Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 34 (2007) 1561–1584
computers & operations research
www.elsevier.com/locate/cor

A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows

G.B. Alvarenga[a,*], G.R. Mateus[b], G. de Tomi[c]

[a] *Department of Computer Science, Federal University of Lavras, UFLA, Lavras - Brazil*
[b] *Department of Computer Science, Federal University of Minas Gerais, UFMG, Belo Horizonte - Brazil*
[c] *Department of Mining and Petroleum Engineering, University of São Paulo, USP, São Paulo - Brazil*

Available online 20 December 2005

Abstract

The Vehicle Routing Problem with Time Windows (VRPTW) is a well-known and complex combinatorial problem, which has received considerable attention in recent years. This problem has been addressed using many different techniques including both exact and heuristic methods. The VRPTW benchmark problems of Solomon [Algorithms for the vehicle routing and scheduling problems with time window constraints, Operations Research 1987; 35(2): 254–65] have been most commonly chosen to evaluate and compare all algorithms. Results from exact methods have been improved considerably because of parallel implementations and modern branch-and-cut techniques. However, 24 out of the 56 high order instances from Solomon's original test set still remain unsolved. Additionally, in many cases a prohibitive time is needed to find the exact solution. Many of the heuristic methods developed have proved to be efficient in identifying good solutions in reasonable amounts of time. Unfortunately, whilst the research efforts based on exact methods have been focused on the total travel distance, the focus of almost all heuristic attempts has been on the number of vehicles. Consequently, it is more difficult to compare and take advantage of the strong points from each approach. This paper proposes a robust heuristic approach for the VRPTW using travel distance as the main objective through an efficient genetic algorithm and a set partitioning formulation. The tests were produced using real numbers and truncated data type, allowing a direct comparison of its results against previously published heuristic and exact methods. Furthermore, computational results show that the proposed heuristic approach outperforms all previously known and published heuristic methods in terms of the minimal travel distance.

Keywords: Vehicle routing problem; Hybrid algorithm; Genetic algorithm

* Corresponding author.
E-mail addresses: guilherme@dcc.ufla.br (G.B. Alvarenga), mateus@dcc.ufmg.br (G.R. Mateus), gdetomi@usp.br (G. de Tomi).

0305-0548/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2005.07.025

1. Introduction

The vehicle routing problem with time windows (VRPTW) has been extensively studied in the operations research community. Firstly, because VRPTW is still one of the most difficult problems in combinatorial optimization and consequently presents a great challenge. Secondly, in a more practical aspect, this problem contributes directly to a real opportunity to reduce costs in the important area of logistics. Transportation management, and more specifically vehicle routing, has a considerable economical impact on all logistic systems. In the VRPTW, a fleet of K identical vehicles supplies goods to N customers. All vehicles have the same capacity Q. For each customer i, $i=1, \ldots, N$, the demand of goods, q_i, the service time s_i, and the time window $[a_i, b_i]$ to meet the demand in i are known. The component s_i represents the loading or unloading service time at the customer i and a_i describes the earliest time when it is possible to start the service. If any vehicle arrives at customer i before a_i it must wait. The vehicle must start the customer service before b_i. This type of time window constraints is known as a *hard time window.* All vehicle routes start and finish at the central depot. Each customer must be visited once. The locations of the central depot and all customers, the minimal distance d_{ij} and the travel time t_{ij} between all locations are given. The objective is to find the feasible solution with the minimal total travel distance or with the minimal number of vehicles. In this paper, only the first objective is considered.

Significant improvements in Solomon's benchmark problem instances were established by Rochat [1] using a tabu search metaheuristic method. In that publication (1995), Rochat improved or reached 47 heuristic solutions from 56 Solomon's original instances. Another important characteristic is the post-optimization technique used by Rochat. This technique consists of saving all partial solutions identified during the tabu search algorithm for future usage. The routes of each intermediate solution are included in a set T. Then, after the stop criterion of tabu search has been achieved, the best solution that can be built using routes from T may be found by solving a set partitioning problem using CPLEX MIP software. Although Rochat has minimized the travel distance (TD) for the capacitated vehicle routing problem (CVRP), their results for the Solomon's test (VRPTW) were obtained using the number of vehicles (NV) as the first objective.

In general, when heuristic methods were used, the number of vehicles was chosen as the first objective and the total TD only as the second. However, considering the results for the Solomon's test problems, the number of vehicles found was the same in many works, so the second objective, TD, was the distinguishing criterion. In fact, NV and TD represent concurrent objectives. Strategies which treat each objective separately in distinguished phases, have reached the best results at the moment in the literature.

Berger et al. [2] have improved some of the results of Solomon's benchmark using parallel two-population co-evolution genetic algorithms, Pop1 and Pop2. The objective in Pop1 is to minimize the total distance for a fixed number of vehicles using feasible individual solutions within the population. On the other hand, in Pop2 the individual solutions have the number of vehicles fixed in one unit less than in Pop1, hence in principle they are infeasible solutions. The GA objective in Pop2 is to minimize the total violated time windows. The global objective is primarily to minimize NV and then to minimize TD as a secondary objective. Each time a feasible individual is found in Pop2, the population Pop1 is replaced by Pop2 and the fixed numbers of vehicles considered in both populations are decreased by one. The algorithm ensures that at least one feasible solution is present in Pop1. Therefore the difference in the number of vehicle in Pop1 and Pop2 is always equal to one unit.

Berger has also tested the algorithm for the classical 56 Solomon's instances. Although there are two parallel evolutions, only one machine was used (Pentium IV 2.4 GHz). The stop criterion was a

1800-second limit for each run. The parameters were adjusted empirically by running the same Solomon's test set before the actual evaluation runs. The parameters were fixed in all instances, except for the clusterized classes C1 and C2. Berger found 6 new results (R108, R110, RC105, RC106, R210 e R211). Currently, three out of these continue to be the best known and published solutions (R108, RC105 e RC106) considering NV as primary objective and TD as secondary objective. One of the most important advantages of this work is the total NV for all 56 instances of Solomon, with 405 vehicles, one of the best results in the literature.

Homberger et al. [3] have also presented good results for many Solomon's benchmark problems using two evolutionary meta-heuristic methods in a similar two-stage strategy. Two different heuristic methods were proposed, ES1 and ES2. These heuristic methods reduced the number of vehicles in two class R1 instances (R104 and R112). In the R109 instance, the ES1 strategy still produced a new result, maintaining NV and reducing TD. In the same way, ES2 improved the results of TD in R105 and R107. In the R2 class, five new results were produced by ES1 and three by ES2. The results in the clusterized classes, C1 and C2, were equivalent to the best known for all problems. ES2 still produced two new results for RC1 and two others for RC2, while ES1 produced two other new results for RC2. In total, a set of 20 new results were produced, out of which only 2 still continue to be undefeated.

Presently, the best results for the Solomon's benchmark problems, using NV as primary objective and TD as secondary objective, are distributed through many publications. They are summarized in [4]. Few papers on heuristic methods have addressed total TD as the first objective to be minimized, as indicated in Rousseau et al. [5]. In [6] the authors seem to compare their primary objective, TD, with results from research efforts in the literature where the focus was on NV, and the specific comparison criteria are not clear.

While NV was selected in almost all previous efforts based on heuristics, almost all exact works have elected TD as the only objective. In fact, the choice of the most appropriate objective depends on specific rules and peculiarities of each individual business. In Brazil, for example, a significant portion of goods delivering companies have limited the number of vehicles in their own fleet to less or equal to 70% of their actual requirement. Consequently, a large amount of goods are delivered by third-parties, usually small business or even self-employed owners of individual trucks. These third-party companies are generally named "aggregated trucks". The payment rule applied to aggregated trucks is normally based on the total travelled distance. In this case, minimization of the TD is the most attractive and primary objective for the hiring company. Another typical scenario, where the minimization of TD is also appropriate, occurs when the amount of goods is smaller than the total capacity of the available fleet. Consequently, many real-life situations justify the study of new algorithms and techniques to improve the VRPTW results in terms of total travelled distance. The deficiency of heuristic approaches in this direction justifies this effort.

2. The set partitioning model for the VRPTW

The VRPTW described in the previous section can be formulated as a set partitioning problem (SPP) as follows:

$$\min \quad \sum_{r \in R} c_r x_r \tag{2.1}$$

$$\text{s.t.} \quad \sum_{r \in R} \delta_{ir} x_r = 1 \quad \forall i \in C \tag{2.2}$$

$$x_r \in \{0.1\} \tag{2.3}$$

where R is the set of routes, C the set of customers, c_r the travel distance on the route r, x_r the decision variable, 1 if the route r is considered in the solution and 0 otherwise, δ_{ir} the auxiliary parameters to indicate the set of customers present in each route r, 1 if the customer i is served by the route r and 0 otherwise.

