A comparison of a Metaheuristic with an Exact Method to solve the Vehicle Routing Problem: Application to a Case Study in the Distribution Industry

André Guerreiro
Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisboa, Portugal

Abstract
In this paper, it is evaluated a vehicle routing problem model applied to a real case of a distribution company. A combinatorial optimization method was implemented based on a rule-guided heuristic – a tabu search based metaheuristic – in order to obtain practical results and to compare them with the real plan used daily by the company. The problem’s extension is referred in the literature as Heterogeneous Fleet Vehicle Routing Problem with Time Windows and Split Deliveries (HFVRPTWSD). The objective is to minimize the total travelled distance, the necessary number of vehicles and the sum of travel time through the establishment of a set of routes between all clients and a central depot for the given time horizon. The main purpose is to satisfy all clients’ requests while complying with a visit schedule. A number of constraints define the problem working environment, with special attention to time windows constraints, truck capacity and accessibility constraints. A relaxation of the original problem – split delivery – is allowed whenever it improves the solution. The output provided by an implementation of a VRP-solving algorithm will be compared with the results obtained from an exact method applied to a smaller part of the problem. After comparison with the global optimum, we may capture the algorithm’s power and therefore apply it to the overall problem. The case study analyzed is based on a Portuguese company that distributes groceries from distribution centers to urban retailers in Portugal.

1. Introduction
In this paper, it is presented a vehicle routing problem (VRP) applied to a real Case Study in the distribution area. The analyzed problem presents several specifications that add or relax constraints to the original VRP. Therefore, a most suitable variation of the VRP had to be considered, including Time Windows constraints (VRPTW), Capacity Constraints with Heterogeneous Fleet (HFVRP) and Split Deliveries (VRPSD) allowance whenever it improves a solution. The defined problem may be described as HFVRPTWSD – Heterogeneous Fleet Vehicle Routing Problem with Time Windows and Split Deliveries (Belfiore, 2006).

Furthermore, some additional constraints have to be attended in order to comply with the Case Study problem’s restrictions, such as tour duration and clients’ accessibility constraints.

The group Jerónimo Martins (JM) runs five Distribution Centers (DC) in Portugal, three in the Northern region and two in the South, to serve roughly four hundred commercial surfaces spread all over the country, under the brands of Pingo Doce, Feira Nova and Recheio. The Case Study described on this paper focus the planning of the South Region, more specifically the planning of the DC located in Azambuja.

The problem solved in this paper was previously approached by Figueiredo (2007) using an Exact Method. In his work, Figueiredo (2007) achieved significant improvements when compared with the real planning. However, due to limitations imposed by the exact nature of his model, it was not possible to solve the problem in its full extent in a reasonable amount of time.

The present work was motivated by the need of obtaining satisfactory results in a short amount of time. Consequently, it was developed an approximate method to address those requirements: a metaheuristic inspired on Tabu Search and several diversification procedures.

The main goal of this thesis was to develop a model that could obtain high-quality solutions – fully compliant with all problem constraints – in a reasonable amount of time in order to proceed with a successful implementation in the daily operations of the JM company.

A broader objective intends to verify that an approximate method gets advantage in performance over an exact method.

2. Problem Definition and Case Study Analysis
In this paper, it is considered a HFVRPTWSD problem (Heterogeneous Fleet Vehicle Routing Problem with Time Windows and Split Deliveries) consisting of four instances (4 days of planning) with dimension of 40 clients, each with an associated demand and a specific time window to fulfill. Also, there are accessibility constraints that need to be respected, which require a heterogeneous fleet to serve clients with different accessibility conditions, particularly due to urban placement strategy. Another aspect included in the accessibility constraint consists on the unloading strategy. Either the client`s facility includes a docking station or the truck is equipped with an elevator to unload the cargo during each delivery. Therefore, this constraint was defined as the largest truck (in volume capacity) that can supply each client.

The problem’s time horizon is of one day and client’s time windows are divided in morning planning and afternoon planning. The distance and time matrices were collected
from real measurements and the resultant average speed is of 40 km/h. Service times are composed by a fixed component of 20 minutes per visit plus a variable component of 2 minutes per unloaded palette (unit of volume amount).

Due to computational effort limitations, the described problem, originally formulated by Figueiredo (2007), represents a short example of the real planning. The real extent of JM’s group distribution operation in Portugal consists of roughly 400 clients and 5 distribution centers. Also, it does not include some restrictions of the real planning, such as a multi-depot with multi-product scheduling. On the other hand, the incurred simplification excludes mixed cargo routes (with more than one product category) that would certainly improve the quality of the obtained solutions. Some other operations were not contemplated in this formulation, such as cross-docking, backhauling and transshipment operations.

