Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Scheduling (CI-Sched 2007)

A Robust Method for the VRPTW with Multi-Start Simulated Annealing and Statistical Analysis

H. C. B de Oliveira, G. C. Vasconcelos, G. B. Alvarenga, R. V. Mesquita, M. M de Souza

Abstract— **Vehicle Routing Problems have been extensively analyzed to reduce transportation costs. More particularly, the Vehicle Routing Problem with Time Windows (VRPTW) imposes the period of time of customer availability as a constraint, a very common characteristic in real world situations. Using minimization of the total distance as the main objective to be fulfilled, this work implements an efficient hybrid system which associates non-monotonic simulated annealing to hill climbing with random restart (multi-start). Firstly, the algorithm is compared to the best results published in the literature for the 56 Solomon instances. Then, it is shown how statistical methods – analysis of variance and linear regression – can be used to determine the significance degree of the system's parameters to obtain an even better and more reliable performance.**

I. Introduction

Costs with goods transportation are calling special attention in the last decades, since logistic expenses minimization is a big concern for many companies. According to [1], the costs related to people and goods transportation are very significant and are growing rapidly, motivated by the continuing increase of business complexity experienced today. Studies suggest that from 10% to 15% of the final value of the traded goods correspond to its transportation cost [7]. Many studies found in the literature of the Vehicle Routing Problem (VRP) have attempted to contribute to practical advances in this field [22][24]. Since in the real world this problem has to consider many constraints and particularities, some parameters have to be considered for a closer approximation to the market situations. One of such constraints is the load capacity of the vehicle and the time window in which the customers must be visited. This class of problems is known as the Vehicle Routing Problem with Time Windows (VRPTW) and has been the most popular class of the VRP [19].

II. THE Vehicle Routing Problem with Time Windows

The VRPTW is a combinatorial optimization problem. This is a particular case of the well know Vehicle Routing Problem (VRP), introduced by [5]. In the VRP, the vehicle fleet must to visit and deliver a service to a set of customers. Every vehicle starts and finishes at a unique depot. For each pair of customers, or customer and the depot, there is an associated cost. This cost denotes how expensive it is for a vehicle to move from one costumer to another, with the constraint that each customer must be visited exactly once. Additionally, each customer demands a specific number of goods (denoted as the weight of the load). For each vehicle in a fleet, there is an upper limit of load capacity supported. In the basic case (considered here) all the vehicles are of the same type and have the same capacity. Then, basically, the objective of the VRP is to find a set of customers attended for each vehicle of the fleet in order to minimize the transportation costs [9].

With a further complexity, in the VRPTW each customer has an associated time window that determines an interval within which a vehicle has to begin and to finish the service to that customer. In the VRP as well as in the VRPTW several types of optimization objectives have been investigated in the literature. In particular, the total distance traveled is one of the most typical cost measures to be minimized by a given algorithm. Another parameter taken as target is to find the minimum set of possible routes to solve the problem, and to minimize the distance as a second objective. Yet some other authors also consider the time minimization for attending all the costumers as the objective to be completed.

With the aim of minimizing the total traveled distance, such as in the work developed here, De Backer et al. investigated iterative improvement techniques within a Constraint Programming (CP) framework [3]. The improvement techniques are coupled to Tabu Search (TS) and Guided Local Search (GLS) to avoid the search of being trapped into local minima. The CP system is used only as a background operator to check the validity of the solutions found and to speed up legality checks of improvement procedures. Riise and Stølevik [21] studied GLS and Fast Local Search (FLS) combined with simple move operators that relocate single tasks. Kilby et al. [20] introduced a deterministic GLS that use local search operators (2-opt, relocate, exchange and 2-opt*) with a so-called best-acceptance strategy (for details, see [23]). Alvarenga [2] studied the use of a Set Partitioning (SP) formulation after generating several solutions through a given genetic algorithm (GA). The routes of the generated solutions by the GA are combined by the exact method of Set Partitioning, finding the best combination of the routes without violating the constraints of the VRPTW.

In the present paper, a different approach based on the combination of simulated annealing and hill climbing with random restart (multi-start) is proposed.

III. Complexity of the VRPTW

As defined in [13], an instance of an optimization problem is a pair *(F,c)* where *F* is any set, the domain of feasible points; c

H. C. B. Oliveira, G. C. Vasconcelos and M. M. de Souza are with the Center for Informatics, Federal University of Pernambuco - Brazil (e-mail: {hcbo,gcv,mms2} @ cin.ufpe.br).

G. B. Alvarenga and R. V. Mesquita are with the Department of Computer Science, Federal University of Lavras - Brazil (e-mail: guilherme @ dcc.ufla.br, rafael @ swfactory.com.br).

is a cost function, i.e. a mapping $c : F \rightarrow R^1$ (1)

The problem is to find an $f \in F$ for which $c(y) \leq c(f) \quad \forall \quad f \in F$ (2).

Such a point *y* is called a global optimal solution to a given instance or simply an optimal solution.

As the VRPTW is a problem known as NP-hard [10], only solutions of reduced order instances can be found using exact algorithms [4]. The impossibility of guaranteeing optimal solutions ($f \in F$) occurs because the set of functions *F* is huge, given the non-determinism in the search of all the possible solutions to the problem, preventing the sweeping of every solution in a polynomial time. To overcome this problem, heuristics and meta-heuristics are frequently employed to find sub-optimal solutions which are both performance effective and feasible to be determined in non-polynomial time [2].