The search for the optimal integer solution, using the SPP model above, where all possible routes are included in the set R is only possible for instances with a small number of customers. Supposing a problem with 50 customers where time and capacity constraints are sufficient to restrain possible routes up to 10 customers, the number of possible routes is

$$\sum_{n=1}^{10} \left(\frac{50!}{(50-n)!} \right) \approx 3.8 \times 10^{16} \tag{2.4}$$

Consequently, it is not possible for the SPP model to handle directly any MIP algorithm even in instances with 50 customers or less. A common solution is the application of the Dantzig–Wolfe decomposition method which divides the problem into a master problem, with a reduced number of routes, and a secondary problem, where the objective is to find routes (columns) with negative reduced cost to be inserted in the master problem. When no routes with negative reduced cost exist, the solution of the master problem is the solution of the original global problem. The results in the literature show that it continues to be very hard to solve to optimality, despite the fact that decomposition seems to be an option to avoid the increasing number of columns in the original SPP model. Modern branch-and-cut techniques have been used to improve convergence which made possible the identification of new optimal results for previously unsolved VRPTW problems in the literature. See [7] and [8] for modern branch-and-cut techniques applied to routing problems. Exact approaches using Dantzig–Wolfe decomposition for the VRPTW can be found in [9] and [10]. Many problems, as the majority in the classical Solomon's test sets R2 and RC2 continue to be unsolved for 50 and 100 customers. The problems in the real world normally present more than 100 customers which indicated that heuristic methods for the VRPTW are also necessary.

3. Genetic and set partitioning two-phase algorithm

The solution framework was motivated by the fact that a local minimum for the VRPTW has a significant possibility of containing routes that are also found in the global optimum. This fact can be confirmed by the comparison of the global optimal results produced by exact methods with many of the reasonable quality results produced by heuristic methods. If several local optimal solutions are produced with reasonable quality, it is possible to join these routes in a set R and then apply the SPP model described in the previous section. If the total number of routes considered in the R set is not large enough, a reasonable amount of time will be required to produce an improved quality solution using R. In addition, the time required to produce some heuristic quality solution, can be smaller than the amount of time required by heuristic methods dedicated to a final solution. This occurs because the amin effort is focused on escaping from local minimal solutions and then on doing fine adjustments to obtain better quality solutions.

This idea is not entirely new. Rochat [1] has used a similar approach as a post-optimization technique after a tabu search. However, there are new characteristics in the current proposal. First of all, the focus of Rochat's work remained mainly in the quality of the local search result, where the SPP model was used only as a refinement. Using a different approach, a fast genetic algorithm is proposed here as a route generator for the main SPP formulation, where both the amount of time required to identify several local minimal solutions and their diversity are more important than the quality of any single individual solution. Rochat has also mentioned that no significant improvement in the solution was obtained using their post-optimization technique, often less than 1%. The second difference with the current approach is that Rochat used intermediate solutions from a continuous improvement procedure to produce the set of routes for the SPP run against independent and different local minimal solutions. Finally, the objective function used by Rochat is the minimization of the number of vehicles rather than the total travel distance as primary objective.

3.1. Searching for local minimal solutions

The objective in the first phase, as summarized above, is to produce high quality routes to be inserted in the SPP model, Eqs. (2.1) to (2.2). However, to find quality routes is normally a hard task. Many classical heuristic methods, like PFIH [11] have been proposed in the literature to produce routes for the VRPTW. If each route is produced independently, without considering the global problem, it is often very difficult that these routes combined can produce a good solution for the whole problem. It seems there is no evident way to evaluate the quality of any distinct route separately. Consequently, desirable routes to be included in the set R of Eq. (2.1), with more chances to produce a nearly optimal solution, are those routes evaluated as part of a complete solution, i.e., those found in a good local minimal solution.

In order to produce the set of R routes, a fast genetic algorithm was implemented and executed independently many times. Each independent genetic algorithm execution is considered to be an island of evolution, because there is no influence or genetic material interchange between them. The island idea, from the natural evolution theory, is related to a limited set of individuals with more possibility of occurring a crossover and consequently of exchanging their genetic material. As a result, the evolution of each island is completely isolated from each other, enabling the necessary diversity. See [12] for more information about islands in genetic algorithms. Additionally, it is important to distinguish this approach from re-start methods. In the first approach, the solutions from many islands will be combined afterwards to obtain a higher end-quality solution using the SPP model.

3.1.1. The genetic algorithm

The first algorithms that use a natural evolution as the central strategy to solve problems were published in the 1950s, such as Fraser [13] and Box [14]. In 1966, Forgel et al. [15] proposed a method called Evolutionary Programming. Following that, in 1973, Rochenberg [16] introduced the method so called Evolution Strategies. The Genetic Algorithm itself, or simply GA, was proposed by Holland [17] in 1975. All these proposals were based in the natural reproduction, selection and evolution of Darwin's th eory [18], 1859. Ever since, GA has been popular because it can contribute to find good solutions for complex mathematical problems, like the VRP and other NP-hard problems, in a reasonable amount of time.

A basic GA with a global population substitution was used in this work, as illustrated in the pseudo code of Algorithm 1

Algorithm 1. *Simple GA used to produce fast a local minimum.*

Function *GA : individual;*
begin
  TimeIni : = now;
  Start_initial_population; / Using stochastic PFIH*
  while*(now –TimeIni<TimeLimit)* ***do***
  begin
    Individuals_evaluation;
    Selection;
    Crossover;
    Elitism;
    Mutation;
    Update_population;
  end;
  return*(best_individual_current_generation);*
end;

The individual representation is very simple. Each customer has a unique integer identifier $i, i = 1, \ldots, N$, where N is the number of customers. The chromosome was defined as a string of integers, representing a route to be served by only one vehicle. An individual, who represents a complete solution, and consequently many routes, is a set of chromosomes. The central depot is not considered in this representation, because all routes necessarily start and end on it.

3.1.1.1. Initial Population A fast and simple heuristic procedure to distribute all customers in the vehicles, if used to obtain the first individual generation, can reduce significantly the GA time necessary to reach the reasonable local minima. Because of this, the heuristic method proposed by Solomon [11], called *Push Forward Insertion Heuristic* (PFIH), has been frequently used by many researchers with this purpose. For a detailed description of the PFIH method, see [19]. In the present work a modified PFIH, called *stochastic PFIH*, is applied. In the original PFIH, the Eq. (3.1), defines the first customer in each new route. As subsequent customers are chosen one by one minimizing the cost, the original PFIH is totally deterministic. In the proposed stochastic PFIH, a total randomized choice is used to define the first customer to be inserted in each new route. That is necessary to produce distinguished individuals in the first GA generation. After the first customer has been randomly selected, the second one will be the one with the minimal insertion cost. Each feasible customer position in the route in construction is evaluated. A new route is created only if no more customer feasible insertions are possible.

$$c_i = -\alpha d_{0i} + \beta b_i + \gamma((p_i/360)d_{0i}) \tag{3.1}$$

where α is the 0.7 (empirically calculated by Solomon [11]); β the 0.1 (empirically calculated by Solomon [11]); γ the 0.2 (empirically calculated by Solomon [11]); d_{0i} the distance from customer i to the central depot; b_i the upper time window limit to reach the customer i; p_i the polar angle of the customer i from the central depot.

3.1.1.2. Selection Through this step, pairs of individuals are selected to crossover. There are many methods proposed in the literature for selection in GAs. The two most popular are roulette wheel selection and tournament selection. In the roulette wheel selection method, the probability of an individual to participate in a crossover is directly proportional to his relative fitness. This method is very sensitive to the evaluation function and almost always some additional control is necessary, for example fitness scaling.

In this paper, a k-way tournament selection method is used in the GA. In a k-way tournament, k individuals are selected randomly. Then, the individual who presents the highest fitness is the winner. This process is repeated until the number of selected individuals necessary to the crossover phase has been reached. Two individuals are selected for each crossover, which produces only one new offspring.

3.1.1.3. Fitness The fitness function to evaluate the individuals is always related to the objective function, but not necessarily identical. The inverse of the total travel distance is used to represent the fitness of the individuals, Eq. (3.2).

$$fitness = \frac{1}{TD}. \tag{3.2}$$

3.1.1.4. Crossover In the proposed GA, the search space is limited to the feasible region. Therefore, every individual is always feasible. Consequently, caution is necessary in the crossover and mutation operators, because a simple exchange between two customers can violate time and capacity constraints. Consequently, a more complex operator is necessary to be introduced without bringing a bias in any particular direction. A new crossover strategy is proposed to address the following features:

- to inherit as many routes as possible from each parent, equally;
- to avoid high distance routes when adjustments are necessary to inherit them;
- to avoid excessive customer exchange during the crossover, increasing the function cost;
- to avoid new individuals with number of vehicles higher than their parents.

In order to achieve these objectives the crossover operator described in the Algorithm 2 was written:

Algorithm 2. *Crossover algorithm*

```
function Crossover;
begin
    /* FirstStep: entire routes from Parents are inhered by the offspring */
    /* one by one route are randomly selected from each parent by turn*/
    repeat
        /* From Parent #1 in turn */
        Copy Random Route from Parent1 to the offspring;
        /* From Parent #2 in turn */
        Copy Random Route from Parent2 to the offspring;
    Until (no more inherited route is feasible)
    /* There are no more entire routes from parents individuals possible in the offspring*/
    /* SecondStep: insert unrouted customers, if possible, into inherited routes
```

Insert Unrouted Customers in Current Routes;
/* ThirdStep: Create new routes for remainders customers */
Insert Remainder Customers in new routes using ***stochasticPFIH***;
end; /* end function;

In the first step, the algorithm makes a randomly route choice from each parent individual in turns. After all feasible routes have been inserted in the offspring, the insertion of remainder customers are tested in existing routes (second step). If some customers continue to be unrouted; there is no other option than to insert them in empty vehicles (new routes). In this step, the stochastic PFIH is again applied (Fig. 1).

