7,239.55	12.05	35.28
Golden_6	280	7	1,500	900	*8,404.06*	CP	9,420.25	12.09	1.36	9,355.99	11.33	16.96	9,347.86	11.23	131.05
Golden_7	360	8	1,300	900	*10,102.70*	VCGLR	11,857.03	17.36	2.14	11,804.31	16.84	29.37	11,804.31	16.84	202.42
Golden_8	440	10	1,200	900	*11,635.30*	VCGLR	13,227.14	13.68	3.23	13,106.17	12.64	38.80	13,106.17	12.64	268.71
Golden_9	255	14	∞	1,000	*579.71*	PA	668.73	15.36	1.11	646.18	11.47	18.51	642.29	10.80	126.60
Golden_10	323	16	∞	1,000	*735.66*	CP	847.14	15.15	1.78	821.89	11.72	31.02	819.05	11.34	197.38
Golden_11	399	17	∞	1,000	*912.03*	CP	1,041.46	14.19	2.75	1,013.15	11.09	47.67	1,012.84	11.05	294.65
Golden_12	483	19	∞	1,000	*1,101.50*	CP	1,274.47	15.70	3.85	1,231.61	11.81	104.33	1,231.31	11.78	636.75
Golden_13	252	26	∞	1,000	*857.19*	NB	981.12	14.46	1.23	939.89	9.65	18.73	935.65	9.15	152.45
Golden_14	320	29	∞	1,000	*1,080.55**	NB	1,209.29	11.91	1.74	1,193.54	10.46	32.36	1,193.54	10.46	197.50
Golden_15	396	33	∞	1,000	*1,337.87*	CP	1,507.03	12.64	2.55	1,473.78	10.16	23.27	1,471.49	9.99	206.21
Golden_16	480	36	∞	1,000	*1,611.56*	CP	1,795.61	11.42	3.71	1,765.77	9.57	39.10	1,760.05	9.21	403.93
Golden_17	240	22	∞	200	*707.76**	NB	771.18	8.96	1.00	769.16	8.68	7.79	769.16	8.68	61.11
Golden_18	300	27	∞	200	*995.13**	PA	1,071.51	7.68	1.62	1,070.74	7.60	15.48	1,070.74	7.60	147.57
Golden_19	360	33	∞	200	*1,365.60**	PA	1,468.72	7.55	2.26	1,464.94	7.27	41.61	1,464.94	7.27	279.07
Golden_20	420	38	∞	200	*1,817.59**	IBCP	1,964.37	8.08	3.15	1,962.79	7.99	39.19	1,962.79	7.99	343.55
G. avg								*13.71*	*2.16*		*12.04*	*35.14*		*11.90*	*242.42*

Source: Owner

Table 3 Solutions (CPU times) obtained by the pGTS algorithm on benchmarking instances proposed by Li et al. (2005)

Characteristics of the instances					*Previous solutions*		*Phase 1: initial solution*			*Phase 2: probabilistic granular tabu search*					
Instance	*n*	*k*	*D*	*Q*	*BKS*	*Ref. BKS*	*Cost*	*Gap BKS*	*Time*	*Avg. cost*	*Gap avg. BKS*	*Avg. time*	*Best cost*	*Gap best BKS*	*Total time*
Li_21	560	10	1,800	1,200	*16,212.74*	MBE	17,855.06	10.13	5.27	17,263.04	6.48	43.97	17,263.04	6.48	165.98
Li_22	600	15	1,000	900	*14,499.04*	UPSV	19,908.45	37.31	6.10	17,373.46	19.82	251.58	16,598.52	14.48	1,069.66
Li_23	640	10	2,200	1,400	*18,801.12*	MBE	20,385.44	8.43	6.54	19,861.21	5.64	51.38	19,843.39	5.54	206.29
Li_24	720	10	2,400	1,500	*21,389.33*	MBE	22,973.74	7.41	8.40	22,449.89	4.96	54.77	22,441.01	4.92	289.91
Li_25	760	19	900	900	*16,668.51*	UPSV	23,346.54	40.06	9.48	20,855.33	25.12	945.36	19,793.21	18.75	2,947.87
Li_26	800	10	2,500	1,700	*23,971.74*	MBE	25,619.97	6.88	10.28	25,017.47	4.36	70.73	25,004.31	4.31	295.48
Li_27	840	20	900	900	*17,372.64*	VCGLR	25,943.06	49.33	11.73	21,397.87	23.17	1,031.81	21,013.78	20.96	4,598.44
Li_28	880	10	2,800	1,800	*26,565.92*	MBE	28,208.27	6.18	12.94	27,600.89	3.90	129.83	27,592.61	3.86	481.06
Li_29	960	10	3,000	2,000	*29,154.34*	LGW	31,094.52	6.65	15.19	30,381.70	4.21	133.89	30,347.33	4.09	384.59
Li_30	1,040	10	3,200	2,100	*31,742.51*	MBE	33,276.35	4.83	18.72	32,480.97	2.33	146.49	32,294.45	1.74	804.72
Li_31	1,120	10	3,500	2,300	*34,330.84*	MBE	36,271.13	5.65	20.58	35,536.58	3.51	140.15	35,522.47	3.47	474.20
Li_32	1,200	11	3,600	2,500	*36,928.70*	MBE	40,018.16	8.37	23.33	38,839.42	5.17	151.88	38,834.75	5.16	462.64
G. avg								*15.94*	*12.38*		*9.06*	*262.65*		*7.81*	*1,015.07*