IV. A Hybrid System (HS) to the VRPTW

This work aims at finding a solution $f \in F$ such that *y* is a good approximation to the optimal solution *y*, based on a combination of simulated annealing with a hill-climbing strategy. A set of techniques were considered with the principle of generating a solution that could provide good results in the diverse set of possible situations of the VRPTW. A strategy based on simulated annealing and hill climbing took place inspired on the capability of simulated annealing to both evolve solutions to a given problem and escape from local minima and on the capacity of hill climbing to refine initially defined solutions. To complete the method, a technique called 'random restart' [14] of the system is applied in order to cope with the idea of producing solutions to varied configurations of the VRPTW, returning the best solution from the executed restarts. Such strategy performs multiple system restarts with the association of simulated annealing and hill climbing and finds better results by diminishing the variance between the different executions of the system (consequently enhancing the robustness of the method). The idea behind the operation of the hybrid system can be viewed in Fig. 1.

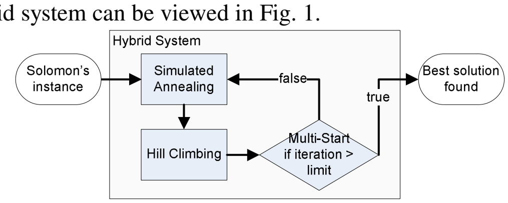

Fig. 1. Illustration of the Hybrid System

A. Simulated Annealing (SA) to Solve the VRPTW

Simulated Annealing is a probabilistic meta-heuristic algorithm, proposed originally in [8] (being a local search method), that accepts search movements that temporarily produces degradations in a current solution found to a problem as a way to escape from local minima.

This meta-heuristic is based on a natural method which uses an analogy to the thermodynamics simulating the cooling of a set of heated atoms, in a operation known as annealing [8] (the term and operation of annealing are widely use in metallurgy).

In its formal description, the simulated annealing begins its search from a random initial solution. The iteration loops that characterize the main procedure randomly generate in each iteration only one neighbor *s´* of the current solution *s*. The variation *Δ* for the value of the objective function *f(x)* is tested for each neighbor generation. To test this variation, the following calculation takes place:

$$\Delta = f(s') - f(s) \qquad (3)$$

If the value of *Δ* is less than *0* (zero), then the new solution *s´* is automatically accepted to replace *s*. Otherwise, accepting the new solution *s´* will depend on the probability established by the Metropolis Criteria, which is given by:

$$e^{-\Delta / T} \qquad (4),$$

where *T* is a temperature parameter, a key variable for operation of the method. The Metropolis Criteria accepts, with higher probability, solutions which have lower values of *Δ*. Higher values of *Δ* will have lower chances if compared to lower values of *Δ*. The higher the temperature, the higher is the probability of accepting the solution *s´* as the new solution, explaining the algorithm analogy to the solid cooling.

1) Starting Point for the Simulated Annealing

For constructing an initial solution to the SA algorithm, this work used the algorithm known as *Push-Forward Insertion Heuristic* (PFIH) [15]. As cited in [9], the PFIH has an efficient constructive strategy for calculating the cost of a new customer in a route. This cost is computed according to its geographic position, the end of its time window and the angle between it and the central depot. For a better understanding consider Fig. 2 as a current solution before the insertion of the customer C5.

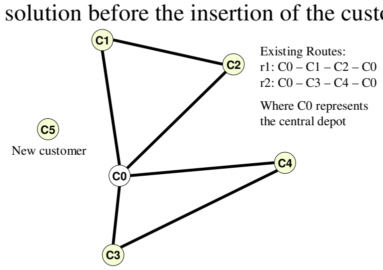

Fig. 2. Current incomplete solution before the insertion of a new customer

The quality of the possible solutions is first checked with the insertion of the customer (C5) in each possible edge of the graph representing the current solution for the VRPTW. Then, the edge in which the insertion of the customer represents the lowest cost is chosen (with respect to the total distance) for the inclusion of the new customer in the route.

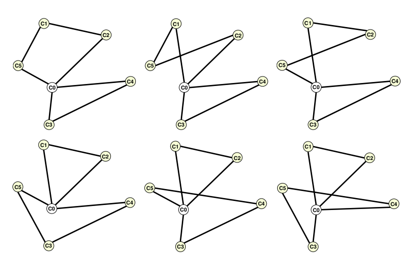

Fig. 3. Feasible solutions found by the PFIH algorithm for the introduction of a new customer

Besides the computation of the costs, the insertion of the customer is also examined to guarantee that it does not violate the restrictions of the VRPTW (see Section 2). If any of the existing (from the smallest to the biggest cost) route solutions (shown in Fig. 3) does not violate any constraint of capacity, vehicle load or customer attendance time, then this route becomes the definite solution for inclusion of the new costumer (Fig. 4). Otherwise, the current routes are discarded and a new route is created for representing the new costumer.

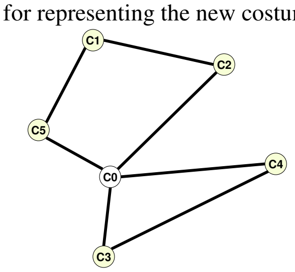

Fig. 4. Solution selected as the best option when inserting the customer C5

The order in which the customers are inserted into the VRPTW solution by the PFIH algorithm directly defines the quality of the final solution produced by the method. Keeping this in mind, Solomon developed in his work [15] a heuristic to determine the order in which the customers should be considered into the solution, according to the cost Equation:

$$C_i = -\alpha\, d_{oi} + \beta\, l_i + \gamma\left[\left(\frac{p_i}{360}\right) d_{0i}\right] \qquad (5)$$

Where: α = 0.7; β = 0.1; γ = 0.2; d_{0i} = the distance between the central depot and customer i; b_i = the upper limit of the time window for arrival of customer i; p_i = polar coordinate angle of the customer i, with respect to the central depot. The constant values for α, β e γ were defined empirically such as in [15].