3.1.1.5. Elitism In order to guarantee that GA never retreats in the quality solution, elitism strategy was adopted. This method consists of bringing the best individual from the current population to the next population. However, to enable mutation over the previous selected individual, two copies of this individual are done. The first copy is subject to mutations, while the second remains intact to the next cycle of evaluation, selection and crossover, as showed in Fig. 2.

3.1.1.6. Mutation A total of eight different operators were used in the mutation phase of the GA proposed. The mutation process is necessary for the insertion of new characteristics in the current population. Not considering mutation, the GA search is limited to a very small area of the total feasible search space. In the present work, some very specialized operators were created to speed up the evolution of the individuals, as explained below.

Mutation_01 (Random customer migration): This operator chooses a vehicle randomly and a random customer associated to it; a migration of this customer to other non-empty vehicle is tried. If the insertion results a feasible route, that is accepted independently of the new function cost (See Fig. 3).

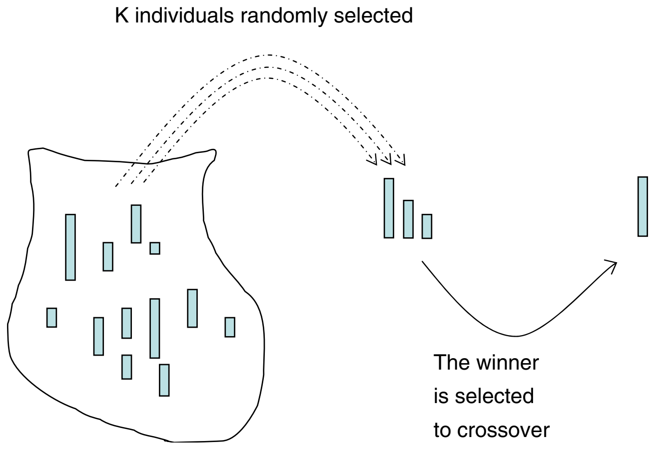

Fig. 1. The k-way tournament selection. After the pool has been chosen, the winner is the highest fitness.

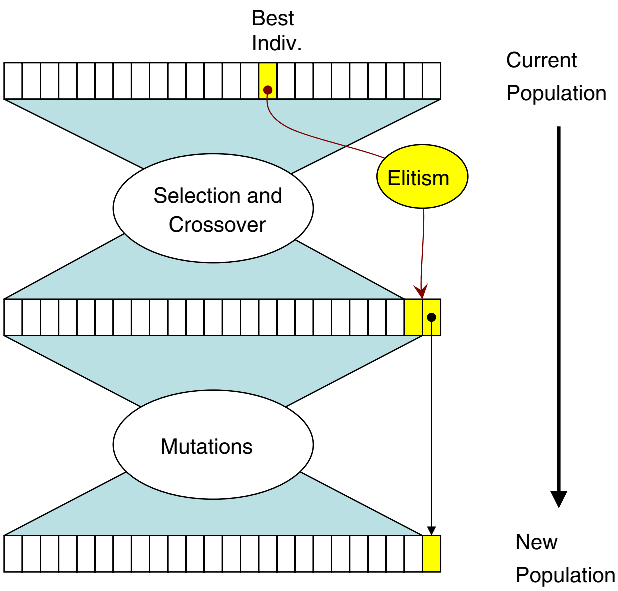

Fig. 2. Steps to complete the next population (Next Pop) from the current one (Current Pop). The elitism is emphasized.

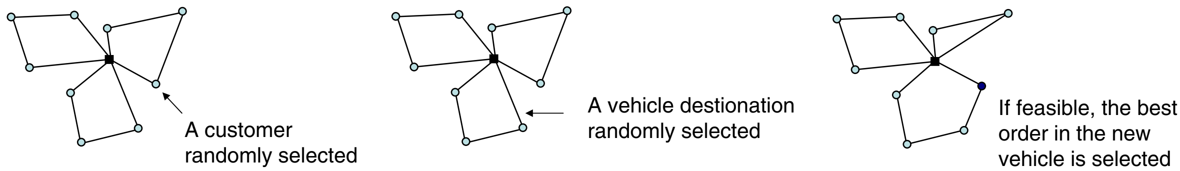

Fig. 3. Mutation_01: Random customer migration.

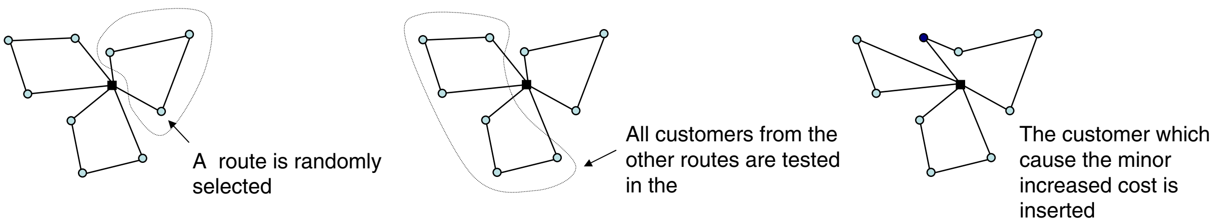

Fig. 4. Mutation_02: Bringing the best customer.

Mutation_02 (Bringing the best customer): This operator chooses a vehicle randomly and searches for the customer from others vehicles, which represents the minimal increased travel distance (See Fig. 4).

Mutation_03 (Re-insertion using stochastic PFIH): This operator chooses a route randomly and applies the stochastic PFIH procedure, as described in section "Initial population" (See Fig. 5).

Mutation_04 (Similar customer exchange): This operator chooses a route randomly and searches for a "similar" time window customer from others vehicles to try an exchange. The "similar" time window is

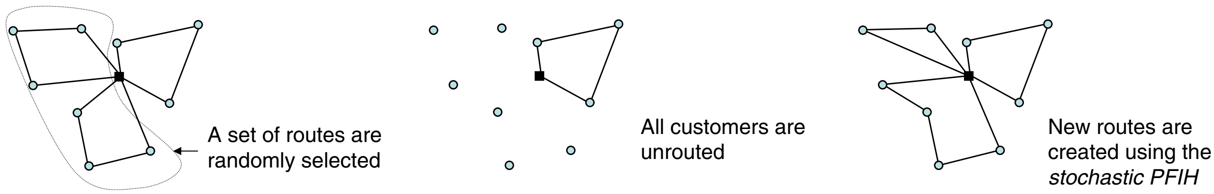

Fig. 5. Mutation_03: Re-insertion using stochastic PFIH.

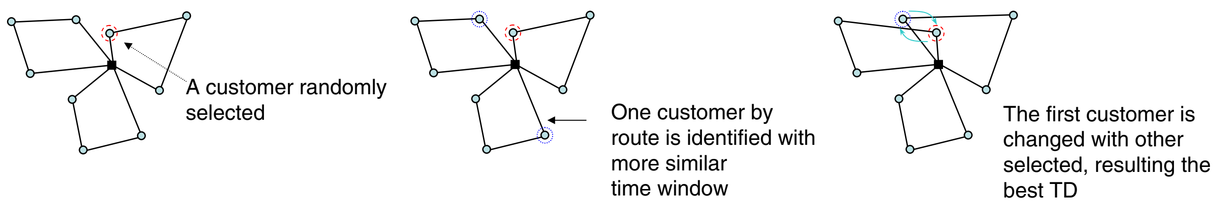

Fig. 6. Mutation_04: Similar customer exchange.

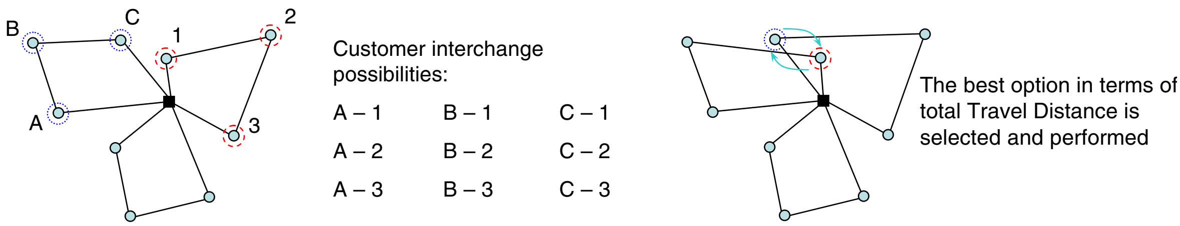

Fig. 7. Mutation_05: Customer exchange with positive gain.

considered as the minimal difference between the earliest time of two different customers, i.e., $a_i - a_j$, where i and j are two customers (See Fig. 6).