Source: Owner

Table 4 Solutions (CPU Times) obtained by the pGTS algorithm on benchmarking instances by Rochat and Taillard (1995)

Characteristics of the instances					*Previous solutions*		*Phase 1: initial solution*			*Phase 2:probabilistic granular tabu search*					
Instance	*n*	*k*	*D*	*Q*	*BKS*	*Ref. BKS*	*Cost*	*Gap BKS*	*Time*	*Avg. cost*	*Gap avg. BKS*	*Avg. time*	*Best cost*	*Gap best BKS*	*Total time*
tai75a	75	10	∞	1,445	*1,618.36**	PISM	1,645.50	1.68	0.22	1,632.78	0.89	1.74	1,632.78	0.89	10.20
tai75b	75	9	∞	1,679	*1,344.62**	PISM	1,356.56	0.89	0.24	1,355.87	0.84	1.42	1,355.87	0.84	8.45
tai75c	75	9	∞	1,122	*1,291.01**	PISM	1,334.84	3.39	0.24	1,334.84	3.39	1.67	1,334.84	3.39	9.31
tai75d	75	9	∞	1,699	*1,365.42**	PISM	1,421.87	4.13	0.23	1,413.56	3.53	2.24	1,413.56	3.53	12.24
tai100a	100	11	∞	1,409	*2,041.34**	MACS	2,166.05	6.11	0.30	2,166.05	6.11	2.09	2,166.05	6.11	12.41
tai100b	100	11	∞	1,842	*1,939.90**	MBE	2,034.31	4.87	0.18	2,016.66	3.96	1.88	2,016.66	3.96	10.32
tai100c	100	11	∞	2,043	*1,406.20**	MACS	1,434.07	1.98	0.26	1,433.52	1.94	4.03	1,433.52	1.94	24.16
tai100d	100	11	∞	1,297	*1,580.46**	NB	1,818.20	15.04	0.27	1,817.42	14.99	4.41	1,817.42	14.99	24.29
tai150a	150	15	∞	1,544	*3,055.23**	PISM	3,388.60	10.91	0.44	3,368.55	10.26	7.24	3,368.55	10.26	41.84
tai150b	150	14	∞	1,918	*2,727.03**	IBCP	2,890.40	5.99	0.59	2,877.01	5.50	5.40	2,877.01	5.50	32.69
tai150c	150	14	∞	2,021	*2,358.66**	IBCP	2,457.23	4.18	0.31	2,447.68	3.77	4.62	2,447.68	3.77	25.68
tai150d	150	14	∞	1,874	*2,645.39**	PISM	2,788.23	5.40	0.43	2,772.15	4.79	6.33	2,770.62	4.73	40.08
tai385	385	46	∞	65	*24,366.41*	UPSV	25,342.98	4.01	2.38	25,342.98	4.01	146	25,342.98	4.01	839.08
G. avg								*5.28*	*0.47*		*4.92*	*14.51*		*4.92*	*83.90*

Source: Owner

Table 5 Solutions (CPU times) obtained by the pGTS algorithm on benchmarking instances proposed by Uchoa et al. (2014)