From the first chosen customer, the remaining customers are tested one by one with respect to each possible route solution for construction, according to the cost Equation 6. The position and the customer that resulted in the lowest increase in the total traveled distance, without violating the time window capacity, are chosen. After there are no more customers to insert in the route under construction, this particular route is closed and the same process is restarted again with a new empty route, being the first customer the one with the lowest cost according to Equation 6 (among those customers yet to be routed).

Aiming at introducing flexibility for starting the SA algorithm from different positions in the search space, this work introduced a variation in the original cost formula, where, instead of been treated as constants in the original algorithm, the α, β and γ elements are turned into PFIH parameters. The new values change at each execution being captured by a normal distribution $N(\mu,\sigma)$, with an average in the points ($\alpha\mu = 0.7$; $\beta\mu = 0.1$; $\gamma\mu = 0.2$), suggested as optimal by Solomon, and with a deviation by the unit ($\sigma = 1$). With this variation of the heuristic orders, the customers being inserted by the PFIH algorithm maintain a good arrangement because the averages are centered on the optimal values obtained empirically and the small variations cause perturbations that create different initial solutions for the SA. This modification in the PFIH algorithm proposed here was inspired on the work of [11] that implements the change of the heuristic through taking the values of α, β and γ from a uniform distribution changing from 0 to 1.

2) *Neighborhood Operators Applied to the Simulated Annealing*

Given a feasible point $f \in F$ in a particular problem, it is useful in many situations to define a set $N(f)$ of points that are 'close' in the same sense to the point f [13], where F represents the set of any solution that satisfies the problem. For example, if $F = R^n$, then the set of points within a fixed Euclidean distance provides a natural neighborhood solution for F [13].

As this work focus on ordered lists without repetitions (routes with its respective customers), it is guaranteed according to [6] that for this type of representation four basic permutation operators can describe a generic way to capture the neighborhood of a solution *f*. These operators perform permutations between the elements in order to capture the neighborhood of a solution *f* and were originally implemented as mutation operators in evolutionary algorithms [6]. These have been adapted in this research to be employed in the SA for the neighborhood definition.

The first neighborhood operator is the simplest and is called Swap Mutation (see the work of [6]). It is described to find a neighborhood to a solution for the Traveling Salesman Problem (TSP) and is called "2-change" in [13]. The *swap* (*f*) operator can be described as: *swap(f)* = {g : $g \in F$ e g can be obtained from f swapping 2 customers (*c1, c2*) of any routes (*r1, r2*)}.

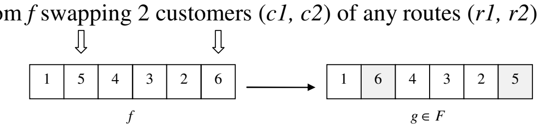

Fig. 5. Application of the swap neighborhood operator in the same route of solution *f*

The second neighborhood operator, also based on random changes, is called Insert Mutation and is formally defined as: *insert(f)* = {g : $g \in F$ and g may be obtained by removing one customer from any route of f and inserting it again in any position of any route of f}.

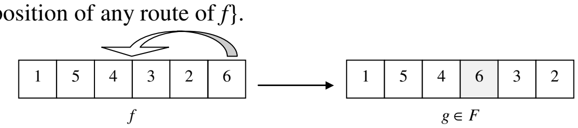

Fig. 6. Example showing the application of the Insert neighborhood operator in a route of solution *f*

The third operator is the Scramble Mutation, which is another random operator defined as: *scramble(f)* = {g : $g \in F$ and g may be obtained by choosing any continuous sequence q of customers in a route r chosen randomly from f and mixing the customers of q in order to create a sequence q' which will substitute q in r }

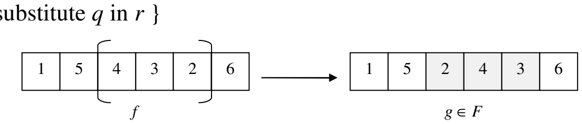

Fig. 7. Example of application of the neighborhood operator Scramble on a route *r* of the solution *f*

The fourth operator, totally random, is based on the customers' inversion and can be defined as: *inversion(f)* = {g : $g \in F$ e g may be obtained by choosing a sequence s of customers in a route *r,* randomly chosen from *f,* and after inverting them systematically for the generation of a new sequence *s´* that will replace *s* in *r*}.

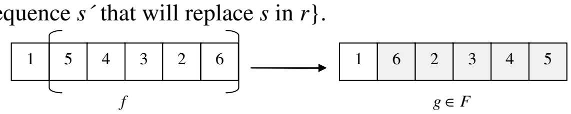

Fig. 8. Example of the application of the neighborhood operator inversion in the same route *r* of solution *f*

A fifth operator, called OP5 (based on the work of [12]), was also employed. This is also a mutation operator for evolutionary algorithms that uses specific information on the problem (heuristics) to navigate in the solution space constructed by the instantiated VRPTW. Firstly, *m* customers are withdrawn from each route of solution *f*. The number of withdrawn customers varies for each route *r* and is chosen by selecting a value from a uniform distribution that varies from 0 to the number of customers present in *r*.

After selecting the customers withdrawn from *f* (creating an incomplete solution *h*), all the selected customers are inserted back into *h* through the *PFIH* method, until a complete solution for the VRPTW is found. This neighborhood operator never generates a new solution *f´* that violates any constraints, because the *PFIH* algorithm does not allow such possibility.