Mutation_05 (Customer exchange with positive gain): This operator is very expensive in terms of time complexity, O(nm), where n and m are the number of customers in two routes. Nevertheless, it is very important to improve the GA convergence. It verifies all possibilities to exchange a couple of customers and only carries it out if a reduction in the total travel distance is obtained (See Fig. 7).

Mutation_06 (Merge two routes): This operator chooses two random routes and tries to merge them in a random way. Very often, the remainder customers are inserted in other routes or in a new one. If new route is necessary, the stochastic PFIH is applied (See Fig. 8).

Mutation_07 (Reinserting customer): This operator chooses a random customer, removes it, and re-inserts in a better position, i.e., the position in the same route with a minimal travel distance (See Fig. 9).

Mutation_08 (Route partitioning): This operator chooses a random vehicle, a random customer and divides this route in two others, using that customer as reference (See Fig. 10).

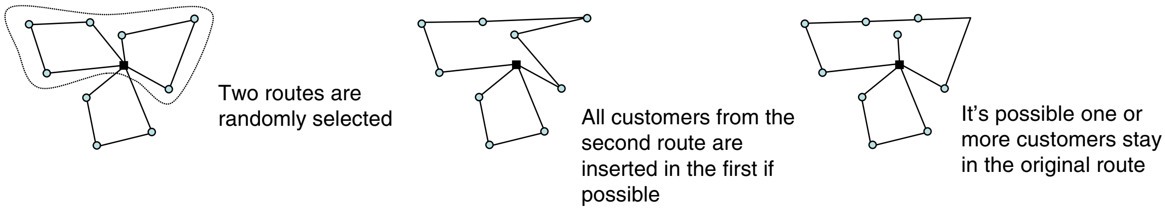

Fig. 8. Mutation_06: Merge two routes.

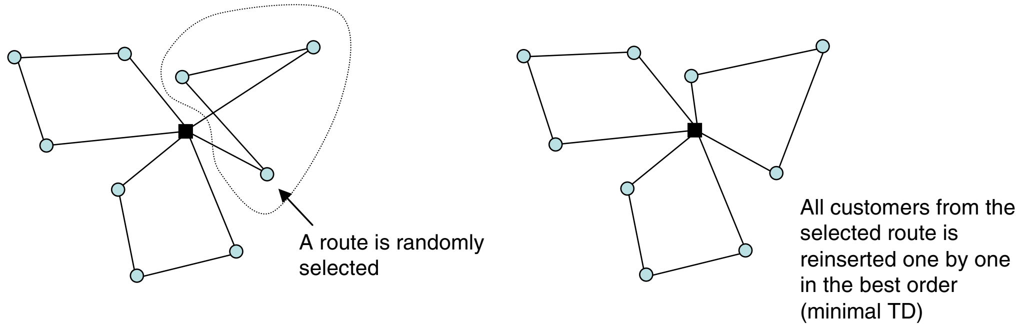

Fig. 9. Mutation_07: Reinserting customer.

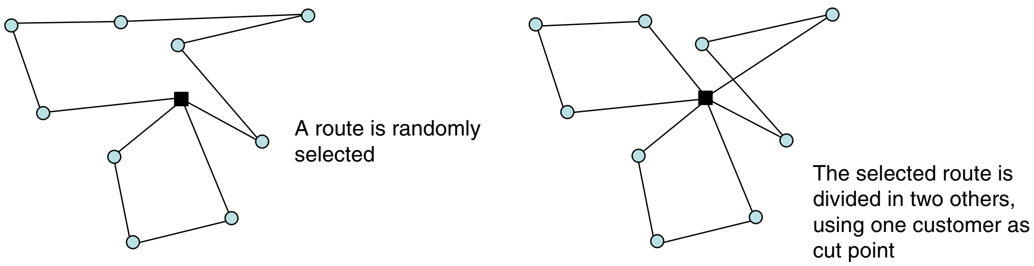

Fig. 10. Mutation_08: Route partitioning.

3.1.2. Initial set of routes using the whole problem

The framework starts producing an initial set of local minima using the GA described. The flow chart is showed in the Fig. 11. All routes from the best individual from each island are included in the subset *R*. In this phase, many (MAX_ISLAND) independent local minimal solutions are obtained.

3.1.3. Improving the set of routes using a reduced problem

The literature shows research efforts, such as the classical Pethal Methods, that propose the partition of the problem into sub-regions, and then treating each sub-region as a reduced-order and independent problem. These sub-region based methods can be useful for problems without capacity and time window constraints. However, for the VRPTW, any artificial sub-region can have the effect of forcing the creation of additional routes in the solution; or of routes with higher travel distances. This occurs in the

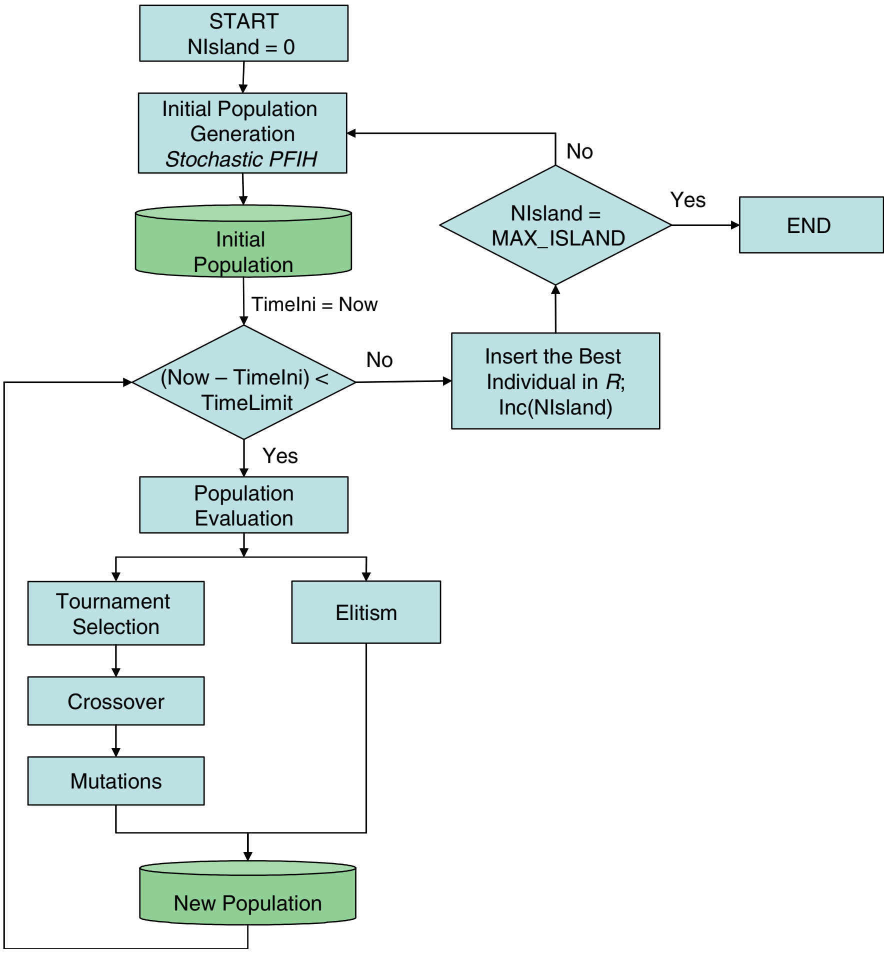

Fig. 11. GA Flow chart proposed. Many evolutions (MAX_ISLAND) are generated and all routes from the best individual are included in the subset R.

VRPTW because the time window constraints obstruct routes according to the physical proximity of the customers.

On the other hand, it may be easier for the GA to find out little imperfections in a subset of routes previously produced, like wrong customer order or changed customers between routes. In this way, additional evolutions are applied using the GA described above, where a subset of customers is eliminated, reducing the problem to 30% in size. The customers are grouped in accordance with the intermediate solution produced using the SPP model over the subset of routes R from the previous phase, described in Section 3.1.2. Again, many islands of evolution are generated; each island uses a different set of routes, randomly chosen. The routes from the SPP solution are used only to decide what customer will be inserted in the subproblem, but new routes are obtained using the stochastic PFIH to generate each initial GA population. The SPP is solved to optimality, using the GLPK MIP software. Fig. 12 shows the flow of this process and Fig. 13 summarizes the phases to generate the subproblem.