Characteristics of the instances					*Previous solutions*		*Phase 1: initial solution*			*Phase 2: probabilistic granular tabu search*					
Instance	*n*	*k*	*D*	*Q*	*BKS*	*Ref. BKS*	*Cost*	*Gap BKS*	*Time*	*Avg. cost*	*Gap avg. BKS*	*Avg. time*	*Best cost*	*Gap best BKS*	*Total time*
X-n101-k25	100	25	∞	206	*27,591*	ILS-SP	29,372	6.46	0.25	29,267	6.07	26.28	29,267	6.07	203.32
X-n106-k14	105	14	∞	600	*26,362*	ILS-SP	27,283	3.49	0.34	27,202	3.19	20.32	27,202	3.19	182.87
X-n110-k13	109	13	∞	66	*14,971*	ILS-SP	16,136	7.78	0.34	16,084	7.43	20.30	16,077	7.39	83.80
X-n115-k10	114	10	∞	169	*12,747*	ILS-SP	13,487	5.81	0.34	13,487	5.81	20.52	13,487	5.81	41.35
X-n120-k6	119	6	∞	21	*13,332*	ILS-SP	14,541	9.07	0.41	14,505	8.80	24.78	14,505	8.80	122.82
X-n125-k30	124	30	∞	188	*55,539*	ILS-SP	58,077	4.57	0.45	58,077	4.57	27.15	58,077	4.57	352.15
X-n129-k18	128	18	∞	39	*28,940*	UHGS	30,328	4.80	0.47	30,281	4.63	28.13	30,281	4.63	805.08
X-n134-k13	133	13	∞	643	*10,916*	ILS-SP	11,518	5.51	0.47	11,502	5.37	28.05	11,502	5.37	58.16
X-n139-k10	138	10	∞	106	*13,590*	ILS-SP	14,497	6.67	0.50	14,497	6.67	30.28	14,497	6.67	63.53
X-n143-k7	142	7	∞	1,190	*15,700*	UHGS	17,487	11.38	0.54	17,468	11.26	32.51	17,468	11.26	60.53
X-n148-k46	147	46	∞	18	*43,448*	ILS-SP	45,109	3.82	0.56	45,109	3.82	33.43	45,109	3.82	935.32
X-n153-k22	152	22	∞	144	*21,220*	UHGS	22,562	6.32	0.60	22,562	6.32	35.92	22,562	6.32	226.10
X-n157-k13	156	13	∞	12	*16,876*	ILS-SP	17,837	5.69	0.61	17,679	4.76	36.80	17,671	4.71	640.42
X-n162-k11	161	11	∞	1,174	*14,138*	ILS-SP	15,487	9.54	0.63	15,487	9.54	37.54	15,487	9.54	57.92
X-n167-k10	166	10	∞	133	*20,557*	UHGS	22,170	7.85	0.68	22,025	7.14	40.72	22,025	7.14	347.97
X-n172-k51	171	51	∞	161	*45,607*	ILS-SP	48,228	5.75	0.74	48,228	5.75	44.66	48,228	5.75	663.03
X-n176-k26	175	26	∞	142	*47,812*	UHGS	52,554	9.92	0.78	52,554	9.92	47.01	52,554	9.92	807.65
X-n181-k23	180	23	∞	8	*25,569*	ILS-SP	26,532	3.77	0.72	26,332	2.98	43.26	26,300	2.86	1,276.69
X-n186-k15	185	15	∞	974	*24,145*	ILS-SP	25,569	5.90	0.84	25,569	5.90	50.44	25,569	5.90	130.41
X-n190-k8	189	8	∞	138	*16,980*	UHGS	18,128	6.76	0.82	17,974	5.85	49.42	17,925	5.57	755.84
X-n195-k51	194	51	∞	181	*44,225*	ILS-SP	45,749	3.45	0.86	45,749	3.45	51.75	45,749	3.45	551.57
X-n200-k36	199	36	∞	402	*58,578*	UHGS	61,197	4.47	0.97	61,197	4.47	58.46	61,197	4.47	389.53
X-n204-k19	203	19	∞	836	*19,565*	UHGS	21,392	9.34	0.83	21,367	9.21	49.60	21,367	9.21	185.24
X-n209-k16	208	16	∞	101	*30,656*	UHGS	32,510	6.05	0.91	32,510	6.05	54.53	32,510	6.05	448.03
X-n214-k11	213	11	∞	944	*10,856*	UHGS	11,970	10.26	0.90	11,970	10.26	54.03	11,970	10.26	81.11
G. avg								*6.58*	*0.62*		*6.37*	*37.84*		*6.35*	*378.82*

Source: Owner

Table 5 Solutions (CPU times) obtained by the pGTS algorithm on benchmarking instances proposed by Uchoa et al. (2014) (continued)