The *f´* solution generated through the *f* solution after the application of any neighborhood operator is accepted only if *f´* satisfies every constraint of the VRPTW. In any case of constraint violation, the solution *f* is kept into the SA for the next iteration of the system. This possibility may occur with the operators *swap, insert, scramble* and *inversion*.

For each iteration of the SA, an operator is chosen by withdrawing a positive number from a uniform distribution, which varies from 1 to *k*, being *k* the number of operators of the system (in this work *k* is equal to 5). The operators are stored in a vector of *k* positions and the one whose index is raffled is applied.

3) Temperature Control

The temperature *T* used initially on the system was set up to 100 and, at every 100 iterations, was reduced to 95% of its current value: $T_{(t+1)} = 0.95 \times T_t$ **(6)**

In addition to the temperature decrease, the algorithm can also produce a temperature increase, characterizing the non-monotonic aspect of the *T* variable. This increase (by the unit) occurs when there is no improvement in the best solution in the last 1000 iterations of the system. In the total, 30000 iterations were used for each execution of the simulated annealing. Another example of efficient use of non-monotonic temperature control is reported in [16].

B. Hill Climbing (HC) Strategy

After the conclusion of the SA process, the hill climbing (HC) strategy takes places to compose the hybrid solution (HS) proposed in this work. The motivation to introduce the HC strategy comes from the observation that the solution given by the SA may be found in the system when its temperature is considerably high, and, in this case, the neighborhood close (which may contain a better solution) to the best solution of the SA will probably never be explored. This happens because, when the temperature is high, the Metropolis Criteria will tend to easily carry out a drastic locomotion in the solution space to look for the solution and worse solutions may be accepted by the method with high probability. Taking that into account, it was considered the application of the HC strategy to find a local minimum equal or better than the returned solution of the SA method. When the best solution of the SA is found under low temperatures, however, this local search (HC) turns itself to be unnecessary since it is basically executed by the SA method itself, considering that the Metropolis Criteria will only accept worse solutions with very low probabilities.

The HC process is then executed three times at the end of the SA method, each execution corresponding to 1000 iterations, to consider that different executions of a hill climbing may drive to different regions of the search space (solutions). This is the case because the implemented hill climbing is non deterministic and the execution with the best result is returned by the method. The neighborhood operators are those described in Section A.2.

C. Random Restart

With respect to the issue of random initialization, different works have followed different paths. Some works choose the idea of performing a small number of short iterations, based on [14] which poses that, "if the problem is NP-complete, then in all likelihood we cannot do better than exponential time". Although theoretically, in such situations, there may be an exponential number of local minima where the solution may get stuck on, fortunately, in practice, a reasonably good solution can be found after a small number of iterations. Other works, however, were based on the strategy of executing for each instance the evolutionary algorithm for a long period of time (40 minutes on average) [11].

In the work of [12] it was identified that short executions of the system (1 minute and 17 seconds on average), repeated many times, produced more robust results for the VRPTW. Since simulated annealing is also a stochastic algorithm, the quality of the final solutions over a number of runs shows a certain variance, the same strategy was adopted in implementing the proposed HS (SA with HC). 30 system restarts were applied and the solution that presented the shorter total distance of all the system restarts was considered as the HS solution to the VRPTW.

V. Test Base

There are many publications using heuristics and meta-heuristics in the resolution of the VRPTW, making easier comparisons and analysis on new proposed approaches. For discovering the quality and robustness of the different algorithms, these are frequently applied over the Solomon instances [15].

Likewise, the tests of this work were executed over the 56 Solomon instances with 100 customers each. The data descriptions and sources may be found on the internet (http://neo.lcc.uma.es/RADIAEB/WebVRP/data/instances/solomon/solomon_100.zip).

The Solomon instances are divided into six classes: R1, R2, C1, C2, RC1 and RC2. The R1 and R2 instances present customers with random Euclidean coordinates. Instances C1 and C2 present customers grouped in clusters. Instances of type RC1 and RC2 present a mix of the two first characteristics (spare and clustered). One common characteristic of the types R1, C1 and RC1 is that their instances impose that few customers have to be attended by each vehicle, introducing the need for more vehicles to attend all the demand. The types R2, C2 e RC2 present few vehicles in the solution to attend a great number of customers in each route.

VI. RESULTS

A. *Best known results*

For ease of visualization, Tables I, II, III, IV, V and VI point out if the hybrid system proposed obtains equal or better results when compared to the best individual published results with respect to the total traveled distance minimization for the VRPTW. The instances marked with ** denote the situations where the best previous individual result for that instance (picked up from many different authors) were overcame by the proposed method, whereas those marked with * represent the cases where the results of the method paired the previous results. Each table in this section considers the total traveled distance minimization as the main focus for the VRPTW.