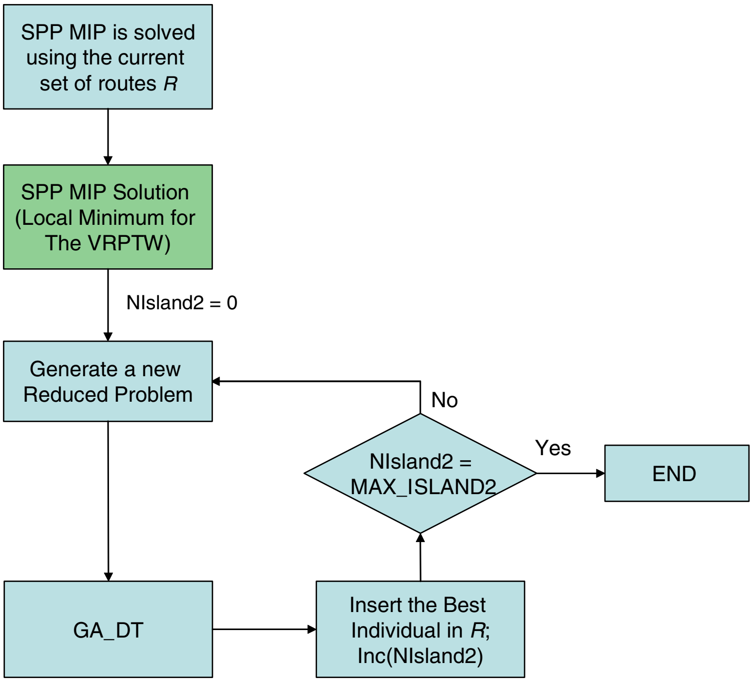

Fig. 12. Flow chart illustrating the subproblem generation and the GA evolution to improve the set of routes R.

3.2. The complete algorithm

In the previous sections, the proposed GA is applied in two different phases to produce quality routes to the set R from Eq. (2.1). These routes are from independent local minimal solutions. Initially handling the whole problem, where all customers are considered, and in a second phase, using a reduced problem. Even though the quality of these produced routes is directly related to the whole solution quality of the GA, it is necessary to produce as many routes as possible, increasing the possibility to find the desired set of routes present in the global optimum.

These two sequential phases can be considered as diversification and intensification steps. The first has a diversification bias because many independent and fast evolution are done, generating many different local minimal solutions. Thus, the SPP model is solved to obtain the best combination of routes in an unique solution. The successive GA evolutions can be considered as an intensification search step because any subproblem generated will require similar routes solutions, once the customers are divided based on the SPP solution. In Fig. 13 it is possible to see this tendency in the set of customers to the subproblem.

The framework is composed by successive cycles of diversification and intensification, i.e., islands of evolution handling the whole problem and the subproblem. Finally, a global set of routes, R_{GLOBAL}, are constructed to provide a global and final SPP MIP solution, obtaining a near global optimal solution. Since the entirely framework can be considered as a column generation heuristic method, it is denominated CGH. All parts together can be showed in Fig. 14.

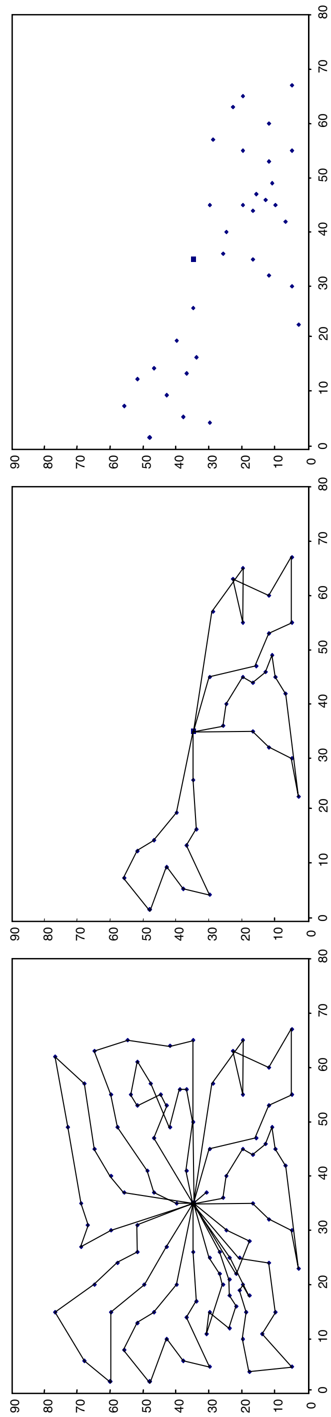

Fig. 13. From left to right, the intermediate solution (SPP MIP over the set R), the set of routes randomly chosen and the subproblem created.

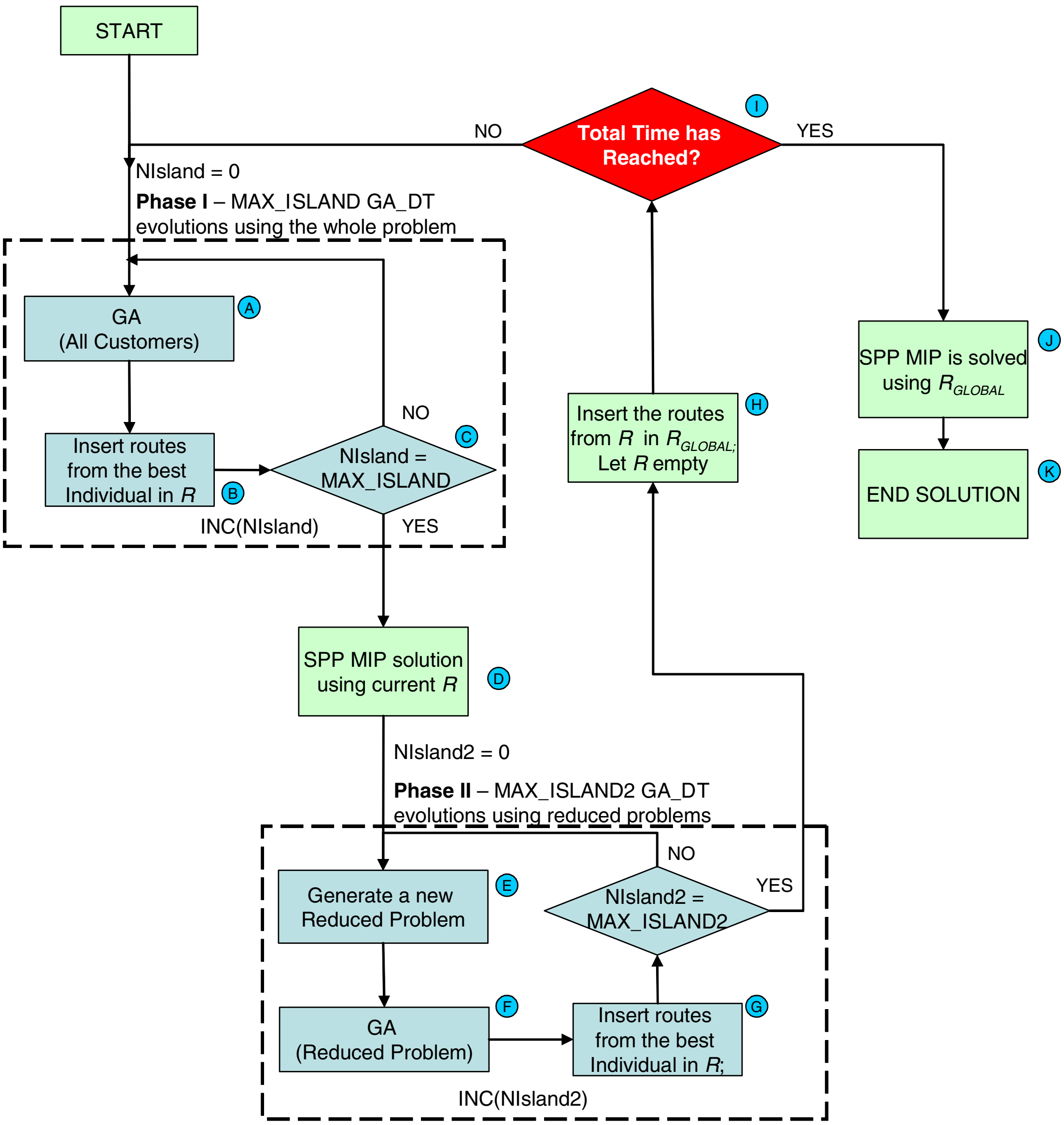

Fig. 14. Putting all parts together: The entire framework of the CGH.

The whole process showed in Fig. 14 can be summarized as follows. At first, MAX_ISLAND islands of GA evolutions are generated considering the original instances of Solomon, represented by the blocks with labels A, B and C. The set of routes R are coming up using the routes from the best individuals from each island, block B. When the number of evolutions NIsland achieves the maximal number of islands (MAX_ISLAND) the SPP is solved using the GLPK MIP package over the set of routes R, block D. Based on the SPP solution, the algorithm decides how the customers will be clustered to generate many different reduced problems. As mentioned before, each route found in the SPP solution has the possibility of 30%

of having its customers inserted in a reduced problem. This step is represented by the block E in Fig. 14. Again, the same GA is executed to optimize travel distance over this reduced problem, block F. Here, the only parameter changed in the GA is the time horizont, reduced to 50%. In the same way, many islands are generated (MAX_ISLAND2), but then with different set of customers, because the routes selected are different each time. When MAX_ISLAND2 is reached, the set R contains additional routes. This step can be considered as an improvement in the former result, because the division of the customers based on the current SPP solution will lead the search in that region. The entire set of routes R is added to the global set R_{GLOBAL} and made empty. If time is not reached (Total Time, block I), the cycle described starts again. Actually, the process can be also interrupted between any GA generations and is not represented in the flow chart to simplify it. When the limited time expires, normally after many cycles, the global set R_{GLOBAL} is used in the final SPP formulation, where all produced routes are considered. The MIP GLPK package is also used to find an integer optimal solution considering this set R_{GLOBAL}. The results in the next section show that this local minimum is very close or even coincident with the global optimal solution.