Characteristics of the instances					*Previous solutions*		*Phase 1: initial solution*			*Phase 2: probabilistic granular tabu search*					
Instance	*n*	*k*	*D*	*Q*	*BKS*	*Ref. BKS*	*Cost*	*Gap BKS*	*Time*	*Avg. cost*	*Gap avg. BKS*	*Avg. time*	*Best cost*	*Gap best BKS*	*Total time*
X-n219-k73	218	73	∞	3	*117,595*	ILS-SP	118,364	0.65	1.03	118,015	0.36	667.94	118,010	0.35	5,238.85
X-n223-k34	222	34	∞	37	*40,437*	UHGS	42,391	4.83	0.99	42,391	4.83	124.86	42,391	4.83	712.42
X-n228-k23	227	23	∞	154	*25,742*	UHGS	27,156	5.49	1.17	27,156	5.49	57.15	27,156	5.49	318.24
X-n233-k16	232	16	∞	631	*19,230*	UHGS	20,361	5.88	1.01	20,243	5.27	22.27	20,243	5.27	152.10
X-n237-k14	236	14	∞	18	*27,042*	ILS-SP	29,857	10.41	1.03	29,658	9.67	74.14	29,658	9.67	694.53
X-n242-k48	241	48	∞	28	*82,751*	UPSV	85,228	2.99	1.31	85,228	2.99	485.52	85,228	2.99	2,614.47
X-n247-k50	246	47	∞	134	*37,274*	UHGS	40,551	8.79	1.19	40,551	8.79	239.00	40,551	8.79	1,411.52
X-n251-k28	250	28	∞	69	*38,684*	UPSV	40,566	4.87	1.17	40,566	4.87	148.87	40,566	4.87	940.17
X-n256-k16	255	16	∞	1,225	*18,880*	ILS-SP	20,736	9.83	1.13	20,431	8.22	25.96	20,359	7.83	189.40
X-n261-k13	260	13	∞	1,081	*26,558*	UHGS	28,763	8.30	1.34	28,763	8.30	31.98	28,763	8.30	165.02
X-n266-k58	265	58	∞	35	*75,478*	ILS-SP	78,982	4.64	1.57	78,982	4.64	452.60	78,982	4.64	2,706.02
X-n270-k35	269	35	∞	585	*35,291*	UPSV	37,291	5.67	1.63	37,282	5.64	69.66	37,282	5.64	567.00
X-n275-k28	274	28	∞	10	*21,245*	ILS-SP	22,477	5.80	1.34	22,421	5.53	174.17	22,414	5.50	1,383.24
X-n280-k17	279	17	∞	192	*33,503*	UPSV	36,313	8.39	1.42	36,313	8.39	90.12	36,313	8.39	510.87
X-n284-k15	283	15	∞	109	*20,226*	UPSV	22,262	10.07	1.74	22,224	9.88	169.98	22,224	9.88	1,051.54
X-n289-k60	288	60	∞	267	*95,185*	UPSV	98,139	3.10	1.62	98,139	3.10	276.45	98,139	3.10	1,490.07
X-n294-k50	293	50	∞	285	*47,167*	UPSV	48,441	2.70	1.87	48,441	2.70	100.07	48,441	2.70	632.75
X-n298-k31	297	31	∞	55	*34,231*	UHGS	36,317	6.09	1.60	36,317	6.09	131.39	36,317	6.09	709.79
X-n303-k21	302	21	∞	794	*21,744*	UPSV	23,646	8.75	1.85	23,646	8.75	30.30	23,646	8.75	198.14
X-n308-k13	307	13	∞	246	*25,859*	UHGS	28,436	9.97	1.65	28,414	9.88	51.45	28,414	9.88	482.95
X-n313-k71	312	71	∞	248	*94,044*	UPSV	97,434	3.60	1.84	97,434	3.60	372.15	97,434	3.60	2,211.75
X-n317-k53	316	53	∞	6	*78,355*	ILS-SP	79,586	1.57	1.83	79,164	1.03	858.49	79,163	1.03	6,408.86
X-n322-k28	321	28	∞	868	*29,866*	UPSV	31,848	6.64	2.08	31,757	6.33	30.71	31,757	6.33	227.34
X-n327-k20	326	20	∞	128	*27,556*	UPSV	29,905	8.52	1.79	29,905	8.52	103.19	29,905	8.52	635.25
X-n331-k15	330	15	∞	23	*31,103*	UHGS	34,348	10.43	1.75	34,244	10.10	168.19	34,244	10.10	1,653.74
G. avg								*6.32*	*1.48*		*6.12*	*198.26*		*6.10*	*1,332.24*

Source: Owner

Table 5 Solutions (CPU times) obtained by the pGTS algorithm on benchmarking instances proposed by Uchoa et al. (2014) (continued)