TABLE I
BEST RESULTS FOUND IN THE R1 PROBLEM CLASS

		Other works			This work	
	Instance	Vehicles	Distance	Work	Vehicles	Distance
*	**R101**	**20**	**1642.88**	**[1]**	**20**	**1642.88**
	R102	18	1472.62	[1]	18	1475.35
	R103	14	1213.62	[17]	15	1222.68
	R104	11	986.10	[2]	11	990.78
	R105	15	1360.78	[2]	15	1363.74
	R106	13	1241.52	[2]	13	1244.58
	R107	11	1076.13	[2]	11	1081.88
	R108	10	948.57	[2]	10	952.37
	R109	13	1151.84	[2]	12	1153.89
	R110	11	1080,36	[17]	12	1087.94
	R111	12	1053,50	[2]	12	1053.80
	R112	10	953,63	[17]	11	973.34

TABLE II
BEST RESULTS FOUND IN THE R2 PROBLEM CLASS

		Other works			This work	
	Instance	Vehicles	Distance	Work	Vehicles	Distance
**	**R201**	**5**	**1148.48**	**[2]**	**8**	**1147.80**
**	**R202**	**9**	**1042.35**	**[12]**	**8**	**1039.32**
**	**R203**	**5**	**876.94**	**[12]**	**6**	**874.87**
**	**R204**	**4**	**736.66**	**[12]**	**5**	**735.80**
**	**R205**	**5**	**960.07**	**[12]**	**5**	**954.16**
**	**R206**	**4**	**887.90**	**[12]**	**5**	**884.25**
**	**R207**	**4**	**811.93**	**[12]**	**4**	**797.99**
**	**R208**	**3**	**707.01**	**[12]**	**4**	**705.62**
**	**R209**	**5**	**860.11**	**[12]**	**5**	**860.11**
**	**R210**	**6**	**912.48**	**[12]**	**5**	**910.98**
**	**R211**	**4**	**761.75**	**[12]**	**4**	**755.82**

TABLE III
BEST RESULTS FOUND IN THE C1 PROBLEM CLASS

		Other works			This work	
	Instance	Vehicles	Distance	Work	Vehicles	Distance
*	**C101**	**10**	**828.94**	**[17]**	**10**	**828.94**
*	**C102**	**10**	**828.94**	**[17]**	**10**	**828.94**
*	**C103**	**10**	**828.06**	**[17]**	**10**	**828.06**
*	**C104**	**10**	**824.78**	**[17]**	**10**	**824.78**
*	**C105**	**10**	**828.94**	**[17]**	**10**	**828.94**
*	**C106**	**10**	**828.94**	**[17]**	**10**	**828.94**
*	**C107**	**10**	**828.94**	**[17]**	**10**	**828.94**
*	**C108**	**10**	**828.94**	**[17]**	**10**	**828.94**
*	**C109**	**10**	**828.94**	**[17]**	**10**	**828.94**

TABLE IV
BEST RESULTS FOUND IN THE C2 PROBLEM CLASS

		Other works			This work	
	Instance	Vehicles	Distance	Work	Vehicles	Distance
*	**C201**	**3**	**591.56**	**[17]**	**3**	**591.56**
*	**C202**	**3**	**591.56**	**[17]**	**3**	**591.56**
*	**C203**	**3**	**591.17**	**[17]**	**3**	**591.17**
*	**C204**	**3**	**590.60**	**[17]**	**3**	**590.60**
*	**C205**	**3**	**588.88**	**[17]**	**3**	**588.88**
*	**C206**	**3**	**588.49**	**[17]**	**3**	**588.49**
*	**C207**	**3**	**588.29**	**[17]**	**3**	**588.29**
*	**C208**	**3**	**588.32**	**[17]**	**3**	**588.32**

TABLE V
BEST RESULTS FOUND IN THE RC1 PROBLEM CLASS

		Other works			This work	
	Instance	Vehicles	Distance	Work	Vehicles	Distance
	RC101	15	1623.58	[17]	16	1642.83
	RC102	14	1466.84	[2]	15	1480.46
	RC103	11	1261.67	[18]	13	1308.64
	RC104	10	1135.48	[19]	11	1162.75
	RC105	16	1518.60	[1]	15	1534.60
	RC106	13	1377.35	[1]	13	1386.82
	RC107	12	1212.83	[1]	12	1247.53
	RC108	11	1117.53	[1]	11	1135.87

TABLE VI
BEST RESULTS FOUND IN THE RC2 PROBLEM CLASS

		Other works			This work	
	Instance	Vehicles	Distance	Work	Vehicles	Distance
**	**RC201**	**8**	**1267.27**	**[12]**	**9**	**1266.11**
*	**RC202**	**8**	**1096.75**	**[12]**	**8**	**1096.75**
**	**RC203**	**5**	**941.31**	**[12]**	**5**	**926.89**
**	**RC204**	**4**	**788.66**	**[12]**	**4**	**786.38**
**	**RC205**	**7**	**1161.32**	**[12]**	**7**	**1157.55**
**	**RC206**	**7**	**1059.88**	**[2]**	**6**	**1056.21**
**	**RC207**	**5**	**970.78**	**[12]**	**6**	**966.08**
	RC208	4	779.84	[12]	4	780.72

B. *Comparisons between different works*

Table VII summarizes the comparisons of the proposed algorithm with the best works ([20],[21],[3] and [2]) for the Solomon's benchmarks that consider the total traveled distance as the main objective function. The columns represent the algorithm whereas the lines show the average number of vehicles and the total traveled distance for each class. For each algorithm, the average results with respect to Solomon's benchmarks are reported through NV (number of vehicles) and TD (total distance). CNV and CTD indicate the cumulative number of vehicles and cumulative total distance over all 56 test problems.