4. Computational results

The Solomon's test set was proposed in 1987 [11] for three different customers space classes: the classes R1 and R2 have customers randomly disposed; the classes C1 and C2 have customers clustered in some groups; and, finally, the classes RC1 and RC2 contain a subset of customers randomly disposed and the other part clustered. The distance between two customers is the simple euclidean distance. One unit of time is necessary to run one unit of distance by any vehicle. Each customer i has a time window $[a_i, b_i]$, which represents the time interval to arrive in that customer. Different capacity constraints are considered for the vehicle in each class of instance, as well as the demands from the customers. There are a total of 56 instances for each dimension category problem, that are 25, 50 and 100 customers. Because the Solomon's test set represents relatively well different kinds of scenarios, it has been chosen to evaluate many solution proposals in the literature. However, the incomplete definition of the problem has resulted in different assumptions, and consequently it is more difficult to compare the works. The first lack of agreement in the literature is the objective function of the problem. Many different objectives have been chosen in the literature for the Solomon's test set, like total travel distance minimization, number of vehicle minimization, total wait time minimization and the combinations of them. The second important lack of agreement is related to the data type, real or truncated numbers. Many authors justify the use of integer numbers, to reduce the influence of computer hardware, but in other cases the double precision arithmetic has been chosen. However, the instances of Solomon continue to be the best way to evaluate a new approach, and were used in this work. In order to reduce these weaknesses mentioned above and make possible a more complete comparison, both real and integer numbers have been considered.

4.1. General GA parameters adjustment

Appropriated adjustment of parameters in GAs can make a significant difference in terms of performance. Some values can provide very high performance in specific instances while giving premature convergence in others, even over the same kind of problems. Today, robust GAs with very specialized

Table 1
Average time to generate islands of evolution. The stochastic PFIH time is included

Classes	Original problem (s)	Reduced problem (s)
R1	10	02
R2	21	05
C1	10	03
C2	54	11
RC1	09	02
RC2	56	12

and complex operators are developed for many problems. For the VRPTW, these characteristics are very critical, because there is not only one kind of instance but many classes of problems. For example, some instances are very relaxed in terms of time window constraints. In other instances, customer sequences are absolutely necessary to satisfy time restrictions.

In this work it is possible to define a satisfactory set of operators to allow a fast algorithm. Considering the set partitioning problem used to complete the search, there is not much possibility to entrap in the local minimum, once the GA has many opportunities to obtain different solutions.

In Fig. 14, MAX_POP represents the constant used to define the number of generations used in the GA for the original problem. The value of 170 is empirically chosen to MAX_POP. Differently, the number of generation used in the same GA for the reduced problem, MAX_POP2, is 120. Using these parameters it is possible to generate an island of evolution sufficiently fast. In Table 1 it is shown the average time measured by islands of evolution using each Solomon's classes of problems.

The k-way tournament selection process is tested using groups of 2, 3 and 4 individuals. Groups of 3 randomly selected individuals have obtained the best results. It is fixed, before adjusting all other parameters.

The number of times to execute all mutation operators is adjusted empirically as follows. The operators are divided in two blocks, one that increases the diversification search (block A) and another that permits an intensification search (block B). In this work, intensification stands for the operation in a current solution which ensures a positive gain in the travel distance. On the other hand, diversification stands for random moves which can deteriorate the current solution. If a preponderant intensification search is used in the GA, the population generally stops in a premature local minimum. In opposite, an excessive diversification search is equivalent to a random and/or blind search in the feasible solution space.

Initially, the numbers of execution for these blocks applied in each GA generation were decided by feeling, considering basically the time complexity cost of the related algorithm. This procedure has enabled to find the best rate of each block in total, using empirical tests. Tests have been done using different problem types. After the best mutation amount was selected for these two blocks, a fine adjustment is done, changing the number of execution for each specific mutation. These two steps are repeated many times, sequentially, using different problems. Finally, all parameters were fixed to test the solution for all instances. The values of the parameters found can be seen in Tables 2 and 3.

Table 2
Number of executions by mutation operator in each GA generation

Mutation operator	Execution/Generation
Mutation_01 (Random customer migration)	20
Mutation_02 (Bringing the best customer)	10
Mutation_03 (Re-insertion using stochastic PFIH)	1
Mutation_04 (Similar customer exchange)	20
Mutation_05 (Customer exchange with positive gain)	2
Mutation_06 (Merge two routes)	1
Mutation_07 (Reinserting customer)	30
Mutation_08 (Route partitioning)	1

Table 3
Additional parameters used in the GA

GA Parameter (description)	Value
MAX_POP (number of generations in the GA - whole problem)	170
MAX_POP2 (number of generations in the GA- reduced problem)	120
Number of individuals in all Population	31
MAX_EVOL (number of islands for each cycle - whole problem)	8
MAX_EVOL2 (number of islands for each cycle - reduced problem)	25
TIME_LIMIT (Total time limit)	60 min

4.2. Results for total distance minimization

The proposed heuristic method, CGH is evaluated using 56 Solomon's problems with 100 customers. Although many heuristic approaches have been tested using this Solomon's test set, there is no information about previous heuristic methods considering total travel distance as the first objective together with integer numbers (first decimal truncated) to make possible the comparison with exact method solutions. Although data type seems to be a simple detail, it can change significantly the results because it can decide if a route will be feasible or not. The inputs of the problem like coordinates, capacity, customer demands and others are multiplied by 10 and all subsequent calculations are truncated to integers.

Table 4 shows the results obtained by the CGH for all Solomon's problems with 100 customers and the exact results obtained by optimization methods, when they are known. The objective function applied is only the total TD. The NV found in the solutions were presented only as reference. Results emphasized in bold in Table 4 represent the exact solution reached by CGH. The blank cells in the exact solution column mean that the exact solution for these problems has not been obtained yet. The CGH results represent the best after 3 algorithm runs. The time limit to the execution is 60 min, including many intermediate MIP solutions (block D in Fig. 14). These SPP problems, although always executed to optimality, have few routes and do not represent an expressive time spender. The final SPP solved (block J in Fig. 14) can spend a bit more time, in spite of the set R_{GLOBAL} still continue small. The necessary time in this step are not included in the limited time of 60 min. No limit is imposed to any MIP steps in the algorithm. Even the last stage generally the results of the SPP problem come up after few seconds.

Table 4
Known exact solutions from the literature and the best CGH results (3 runs). Emphasized in bold the CGH results means the optimal solution was reached. NV—Number of Vehicles, TD–Total Distance

Prob.	Known optimal solution		Best CGH		Average CGH		Prob.	Known optimal solution		Best CGH		Average CGH	
	NV	TD	NV	TD	NV	TD		NV	TD	NV	TD	NV	TD
C101	10	827.3	**10**	**827.3**	**10**	**827.3**	R201	8	1143.2	8	1165.3	7	1172.8
C102	10	827.3	**10**	**827.3**	**10**	**827.3**	R202			6	1053.9	6	1064.4
C103	10	826.3	**10**	**826.3**	**10**	**826.3**	R203			5	910.1	5	919.6
C104	10	822.9	**10**	**822.9**	**10**	**822.9**	R204			4	770.8	5	790.6
C105	10	827.3	**10**	**827.3**	**10**	**827.3**	R205			5	972.5	5	986.8
C106	10	827.3	**10**	**827.3**	**10**	**827.3**	R206			5	934.4	5	944.0
C107	10	827.3	**10**	**827.3**	**10**	**827.3**	R207			4	860.1	5	868.3
C108	10	827.3	**10**	**827.3**	**10**	**827.3**	R208			4	744.1	4	756.1
C109	10	827.3	**10**	**827.3**	**10**	**827.3**	R209			5	888.5	5	903.1
C201	3	589.1	**3**	**589.1**	**3**	**589.1**	R210			5	952.2	5	955.6
C202	3	589.1	**3**	**589.1**	**3**	**589.1**	R211			5	815.5	5	821.5
C203	3	588.7	**3**	**588.7**	**3**	**588.7**	RC101	15	1619.8	15	1623.5	16	1628.4
C204			3	590.6	3	599.4	RC102	14	1457.4	14	1458.2	14	1468.4
C205	3	586.4	**3**	**586.4**	**3**	**586.4**	RC103	11	1258.0	12	1272.5	12	1282.3
C206	3	586.0	**3**	**586.0**	**3**	**586.0**	RC104			10	1137.6	10	1141.7
C207	3	585.8	**3**	**585.8**	**3**	**585.8**	RC105	15	1513.7	**15**	**1513.7**	15	1518.8
C208	3	585.8	**3**	**585.8**	**3**	**585.8**	RC106			13	1373.9	13	1374.2
R101	20	1637.7	**20**	**1637.7**	**20**	**1637.7**	RC107			12	1209.3	12	1221.3
R102	18	1466.6	**18**	**1466.6**	**18**	**1466.6**	RC108			11	1115.2	11	1124.3
R103	14	1208.7	14	1210.2	14	1211.2	RC201	9	1261.8	8	1274.3	8	1281.0
R104			11	979.3	11	983.1	RC202	8	1092.3	7	1119.5	7	1139.8
R105	15	1355.3	**13**	**1355.3**	14	1357.1	RC203			5	958.0	5	959.5
R106	13	1234.6	**13**	**1234.6**	13	1235.4	RC204			4	807.6	4	822.2
R107	11	1064.6	11	1067.3	11	1069.0	RC205	7	1154.0	**7**	**1154.0**	6	1161.1
R108			10	946.2	10	947.3	RC206			6	1080.4	6	1084.6
R109	13	1146.9	**13**	**1146.9**	13	1150.1	RC207			7	1005.6	6	1015.9
R110	12	1068.0	12	1079.4	12	1081.5	RC208			5	820.5	6	830.5
R111	12	1048.7	**12**	**1048.7**	12	1055.0							
R112			10	962.3	10	967.5							