Characteristics of the instances					*Previous solutions*		*Phase 1: initial solution*			*Phase 2: probabilistic granular tabu search*					
Instance	*n*	*k*	*D*	*Q*	*BKS*	*Ref. BKS*	*Cost*	*Gap BKS*	*Time*	*Avg. cost*	*Gap avg. BKS*	*Avg. time*	*Best cost*	*Gap best BKS*	*Total time*
X-n336-k84	335	84	∞	203	*139,197*	ILS-SP	144,588	3.87	2.04	144,588	3.87	622.76	144,588	3.87	3,590.28
X-n344-k43	343	43	∞	61	*42,099*	UHGS	44,747	6.29	2.02	44,683	6.14	277.11	44,683	6.14	1,824.85
X-n351-k40	350	40	∞	436	*25,946*	UHGS	27,096	4.43	2.54	27,096	4.43	61.90	27,096	4.43	642.66
X-n359-k29	358	29	∞	68	*51,509*	UHGS	53,306	3.49	2.18	53,306	3.49	225.75	53,306	3.49	1,352.91
X-n367-k17	366	17	∞	218	*22,814*	UHGS	25,043	9.77	2.27	25,043	9.77	68.68	25,043	9.77	536.86
X-n376-k94	375	94	∞	4	*147,713*	ILS-SP	149,321	1.09	2.56	149,044	0.90	2,074.40	149,017	0.88	15,321.31
X-n384-k52	383	52	∞	564	*66,081*	UHGS	69,478	5.14	2.53	69,478	5.14	190.04	69,478	5.14	1,382.04
X-n393-k38	392	38	∞	78	*38,269*	UHGS	40,583	6.05	2.49	40,437	5.67	304.91	40,432	5.65	2,462.79
X-n401-k29	400	29	∞	745	*66,243*	UHGS	68,913	4.03	2.70	68,831	3.91	152.22	68,831	3.91	1,294.85
X-n411-k19	410	19	∞	216	*19,718*	UHGS	21,747	10.29	3.40	21,747	10.29	62.60	21,747	10.29	335.15
X-n420-k130	419	130	∞	18	*107,798*	ILS-SP	112,308	4.18	3.10	112,308	4.18	1,874.09	112,308	4.18	10,199.56
X-n429-k61	428	61	∞	536	*65,501*	UHGS	68,521	4.61	3.01	68,511	4.60	207.95	68,511	4.60	1,880.68
X-n439-k37	438	37	∞	12	*36,395*	ILS-SP	38,533	5.87	3.18	38,380	5.45	437.01	38,380	5.45	2,701.28
X-n449-k29	448	29	∞	777	*55,358*	UPSV	58,487	5.65	3.15	58,487	5.65	108.63	58,487	5.65	589.71
X-n459-k26	458	26	∞	1,106	*24,181*	UHGS	26,244	8.53	4.15	26,120	8.02	81.81	26,061	7.77	635.56
X-n469-k138	468	138	∞	256	*221,909*	ILS-SP	235,387	6.07	3.76	234,875	5.84	2,172.87	234,875	5.84	15,125.36
X-n480-k70	479	70	∞	52	*89,535*	UHGS	93,383	4.30	3.69	93,204	4.10	3,323.14	93,204	4.10	19,337.51
X-n491-k59	490	59	∞	428	*66,633*	UHGS	69,362	4.10	3.80	69,362	4.10	122.70	69,362	4.10	840.24
X-n502-k39	501	39	∞	13	*69,253*	UHGS	71,227	2.85	4.00	70,820	2.26	1,454.49	70,802	2.24	8,813.87
X-n513-k21	512	21	∞	142	*24,201*	UHGS	27,312	12.85	4.15	27,312	12.85	111.73	27,312	12.85	676.15
X-n524-k153	523	137	∞	125	*154,594*	UPSV	165,700	7.18	4.57	165,700	7.18	1,887.66	165,700	7.18	13,145.24
X-n536-k96	535	96	∞	371	*95,122*	UHGS	99,444	4.54	6.08	99,432	4.53	646.10	99,432	4.53	4,506.78
X-n548-k50	547	50	∞	11	*86,710*	ILS-SP	89,811	3.58	5.15	89,383	3.08	2,051.78	89,383	3.08	13,502.58
X-n561-k42	560	42	∞	74	*42,756*	UHGS	45,596	6.64	6.27	45,596	6.64	285.34	45,596	6.64	1,569.86
X-n573-k30	572	30	∞	210	*50,780*	UHGS	52,439	3.27	4.98	52,439	3.27	417.93	52,439	3.27	2,516.24
G. avg								*5.55*	*3.51*		*5.41*	*768.94*		*5.40*	*4,991.37*

Source: Owner

Table 5 Solutions (CPU times) obtained by the pGTS algorithm on benchmarking instances proposed by Uchoa et al. (2014) (continued)