TABLE VII
COMPARISONS BETWEEN DIFERENT WORKS THAT OPTIMIZE THE TOTAL TRAVELED DISTANCE OF THE VRPTW

Class \ Work		[20]	[21]	[3]	[2]	SA	This work
R1	NV	12.67	13.92	14.17	13.25	12.40	13.33
	TD	1200.33	1211.22	1214.86	1183.38	1287.00	1186.94
R2	NV	3.00	4.91	5.27	5.55	3.2	5.36
	TD	966.56	917.54	930.18	899.90	1052.00	878.79
C1	NV	10.00	10.56	10.00	10.00	10.00	10.00
	TD	830.75	846.88	829.77	828.38	937.00	828.38
C2	NV	3.00	3.88	3.25	3.00	3.0	3.00
	TD	592.24	598.70	604.84	589.86	684	589.86
RC1	NV	12.13	13.75	14.25	12.88	12.10	13.25
	TD	1388.15	1399.76	1385.12	1341.67	1471.00	1362.44
RC2	NV	3.38	5.63	6.25	6.50	3.4	6.13
	TD	1133.42	1055.61	1099.96	1015.90	1307.00	1004.59
All classes	CNV	423	502	508	489	423	488
	CTD	57423	56682	56998	55134	63145	55020

Fig. 9 illustrates the cumulative distance for all the works taken for performance comparison.

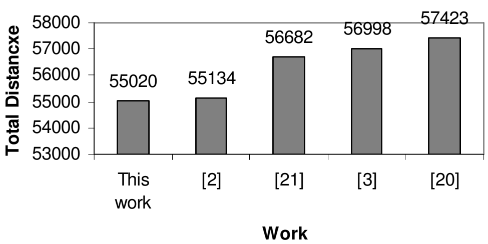

Fig. 9. Cumulative total distance for all the Solomon's instances

VII. Parameter Adjustment

Just like many other methods, the original experiments taken with the HS were conducted on a trial and error basis for optimal parameter adjustment. Although this alone was capable of producing improved results when compared to previous works, such manual setting up may take time and, sometimes, not attain the expected results.

Some works apply optimization algorithms to adjust their own parameters. Eiben and Smith [6], for example, consider the auto-adaptive control as the most appropriated for evolutionary algorithms, since it finds optimal parameters in every state of the algorithm, estimated according to the population dynamics in a given epoch. Alvarenga [2] implements a genetic algorithm that searches optimal values for the parameter of its hybrid system. In this case, the parametric correlations may be identified implicitly by the optimizer system but, after adjustment, it is not possible to say which parameters induced the response of the system, and what important correlations produced the good results.

This work, conversely, uses analysis of variance [25] to identify which parameters and respective interactions influence more on the quality of the HS. Thus, through linear regression [26], it is found a polynomial that identifies the tendencies of these influences onto the target variable, making possible the proposal of an optimized configuration for the parameters of the HS.

A. Analysis of variance

The analysis of variance (ANOVA) can be conducted by testing the hypothesis that different configurations of the HS (i.e. varying its parameters) produce the same effect [25]. The hypothesis (H_0) that does not exist any difference in the effects of the treatments on an experiment can be defined as $H_0 = \tau_1 = \tau_2 = \ldots = \tau_i$, where *H0* represents the "nullity hypothesis" and τ_i represents the effect of treatment *i* on the response of the system (in this work, τ_i represents a specific parameter configuration). The parameters studied corresponded to the percentage of application (*P*) of the OP5 to find a neighbor solution, the reduction factor (*FR*) of the temperature in the SA, and the re-heating temperature of the system (*TR*), which characterizes the non-monotonic nature of the SA. The OP5 operator was examined because of its stronger impact on the method due to its $O(n^2)$ complexity, as opposed to the other operators which have complexity O(1). Each of those parameters was tested with pre-specified levels. *P* was tested with values 0.1%, 0.2%, 0.3% and 0.4%; *FR* was tested with values 0.985, 0.990, 0.995 and 0.999; and *TR* was tested with values 15 and 25.

The experimental framework was set up with 32 parametric combinations, based on the values considered for the HS parameters, and each combination was repeated five times, totalizing 160 samples for the analysis of variance.

B. Linear Regression

Linear regression is an analysis technique that uses the relation between two or more quantitative variables to define a mathematical model, such that the effect of a given variable, may be predicted by means of another variable (or other variables) [26]. In experimental analysis, the most frequently used mathematical model for explaining the effects of the treatments in the response variable is the polynomial model. This model was employed in this work.

VIII. Further Experimental Results

The experiments carried out for evaluating the parameters of the HS were conducted with the RC208 instance of the Solomon's benchmark.

A. Results on analysis of variance

The analysis of variance showed that, amongst the four studied parameters, the percentage of application of the operator OP5 in the algorithm (*P)* and the reduction factor (*FR*) of the temperature in the SA, indeed influence the quality of the hybrid system. At the same time, the only parametric interaction that induces a statistically significant change in the HS quality is that between *P* and *FR* (the significance level used was 5%). TABLE 8 shows the results for the different factors examined.

TABLE 8
Synopsis of the Analysis of Variance on the Parameters of the HS

Factor	*p-value*
P	**1.163×10^{-11}**
FR	**9.710×10^{-05}**
TR	0.517474
P:FR	**0.008894**
P:TR	0.202591
FR:TR	0.457425
P:FR:TR	0.385875

The interpretation of the results is that if the *p*-value is less than 0.05, the nullity hypothesis (H_0) is denied and, therefore, there is an impact of the parameter or parametric interaction on the final distance (target variable) reached by the HS. To assure the validity of the application of the ANOVA method, the Shapiro-Wilk test [27] was employed on the ANOVA residuals and, with 95% of confidence, the test pointed out normality of the residuals (*p-value=0.07*).

B. *Linear regression results*

The linear regression technique was applied (only) to the significant parameters indicated by the ANOVA: *P and FR.* Fig. 10 and Fig. 11 illustrate, respectively, the influence of those parameters on the minimization of the distance found by the HS algorithm for the VRPTW problem.