The results show that many previously known optimal solutions from exact methods have been reached, 24 out of 33 (72.7%). Globally, the TD average results are only 0.29% higher considering the 33 instances where the optimal solutions are known. In RC101, RC103 and RC201, the numbers of used vehicle in the solution are not the same. Larsen [9] suggests heuristic methods using TD as the only objective and the truncated number criterion, making it easier to compare the results between the algorithms. The results presented here, seem to be the first in this way, and consequently current heuristic benchmark under these conditions.

It is possible to improve the knowledge of the heuristic performance under the same arithmetic assumptions and the same objective adopted in exact works.

Table 5
Best total distance (TD) published heuristic results and CGH results using real arithmetic (double precision). The results are emphasized in bold when CGH overcomes the previous best solutions

Problem	Previous best TD solution			CGH		Problem	Previous best TD solution			CGH	
	Ref.	NV	TD	NV	TD		Ref.	NV	TD	NV	TD
R101	[20]	19	1645.79	**20**	**1642.870**	R201	[21]	4	1252.37	**9**	**1148.483**
R102	[1]	17	1486.12	**18**	**1472.620**	R202	[1]	_4_	_*1088.07_	**7**	**1049.737**
R103	[1]	_14_	_*1213.62_	**14**	**1213.620**	R203	[22]	3	939.54	**5**	**900.080**
R104	[1]	_10_	_*982.01_	11	986.096	R204	[23]	2	825.52	**4**	**772.330**
R105	[1]	14	1377.11	**15**	**1360.783**	R205	[5]	3	994.42	**6**	**970.886**
R106	[22]	12	1251.98	**13**	**1241.518**	R206	[24]	3	906.14	**5**	**898.914**
R107	[25]	10	1104.66	**11**	**1076.125**	R207	[1]	_3_	_*814.78_	**4**	**834.930**
R108	[26]	9	960.88	**10**	**948.573**	R208	[22]	2	726.75	**3**	**723.610**
R109	[21]	11	1194.73	**13**	**1151.839**	R209	[20]	3	909.16	**6**	**879.531**
R110	[1]	_11_	_*1080.36_	12	1092.347	R210	[22]	3	939.34	**7**	**932.887**
R111	[5]	10	1096.72	**12**	**1053.496**	R211	[23]	2	892.71	**5**	**787.511**
R112	[1]	_10_	_*953.63_	10	960.675						
C101	[1]	10	828.94	10	**828.94**	C201	[1]	3	591.56	3	**591.56**
C102	[1]	10	828.94	10	**828.94**	C202	[1]	3	591.56	3	**591.56**
C103	[1]	10	828.06	10	**828.06**	C203	[1]	3	591.17	3	**591.17**
C104	[1]	10	824.78	10	**824.78**	C204	[1]	3	590.60	3	**590.60**
C105	[1]	10	828.94	10	**828.94**	C205	[1]	3	588.88	3	**588.88**
C106	[1]	10	828.94	10	**828.94**	C206	[1]	3	588.49	3	**588.49**
C107	[1]	10	828.94	10	**828.94**	C207	[1]	3	588.29	3	**588.29**
C108	[1]	10	828.94	10	**828.94**	C208	[1]	3	588.32	3	**588.32**
C109	[1]	10	828.94	10	**828.94**						
RC101	[1]	_15_	_*1623.58_	16	1639.968	RC201	[22]	4	1406.91	**9**	**1274.537**
RC102	[1]	_13_	_*1477.54_	**14**	**1466.840**	RC202	[1]	_4_	_*1165.57_	**8**	**1113.526**
RC103	[27]	11	1261.67	11	1264.707	RC203	[28]	3	1049.62	**5**	**945.960**
RC104	[29]	10	1135.48	10	1135.520	RC204	[22]	3	798.41	4	799.670
RC105	[26]	13	1629.44	**16**	**1518.600**	RC205	[22]	4	1297.19	**7**	**1161.810**
RC106	[1]	_12_	_*1384.92_	**13**	**1377.352**	RC206	[20]	3	1146.32	**7**	**1059.886**
RC107	[25]	11	1230.48	**12**	**1212.830**	RC207	[23]	3	1061.14	**7**	**976.396**
RC108	[30]	10	1139.82	**11**	**1117.526**	RC208	[31]	3	828.14	**5**	**795.391**

Additionally, the same heuristic CGH is tested making use of real arithmetic (double precision) and the results are also compared with real arithmetic heuristic solutions in the literature, although few works have considered Solomon's benchmark problems minimizing TD. The results are presented in Table 5 together with the best solution known in the literature. It is possible to see that the algorithm proposed continues to be very competitive in terms of total TD. Although some works have considered TD as the first objective [5,6,30], the best previous result comes from heuristic methods minimizing the number of vehicles as the first objective, becoming difficult to compare with CGH results. Following, Table 6 shows the results by problem class, including results from other relevant works in the literature. In this table, the increased percentages from the best results (TD%) are presented for each author. The CGH has reduced significantly the best results. Another interesting fact is that the previous best results considering TD

Table 6
The results in the literature using real arithmetic for each problem class and the CGH results. The best TD results were considered

		R1	C1	RC1	R2	C2	RC2
Published Best	NV	12.25	10.00	13.00	2.91	3.00	6.00
	TD	1195.63	828.38	1360.37	935.35	589.86	1094.16
Taillard [30]	NV	12.17	10.00	11.50	2.82	3.00	3.38
	TD	1209.35	828.38	1389.22	980.27	589.86	1117.44
	TD%	+1.1	0.0	+2.1	+4.8	0.0	+2.1
Rousseau [5]	NV	12.83	10.00	12.50	3.18	3.00	3.75
	TD	1201.10	828.38	1370.26	966.94	594.01	1113.29
	TD%	+0.5	0.0	+0.7	+3.4	+0.7	+1.7
Tan [6]	NV	13.83	10.11	13.63	3.82	3.25	7.00
	TD	1260.71	859.81	1447.06	1058.52	617.10	1169.41
	TD%	+5.4	+3.8	+6.4	+13.2	+4.6	+6.9
Kilby [32]	NV	12.67	10.00	12.12	3.00	3.00	3.38
	TD	1200.33	830.75	1388.15	966.56	592.24	1133.42
	TD%	+0.4	+0.3	+2.0	+3.3	+0.4	+3.6
Homberger [21]	NV	12.00	10.00	11.63	2.73	3.00	3.25
	TD	1226.38	828.38	1392.57	969.95	589.86	1144.43
	TD%	+2.6	0.0	+2.4	+3.7	0.0	+4.6
Berger [2]	NV	11.92	10.00	11.50	2.73	3.00	3.25
	TD	1221.10	828.48	1389.89	975.43	589.93	1159.37
	TD%	+2.1	0.0	+2.2	+4.3	0.0	+6.0
CGH	NV	13.33	10.00	13.00	4.64	3.00	6.00
	TD	**1196.8**	**828.38**	**1341.7**	**899.9**	**589.86**	**1015.9**
	TD%	0.10%	0.00%	−1.37%	−3.79%	0.00%	−7.15%

do not match with the best results considering NV only for 10 problems indicated with (*) in Table 5 as summarized in [4]. All these results came from the Rochat work [1], which reports a better TD but worse NV, with one more vehicle.

In Table 7 is possible to see the gain obtained with the SPP formulation over the set of routes *R*. Those results confirm the viability of this hybrid CGH approach for TD minimization in the VRPTW.