Characteristics of the instances					*Previous solutions*		*Phase 1: initial solution*			*Phase 2: probabilistic granular tabu search*					
Instance	*n*	*k*	*D*	*Q*	*BKS*	*Ref. BKS*	*Cost*	*Gap BKS*	*Time*	*Avg. cost*	*Gap avg. BKS*	*Avg. time*	*Best cost*	*Gap best BKS*	*Total time*
X-n586-k159	585	159	∞	28	190,543	UHGS	199,330	4.61	5.62	199,330	4.61	4,272.05	199,330	4.61	24,696.5
X-n599-k92	598	92	∞	487	108,813	UHGS	113,179	4.01	5.66	113,179	4.01	665.36	113,179	4.01	4,166.4
X-n613-k62	612	62	∞	523	59,778	UHGS	62,564	4.66	7.42	62,466	4.50	252.80	62,466	4.50	1,723.4
X-n627-k43	626	43	∞	110	62,366	UHGS	65,448	4.94	5.92	65,393	4.85	1,631.71	65,393	4.85	10,698.3
X-n641-k35	640	35	∞	1,381	63,839	UHGS	67,933	6.41	6.18	67,832	6.25	269.35	67,832	6.25	2,022.9
X-n655-k131	654	131	∞	5	106,780	ILS-SP	108,195	1.33	7.85	107,874	1.02	6,198.30	107,857	1.01	45,770.2
X-n670-k130	669	126	∞	129	146,705	UHGS	158,702	8.18	9.44	158,702	8.18	3,094.76	158,702	8.18	16,579.2
X-n685-k75	684	75	∞	408	68,425	UHGS	71,485	4.47	6.87	71,454	4.43	377.28	71,454	4.43	2,387.0
X-n701-k44	700	44	∞	87	82,292	UPSV	85,435	3.82	7.34	85,435	3.82	862.31	85,435	3.82	5,304.6
X-n716-k35	715	35	∞	1,007	43,525	UHGS	45,930	5.53	8.09	45,822	5.28	571.06	45,822	5.28	3,123.2
X-n733-k159	732	159	∞	25	136,366	UHGS	139,884	2.58	9.99	139,884	2.58	2,958.89	139,884	2.58	16,846.8
X-n749-k98	748	98	∞	396	77,700	UPSV	792,87	2.04	8.49	79,287	2.04	409.33	79,287	2.04	2,228.2
X-n766-k71	765	71	∞	166	114,683	UHGS	118,974	3.74	9.21	118,974	3.74	1,347.79	118,974	3.74	7,704.1
X-n783-k48	782	48	∞	832	72,727	UPSV	76,807	5.61	11.43	76,725	5.50	771.90	76,725	5.50	5,136.6
X-n801-k40	800	40	∞	20	73,587	UHGS	77,251	4.98	10.16	76,950	4.57	2,303.58	76,950	4.57	12,232.5
X-n819-k171	818	171	∞	358	158,611	UHGS	166,505	4.98	10.88	166,455	4.95	1,125.10	166,455	4.95	16,074.8
X-n837-k142	836	142	∞	44	194,266	UHGS	200,783	3.35	11.55	200,783	3.35	5,039.03	200,783	3.35	26,693.2
X-n856-k95	855	95	∞	9	89,060	ILS-SP	92,505	3.87	10.68	91,832	3.11	5,064.98	91,829	3.11	28,832.6
X-n876-k59	875	59	∞	764	99,715	UHGS	102,126	2.42	15.35	102,017	2.31	758.69	102,017	2.31	5,525.4
X-n895-k37	894	37	∞	1,816	54,172	UHGS	59,258	9.39	12.88	59,087	9.07	738.29	59,087	9.07	4,769.9
X-n916-k207	915	207	∞	33	329,836	UHGS	345,189	4.65	13.28	345,189	4.65	8,937.55	345,189	4.65	58,044.6
X-n936-k151	935	151	∞	138	133,105	UPSV	144,189	8.33	13.45	144,189	8.33	3,477.75	144,189	8.33	20,073.6
X-n957-k87	956	87	∞	11	85,672	UHGS	89,510	4.48	13.56	88,919	3.79	1,661.23	88,903	3.77	22,137.1
X-n979-k58	978	58	∞	998	119,194	UHGS	123,752	3.82	14.84	123,744	3.82	1,134.63	123,744	3.82	6,532.9
X-n1001-k43	1,000	43	∞	131	72,742	UHGS	77,332	6.31	15.99	77,332	6.31	735.43	77,332	6.31	3,966.0
G. avg								*4.74*	*10.09*		*4.60*	*2,186.37*		*4.60*	*14,130.8*

Source: Owner

In addition, for each algorithm, the following global values are reported

- *G.avg*: average percentage gap of the solution cost found by the corresponding algorithm on a complete set of instances.

For the values of BKS and Ref. BKS, we have considered all the previously published methods proposed for the DCVRP. Therefore, also the results obtained by exact algorithms and by early heuristic algorithms have been take into account. The BKS values for which the optimality has been proved by previously published works are remarked with an asterisk (*). For each instance, the costs that are equal to the corresponding value of BKS are reported in bold.

In summary, we solved 159 DCVRP benchmark instances taken from five well-known sets existing in the literature. The results of implementation of the proposed algorithm on the instances are reported from Tables 1 to 5. According to the results reported in Tables 1–5, the pGTS algorithm provides high quality solutions with the objective function values not more than 12% (in average) of the objective values of the solutions obtained by the best published algorithms. Note that the proposed approach allows obtaining high quality solutions within short computing times by using a simple GTS scheme.

4 Concluding remarks and future research

In this paper, we propose an effective pGTS approach for the DCVRP. The algorithm is based on the GTS introduced by Toth and Vigo (2003) for the CVRP. In the proposed approach, after the construction of an initial solution by using a modified saving heuristic, we apply a probabilistic GTS procedure which considers five neighbourhoods and a probabilistic discrete function, which is modified dynamically during the search by favouring the moves that have improved the best solution found so far. A perturbation procedure is applied whenever the algorithm remains in a local optimum for a given number of iterations.