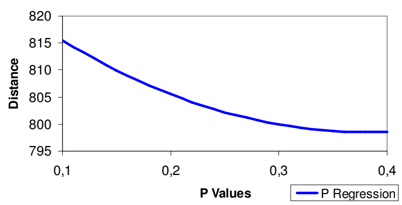

Fig. 10. Linear regression showing the behavior of the *distance* as a function of *P,* through a quadratic polynomial model

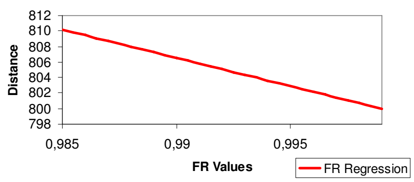

Fig. 11. Linear regression showing the behavior of the *distance* as a function of *FR,* through a quadratic polynomial model

C. *Parametric interaction*

Similarly, Fig. 12 relates the influence of the interaction between the significant pair of parameters (*P* and *FR*) on the mean distance found by the HS algorithm. The analysis shows that, on average, the best parametrical configuration tested for *P* and *FR* were P=0.4 and FR=0.995, respectively.

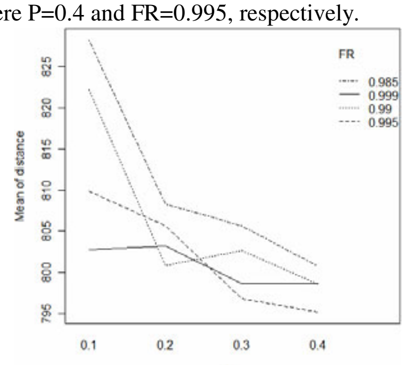

Fig. 12. Mean *distance* found by the HS algorithm as a function of the interaction between parameters *P* and *FR*

D. *Optimal distance minimization with the adjusted parameters*

Confidence intervals (CI) and the average time of execution of four different parametric configurations of the HS algorithm were then considered for observing the impact of the statistical analysis on the performance of the method, as opposed to the manual parameter setting. The first configuration (A1) refers to the optimal manual tuning used before the analysis [P=0.1; FR=0.990; TR=25]. The second configuration (A2) refers to the best combination tested in this work (Section C): [P=0.4; FR=0.995; TR=25]. The third configuration (A3) was an extrapolation of the *P* values following the tendency shown by the linear regression (Fig. 10), with a new value tested for *P*: [P=1.0; FR=0.995; 25]. And, finally, the fourth configuration (A4) was defined with extrapolated values for *P* in the direction where the distance is minimized: [P=5.0; FR=0.995; TR=25]. All the samples A1, A2, A3 e A4 have the same size (30).

1) *Confidence Intervals*

The confidence intervals of the samples A1, A2 A3 e A4 are shown in Fig. 13. The results of the confidence intervals follow the same tendency suggested by the linear regression curve (Fig. 10). The confidence level used was of 95%.

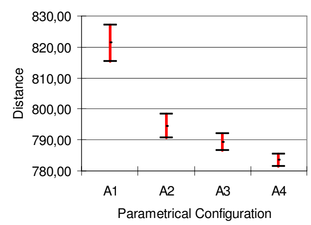

Fig. 13. Confidence intervals for the four configurations

2) *Execution Time*

The average execution time for the samples A1, A2 A3 and A4 are shown in Fig. 14.

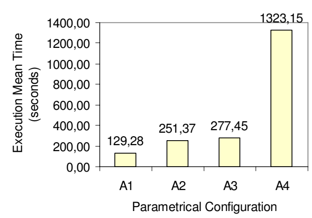

Fig. 14. Execution Time for the four configurations

IX. The Best Results for the RC208 Instance

Table 2 presents the best literature results for the RC208 instance of the Solomon's benchmark for comparison with the results attained by the HS, before and after parameter optimization. The results shown correspond to the minimum distance found (*best*), the average distance ($\bar{x}$), the standard deviation (*s*) and the average execution time for each algorithm. It is seen that the performance reached after statistical analysis (A4 sample) outperformed both the results found with the manual (best) tuning and the results (best) found in the literature.

Table 9 Comparison between the best results published for RC208 instance

work / measure	[28]	[2]	Manual tuning	Statistical tuning
best	828,14	795.40	780.72	778.93

work / measure	[28]	[2]	Manual tuning	Statistical tuning
$\bar{x}$	unknown	814.90	785.50	783.42
s	unknown	28.4	3.0	4.77
mean time (seconds)	~10560	~2700.00	~660.00	1324.15

X. CONCLUSIONS

This work presented a HS that combines simulated annealing with non-monotonic temperature control, random start and hill climbing for the optimization of the total traveled distance in the VRPTW. Its main gain, in relation to previous works, is a substantial improvement in solving the R2 problem classes (R2, C2 and RC2) of the Solomon's benchmark. The solutions found by the HS for each instance contains few routes, and those by themselves, contain many customers for attendance. In the type 2 classes, this work obtained 17 new best results when compared to the ones found in the literature on minimization of the total traveled distance, and equaled other 10 best results (obtaining success in all but one instance (RC208)) out of the 28 tested. With respect to the classes C1, R1 and RC1, (where the solutions for each instance contain many routes and few customers for attendance), this work only paired the best results of the C1 class and the instance R101, being inferior in 19 out of the 29 tested instances.