5. Conclusions

This paper presents a contribution to overcome two of the main weaknesses in the VRPTW literature. Firstly, the behaviour of a heuristic method is compared against exact methods, considering the number of global optimal solutions found using the same assumptions used in the exact methods: truncated calculation and total distance minimization. Secondly, few papers have addressed distance minimization using heuristic methods. The results obtained by CGH are very expressive, establishing many new

Table 7
The best Genetic Algorithm (GA) parcial solutions found and the final CGH results. The last column reports the percentages gain with the SPP MIP formulation

Prob.	Best GA		CGH (Final SPP)		SPP TD reduction		Prob.	Best GA		CGH (Final SPP)		SPP TD reduction	
	NV	TD	NV	TD	NV	TD (%)		NV	TD	NV	TD	NV	TD (%)
C101	10	828.937	10	828.937	0	0.00%	R201	9	1179.790	9	1148.483	0	−2.65%
C102	10	828.937	10	828.937	0	0.00%	R202	8	1071.070	7	1049.737	−1	−1.99%
C103	10	828.065	10	828.065	0	0.00%	R203	7	932.760	5	900.080	−2	−3.50%
C104	10	824.770	10	824.770	0	0.00%	R204	4	807.380	4	772.330	0	−4.34%
C105	10	828.937	10	828.937	0	0.00%	R205	6	1005.340	6	970.886	0	−3.43%
C106	10	828.937	10	828.937	0	0.00%	R206	6	935.240	5	898.914	−1	−3.88%
C107	10	828.937	10	828.937	0	0.00%	R207	3	875.300	4	834.930	1	−4.61%
C108	10	828.937	10	828.937	0	0.00%	R208	3	748.350	3	723.610	0	−3.31%
C109	10	828.937	10	828.937	0	0.00%	R209	6	900.650	6	879.531	0	−2.34%
C201	3	591.557	3	591.557	0	0.00%	R210	7	955.160	7	932.887	0	−2.33%
C202	3	591.557	3	591.557	0	0.00%	R211	4	808.560	5	787.511	1	−2.60%
C203	3	591.173	3	591.173	0	0.00%	RC101	16	1660.100	16	1639.968	0	−1.21%
C204	3	590.599	3	590.599	0	0.00%	RC102	14	1482.910	14	1466.840	0	−1.08%
C205	3	588.876	3	588.876	0	0.00%	RC103	12	1297.580	11	1264.707	−1	−2.53%
C206	3	588.493	3	588.493	0	0.00%	RC104	11	1174.620	10	1135.520	−1	−3.33%
C207	3	588.286	3	588.286	0	0.00%	RC105	15	1572.400	16	1518.600	1	−3.42%
C208	3	588.324	3	588.324	0	0.00%	RC106	14	1417.460	13	1377.352	−1	−2.83%
R101	20	1646.840	20	1642.870	0	−0.24%	RC107	13	1261.100	12	1212.830	−1	−3.83%
R102	18	1482.740	18	1472.620	0	−0.68%	RC108	11	1129.310	11	1117.526	0	−1.04%
R103	15	1230.040	14	1213.620	−1	−1.33%	RC201	9	1286.830	9	1274.537	0	−0.96%
R104	11	1009.540	11	986.096	0	−2.32%	RC202	9	1148.840	8	1113.526	−1	−3.07%
R105	14	1380.870	15	1360.783	0	−1.45%	RC203	6	1001.250	5	945.960	−1	−5.52%
R106	13	1249.400	13	1241.518	0	−0.63%	RC204	5	826.150	4	799.670	−1	−3.21%
R107	11	1079.730	11	1076.125	0	−0.33%	RC205	8	1168.220	7	1161.810	−1	−0.55%
R108	10	965.580	10	948.573	0	−1.76%	RC206	8	1084.300	7	1059.886	−1	−2.25%
R109	13	1156.260	13	1151.839	0	−0.38%	RC207	6	999.260	7	976.396	1	−2.29%
R110	12	1106.990	12	1092.347	0	−1.32%	RC208	5	806.870	5	795.391	0	−1.42%
R111	12	1056.870	12	1053.496	0	−0.32%							
R112	11	996.790	10	960.675	−1	−3.62%							

benchmark solutions. It is possible to see that the number of vehicles and total distance are very competitive objectives, because the CGH results overcome the previous minimal TD results, generally increasing the number of vehicles.

References

[1] Rochat Y, Taillard ED. Probabilistic diversification and intensification in local search for vehicle routing. Journal of Heuristics 1995;1:147–67.

[2] Berger J, Barkaoui M, Bräysy O. A route-directed hybrid GA approach for the VRPTW. Information Systems and Operational Research 2003;41:179–94.

[3] Homberger J, Gehring H. Two evolutionary metaheuristics for the vehicle routing problem with time windows. INFOR, vol. 37, 1999. pp. 297–318.

[4] Best known solutions identified by heuristics for Solomon's benchmark problems. TOP—improved optimization methods in transportation logistics. www.top.sintef.no/vrp/index.htm

[5] Rousseau LM. Gendreau M, Pesant G. Using constraint-based operators with variable neighborhood search to solve the vehicle routing problem with time windows. Presented at the CP-AI-OR'99 Workshop, February 25–26, University of Ferrara, Italy, 1999.

[6] Tan KC, Lee LH, Zhu QL, Ou K. Heuristic methods for vehicle routing problem with time windows. Artificial Intelligence in Engineering 2001;15:281–95.

[7] Achutan N, Cacetta L, Hill S. An improved branch-and-cut algorithm for the capacitated vehicle routing problem. Tansportation Science 2003;37:153–69.

[8] Ralphs T, Kopman L, Pulleyblank W, Trotter Jr. L. On the capacitated vehicle routing problem. Mathematical Programming 2003;94:343–59.

[9] Larsen J. Parallelization of the vehicle routing problem with time windows. Ph.D. Thesis, IMM-PHD-1999-62, Department of Mathematical Modelling, Technical University of Denmark, Lyngby, Denmark, 1999.

[10] Kohl N, Desrosiers J, Madsen OBG, Solomon MM, Soumis F. 2-Path cuts for the vehicle routing problem with time windows. Transportation Science 1999;33(1):101–16.

[11] Solomon MM. Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research 1987;35(2):254–65.

[12] Whitley D. A genetic algorithm tutorial. Statistics and Computing 1994;4:65–85.

[13] Fraser AS. Simulation of genetic system by automatic digital computers. Australian Journal of Biological Science 1957;10:484–91.

[14] Box GEP. Evolutionary operation: a method of increasing industrial productivity. Applied Statistics 1957;6:81–101.

[15] Forgel LJ, Owens e AJ, Walsh MJ. Artificial intelligence through simulated evolution. New York: Wiley; 1966.

[16] Rochenberg I. Evolutions strategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution. Stuttgart: Frommann-Holzboog; 1973.

[17] Holland JH. Adaptation in natural and artificial system. Ann Arbor, Michigan: The University of Michigan Press; 1975.

[18] Darwin C. On the origin of species. 1st ed, MA: Harvard University Press; 1859.

[19] Thangiah SR, Osman IH, Sun T. Hybrid genetic algorithm simulated annealing and tabu search methods for vehicle routing problem with time windows, Technical Report 27, Computer Science Department, Slippery Rock University, 1994.

[20] Homberger J. Verteilt-parallele Metaheuristiken zur Tourenplanung. Wiesbaden: Gaber; 2000.

[21] Homberger J, Gehring H. Parallelization of a two-phase metaheuristic for routing problems with time windows. Journal of Heuristics 2002;8:251–76.

[22] Mester D. An evolutionary strategies algorithm for large scale vehicle routing problem with capacitate and time windows restrictions. Working Paper, Institute of Evolution, University of Haifa, Israel, 2002.

[23] Bent R, Van Hentenryck P. A two-stage hybrid local search for the vehicle routing problem with time windows. Technical Report CS-01-06, Department of Computer Science, Brown University, 2001.

[24] Schrimpf G, Schneider J, Stamm-Wilbrandt H, Dueck G. Record breaking optimization results using the ruin and recreate principle. Journal of Computational Physics 2000;159:139–71.

[25] Shaw P. A new local search algorithm providing high quality solutions to vehicle routing problems. Working Paper, University of Strathclyde, Glasgow, Scotland, 1997.

[26] Berger J, Barkaoui M, Bräysy O. A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. Working paper, Defense Research Establishment Valcartier, Canada, 2001.

[27] Shaw P. Using constraint programming and local search methods to solve vehicle routing problems. In Principles and Practice of Constraint Programming—CP98, Pisa, Italy, 1998.

[28] Czech ZJ, Czarnas P. A parallel simulated annealing for the vehicle routing problem with time windows. Proceedings of the tenth euromicro workshop on parallel, distributed and network-based processing, Canary Islands, Spain, 2002, p. 376–83.

[29] Cordeau J-F, Laporte G, Mercier A. A unified tabu search heuristic for vehicle routing problems with time windows. Working Paper CRT-00-03, Centre for Research on Transportation, Montreal, Canada, 2000.

[30] Taillard E, Badeau P, Gendreau M, Geurtin F, Potvin JY. A tabu search heuristic for the vehicle routing problem with time windows. Transportation Science 1997;31:170–86.

[31] Ibaraki T, Kubo M, Masuda T, Uno T, Yagiura M. Effective local search algorithms for the vehicle routing problem with general time windows. Working Paper, Department of Applied Mathematics and Physics, Kyoto University, Japan, 2001.
[32] Kilby P, Prosser P, Shaw P. Guided Local Search for the Vehicle Routing Problem. Second International Conference on Metaheuristics—MIC97, 1997.