We compare the proposed algorithm with the best-known solutions proposed for the DCVRP on different sets of benchmark instances from the literature. The results suggest that the pGTS is highly competitive with the previous published algorithms providing solutions with relative good quality. The proposed approach could be applied to other extensions of the DCVRP such as the vehicle routing problem with time windows (VRPTW), the multi depot vehicle routing problem (MDVRP), and other problems obtained by adding constraints as pickups and deliveries, periodicity, etc.

Acknowledgements

This work has been partially supported by Pontificia Universidad Javeriana Cali and Universidad del Valle from Colombia; and by Universidad del Bío-Bío and Universidad Andrés Bello from Chile. R. Linfati acknowledges the funds received by project FONDECYT 11150370. This support is gratefully acknowledged.

References

Alba, E. and Dorronsoro, B. (2004) 'Solving the vehicle routing problem by using cellular genetic algorithms', *Evolutionary Computation in Combinatorial Optimization*, pp.11–20, Springer Berlin, Heidelberg, Science + Business Media.

Beasley, J. (1981) 'Adapting the savings algorithm for varying inter-customer travel times', *Omega*, Vol. 9, No. 6, pp.658–659.

Berger, J. and Barkaoui, M. (2003) 'A hybrid genetic algorithm for the capacitated vehicle routing problem', *Genetic and Evolutionary Computation – GECCO 2003*, pp.646–656, Springer Berling Heidelberg, Science + Business Media.

Bouzid, M.C., Aït Haddadene, H. and Salhi, S. (2016) 'An integration of Lagrangian split and VNS: the case of the capacitated vehicle routing problem', *Computers & Operations Research*, doi: 10.1016/j.cor.2016.02.009.

Christofides, N., Christofides, N., Toth, P. and Sandi, C. (1979) *Combinatorial Optimization*, John Wiley & Sons, UK.

Clarke, G. and Wright, J.W. (1964) 'Scheduling of vehicles from a central depot to a number of delivery points', *Operations Research*, Vol. 12, No. 4, pp.568–581.

Cordeau, J-F., Gendreau, M. and Laporte, G. (1997) 'A tabu search heuristic for periodic and multi-depot vehicle routing problems', *Networks*, Vol. 30, No. 2, pp.105–119.

Doyuran, T. and Çatay, B. (2010) 'A robust enhancement to the Clarke-Wright savings algorithm', *Journal of the Operational Research Society*, Vol. 62, No. 1, pp.223–231.

Dror, M. and Trudeau, P. (1986) 'Stochastic vehicle routing with modified savings algorithm', *European Journal of Operational Research*, Vol. 23, No. 2, pp.228–235.

Escobar, J.W. (2013) 'Heuristic algorithms for the capacitated location-routing problem and the multi-depot vehicle routing problem', *4OR-Q J. Oper. Res.*, Vol. 12, No. 1, pp.99–100.

Escobar, J.W. and Linfati, R. (2012) '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, No. 21, pp.139–150.

Escobar, J.W., Linfati, R. and Toth, P. (2013) 'A two-phase hybrid heuristic algorithm for the capacitated location-routing problem', *Computers & Operations Research*, Vol. 40, No. 1, pp. 70–79.

Escobar, J.W., Linfati, R., Baldoquin, M.G. and Toth, P. (2014b) 'A granular variable tabu neighborhood search for the capacitated location-routing problem', *Transportation Research Part B: Methodological*, Vol. 67, pp.344–356.

Escobar, J.W., Linfati, R., Toth, P. and Baldoquin, M.G. (2014a) 'A hybrid granular Tabu search algorithm for the multi-depot vehicle routing problem', *Journal of Heuristics*, Vol. 20, No. 5, pp.483–509.

Feo, T.A. and Resende, M.G.C. (1989) 'A probabilistic heuristic for a computationally difficult set covering problem', *Operations Research Letters*, Vol. 8, No. 2, pp.67–71.

Feo, T.A. and Resende, M.G.C. (1995) 'Greedy randomized adaptive search procedures', *Journal of Global Optimization*, Vol. 6, No. 2, pp.109–133.

Franceschi, R.D., Fischetti, M. and Toth, P. (2005) 'A new ILP-based refinement heuristic for vehicle routing problems', *Mathematical Programming*, Vol. 105, Nos. 2–3, pp.471–499.

Gendreau, M., Hertz, A. and Laporte, G. (1994) 'A tabu search heuristic for the vehicle routing problem', *Management Science*, Vol. 40, No. 10, pp.1276–1290.

Gillett, B.E. and Miller, L.R. (1974) 'A heuristic algorithm for the vehicle-dispatch problem', *Operations Research*, Vol. 22, No. 2, pp.340–349.

Golden, B.L., Raghavan, S. and Wasil, E.A. (2008) *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer-Verlag, New York, USA.

Golden, B.L., Wasil, E.A., Kelly, J.P. and Chao, I.M. (1998) 'The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational result', *Fleet Management and Logistics*, pp.33–56, Springer, USA.