It was shown that the performance of the method could be further enhanced through a consistent statistical analysis as opposed to the typical trial-and-error process (for the instance RC208 of the Solomon's benchmark, the total distance of the VRPTW was minimized to 778.93 Euclidean's units, the best result reached so far in the problem). By increasing the value of the *P* parameter (which refers to the probability of application of the operator OP5), following the tendency of the linear regression in *P*, the system turns out to be more robust and reliable (Fig. 13). By the fact that the operator OP5 has quadratic complexity, the execution time increases substantially with *P* (Fig. 14), and so does the quality of the method. It is the role of the user when applying this hybrid system, to decide what percentage of the operator should be employed, taking into consideration the time available for finding a solution.

REFERENCES

[1] G. B. Alvarenga. G. R. Mateus. A Two-Phase Genetic and Set Partitioning Approach for the Vehicle Routing Problem with Time Windows. Fourth International Conference on Hybrid Intelligent Systems (HIS04). IEEE Computer Society Press. (2004).

[2] G. B. Alvarenga. A Hybrid algorithm for the static and dynamic vehicle routing problems with time Windows. PhD Thesis. Department of Science Computer. Federal University of Minas Gerais .Brazil (2005).

[3] De Backer, B. and V. Furnon, "Meta-heuristics in Constraint Programming Experiments with Tabu Search on the Vehicle Routing Problem", presented at the Second International Conference on Metaheuristics (MIC´97), Sophia Antipolis, France (1997).

[4] T. H. Cormen; C. E. Leiserson; R. L. Rivest; Introduction to algorithms. The MIT Press. (1999).

[5] G. B. Dantzig. R. H. Ramser; The Truck Dispatching Problem Management Science. 6. 80 (1959).

[6] E. Eiben; J. E. Smith. Introduction to Evolutionary Computing. Natural Computing Series. MIT Press. Springer. Berlin. (2003).

[7] G.F.King; C.F. Mast; Excess Travel: Causes. Extent and Consequences; Transportation Research Record 1111. 126-134. (1997).

[8] Kirkpatrick. S.. Gellat. D. Vecchi. M. P.. *Optimizations by Simulated Annealing*. *Science* v. 220. pp. 671-680. 1983.

[9] J. Larsen; Parallelization of the vehicle routing problem with time windows. Phd Thesis. Department of Mathematical Modeling. Technical University of Denmark. (1999).

[10] J. Lenstra; A. Rinnooy Kan; Complexity of vehicle routing and scheduling problems. Networks 11. 221-227. (1981).

[11] H.C.B. de Oliveira. M.M. de Souza. G.B. Alvarenga. R.M.A. Silva. "An adaptation of a genetic algorithm for the vehicle routing problem with time windows". Infocomp Journal of Computer Science. pp. 51-58. (2004).

[12] H.C.B. de Oliveira. G.C. Vasconcelos. G.B. Alvarenga. An evolutionary approach for the vehicle routing problem with time windows. XXXVII SBPO – Brazilian Symposium on Operations Research. Gramado - Brazil (2005).

[13] Papadimitriou. C. H; Steiglitz. K. "Combinatorial Optimization – Algorithms and Complexity". Dover Publicationsm INC. (1982).

[14] Stuart J. Russell and Peter Norvig. editors. *Artificial Intelligence: A Modern Approach*. Prentice Hall. 2003.

[15] M. M. Solomon; Algorithms for the vehicle routing problem and scheduling problem with time window constraints. (1987).

[16] S. R. Thangiah. I. H. Osman. T. Sun. 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).

[17] Y. Rouchat and E. D. Taillard; Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. Journal of Heuristics 1. 147-167. (1995).

[18] P. Shaw; Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems. Principles and Practice of Constraint Programming - CP98. Lecture Notes in Computer Science. 417-431. M. Maher and J.-F.Puget (eds). Springer-Verlag (1998).

[19] J. F. Cordeau. G. Laporte. and A. Mercier; 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).

[20] Kilby, P., P. Prosser and P. Shaw, "Guided Local Search for the Vehicle Routing Problem with Time Windows", in META-HEURISTICS Advances and Trends in Local Search Paradigms for Optimization, S. Voss, S. Martello, I.H. Osman and C. Roucairol (eds), 473−486, Kluwer Academic Publishers, Boston. (1999)

[21] Riise, A., M. Stølevik, "Implementation of Guided Local Search for the Vehicle Routing Problem", SINTEF Internal report STF42 A99013, SINTEF Applied Mathematics, Norway. (1999)

[22] Bräysy, O. and M. Gendreau. Metaheuristics for the Vehicle Routing Problem with Time Windows. Internal Report STF42 A01025, SINTEF Applied Mathematics, Department of Optimization, Norway. (2001)

[23] Bräysy, O. and M. Gendreau, "Route Construction and Local Search Algorithms for the Vehicle Routing Problem with Time Windows", Internal Report STF42 A01024, SINTEF Applied Mathematics, Department of Optimisation, Norway. (2001)

[24] Bräysy, O., W. Dullaert and M. Gendreau. Evolutionary Algorithms for the Vehicle Routing Problem with Time Windows. Internal Report STF90 A04406, SINTEF ICT, Department of Optimization, Norway. (2004)

[25] Chambers, J. M., Freeny, A., Heiberger, R. M. *Analysis of variance; designed experiments*. Chapter 5 of Statistical Models in S eds J. M. Chambers and T. J. Hastie, Wadsworth & Brooks/Cole. (1992).

[26] Searle, S. R. *Linear models*. New York: J. Wiley. (1971).

[27] Patrick Royston (1995) A Remark on Algorithm AS 181: The W Test for Normality. *Applied Statistics,* 44, 547-551. (1995).

[28] T. Ibaraki, M. Kubo, T. Masuda, T. Uno and M. Yagiura, "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).