Groër, C., Golden, B. and Wasil, E. (2011) 'A parallel algorithm for the vehicle routing problem', *INFORMS Journal on Computing*, Vol. 23, No. 2, pp.315–330.

Holmes, R.A. and Parker, R.G. (1976) 'A vehicle scheduling procedure based upon savings and a solution perturbation scheme', *Operational Research Quarterly*, Vol. 27, No. 1, p.83.

Jin, J., Crainic, T.G. and Løkketangen, A. (2014) 'A cooperative parallel metaheuristic for the capacitated vehicle routing problem', *Computers & Operations Research*, Vol. 44, pp.33–41.

Kara, I. (2011) 'Arc based integer programming formulations for the distance constrained vehicle routing problem', *3rd IEEE International Symposium on Logistics and Industrial Informatics*, doi: 10.1109/lindi.2011.6031159.

Li, F., Golden, B. and Wasil, E. (2005) 'Very large-scale vehicle routing: new test problems, algorithms, and results', *Computers & Operations Research*, Vol. 32, No. 5, pp.1165–1179.

Linfati, R., Escobar, J.W. and Cuevas, B. (2014a) 'An algorithm based on granular tabu search for the problem of balancing public bikes by using multiple vehicles', DYNA, Vol. 81, No. 186, p.284–294.

Linfati, R., Escobar, J.W. and Gatica, G. (2014b) 'Un algoritmo metaheurístico para el problema de localización y ruteo con flota heterogénea', *Ingeniería y Ciencia*, Vol. 10, No. 19, pp.55–76.

Mester, D. and Bräysy, O. (2005) 'Active guided evolution strategies for large-scale vehicle routing problems with time windows', *Computers & Operations Research*, Vol. 32, No. 6, pp.1593–1614.

Nagarajan, V. and Ravi, R. (2011) 'Approximation algorithms for distance constrained vehicle routing problems', *Networks*, Vol. 59, No. 2, pp.209–214.

Nagata, Y. and Bräysy, O. (2009) 'Edge assembly-based memetic algorithm for the capacitated vehicle routing problem', *Networks*, Vol. 54, No. 4, pp.205–215.

Paessens, H. (1988) 'The savings algorithm for the vehicle routing problem', *European Journal of Operational Research*, Vol. 34, No. 3, pp.336–344.

Pecin, D., Pessoa, A., Poggi, M. and Uchoa, E. (2016) 'Improved branch-cut-and-price for capacitated vehicle routing', *Mathematical Programming Computation*, doi:10.1007/s12532-016-0108-8.

Prins, C. (2004) 'A simple and effective evolutionary algorithm for the vehicle routing problem', *Computers & Operations Research*, Vol. 31, No. 12, pp.1985–2002.

Puenayán, D.E., Londoño, J.C., Escobar, J.W. and Linfati, R. (2014) '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, No. 25, pp.81–98.

Rochat, Y. and Taillard, É.D. (1995) 'Probabilistic diversification and intensification in local search for vehicle routing', *Journal of Heuristics*, Vol. 1, No. 1, pp.147–167.

Taillard, É. (1993) 'Parallel iterative search methods for vehicle routing problems', *Networks*, Vol. 23, No. 8, pp.661–673.

Tarantilis, C.D. and Kiranoudis, C.T. (2002) 'BoneRoute: an adaptive memory-based method for effective fleet management', *Annals of operations Research*, Vol. 115, Nos. 1–4, pp.227–241.

Toklu, N.E., Gambardella, L.M. and Montemanni, R. (2014) 'A multiple ant colony system for a vehicle routing problem with time windows and uncertain travel times', *Journal of Traffic and Logistics Engineering*, Vol. 2, No. 1, pp.52–58.

Toth, P. and Vigo, D. (2003) 'The granular tabu search and its application to the vehicle-routing problem', *INFORMS Journal on Computing*, Vol. 15, No. 4, pp.333–346.

Toth, P. and Vigo, D. (Eds.) (2014) *Vehicle Routing: Problems, Methods, and Applications*, Society for Industrial & Applied Mathematics (SIAM), 3600 Market St., 6th Fl. Philadelphia, Vol. 18.

Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Subramanian, A. and Vidal, T. (2014) *New Benchmark Instances for the Capacitated Vehicle Routing Problem*, Research Report Engenharia de Produção, Universidade Federal Fluminense.

Vidal, T., Crainic, T.G., Gendreau, M., Lahrichi, N. and Rei, W. (2012) 'A hybrid genetic algorithm for multidepot and periodic vehicle routing problems', *Operations Research*, Vol. 60, No. 3, pp.611–624.

Zhou, Y., Xie, J. and Zheng, H. (2013) 'A hybrid bat algorithm with path relinking for capacitated vehicle routing problem', *Mathematical Problems in Engineering*, Vol. 2013, Article ID 392789, 10pp., doi: 10.1155/2013/392789.