Therefore, the formulated problem describes the planning for a single warehouse of the Azambuja Distribution Center: 5401 – Non Perishable product category (NP). The planning associated with this product category is characterized by its large variability, medium frequency and flexible time windows, resulting in variable routes and therefore subject to optimization.

The remaining product categories (perishable) have a short life time. Hence, the associated demand has a low variability, high frequency and narrow time windows, which results essentially in fixed routes, with little interest for optimization purposes.

Finally, the resolution of the problem allows split deliveries whenever it improves the quality of the solution during the search process. Due to a service time increase per client visit (Figueiredo, 2007), in this work the maximum number of visits per client was limited to two when Split Deliveries are allowed.

The allowance of Split Deliveries on a VRP, first introduced by Dror e Trudeau (1989), brings greater advantages under certain conditions.

According to Archetti et al. (2006), the utilization of Split Deliveries is justified when the following criteria are met: $\frac{\text{average demand}}{\text{vehicle capacity}}$ and demand variability is low.

Graph 1 shows that the use of Split Deliveries in the Case Study problem allows reduction on the number of routes. This comparison is also valid for vehicles with 24-palette and 31-palette capacity but in a lesser extent since most clients can’t be served by 31-palette trucks due to accessibility constraints and the average demand of clients that can be served by 24-palette trucks is generally less than half of its capacity.

3. State-of-the-art

In the past two decades, several hundreds of published articles addressed applications of the Tabu Search (TS) in combinatorial optimization problems. The initial concept was introduced by Glover (1986), described as a metaheuristic inspired on a lower-level heuristic designed to address specific problems. A metaheuristic can be defined as an iterative method that guides the search process of the subordinated heuristic in order to efficiently produce high quality solutions (Voß, 1999).

Like other Local Search (LS) procedures, TS explores the search space by moving from a current solution $S$ to the best found solution in its neighborhood $N(S)$. It can be referred as an extension of the classical Local Search methods, combining LS with short term memory of some solution attributes.

3.1 Metaheuristics vs Exact Methods

Metaheuristics such as Tabu Search had evolved through the time heading towards a greater performance. It allowed the resolution of bigger and more complex problems and narrowed the gap to Exact Methods. According to Breton (1998), several aspects reinforce the utilization of metaheuristics:

- The Exact Methods are prohibitive for large problems due to computational effort requirements;
The approximate methods are simple, easy to implement and to use and require less resources, which is particularly useful for a periodic problem like the daily operation of a real company;
- The available data is not exact or there is limited information, making necessary the use of approximations that may override the error caused by non-optimality;
- A traditional heuristic is currently considered inefficient due to its limitations to overcome local optima;
- The best developed algorithms simultaneously combine hybrid strategies that include parallel processing, learning mechanisms (ex: Ant Algorithms), local search or a population of solutions (ex: Genetic Algorithms). Such examples may be found in Taillard (1993) with the conception of the Tabu Search Parallel; Gambardella et al. (1999), who implemented the Multiple Ant Colony System (MACS); Mester and Bráysy (2004): Active Guided Evolution Strategy (AGES); and Rochat-Taillard (1995), with the introduction of the Adaptive Memory Procedure concept.

The superior performances obtained by these algorithms make them suitable for benchmark purposes. Therefore, the model developed in this work was compared in two sets of well known benchmark instances in order to validate its efficiency and to ensure further applicability. The obtained results are discussed in section 5.

- Christofides, Mingozzi e Toth (1979): 14 CVRP problems with dimension of n clients (50 ≤ n ≤ 199). The problems 6-10, 13 and 14 also include tour length constraints;
- Solomon (1987): 56 CVRP TW problems with dimension of 100 clients, distributed by 6 different categories: The geographical data are randomly generated in problem sets R1 and R2, clustered in problem sets C1 and C2, and a mix of random and clustered structures in problem sets by RC1 and RC2. Problem sets R1, C1 and RC1 have a short scheduling horizon and allow only a few customers per route (approximately 5 to 10). In contrast, the sets R2, C2 and RC2 have a long scheduling horizon permitting many customers (more than 30) to be serviced by the same vehicle.

### 3.2 HFVRPTWSD

The Heterogeneous Fleet Vehicle Routing Problem with Time Windows and Split Deliveries has been approached previously in several applications of combinatorial optimization problems. For this matter, it is shortly described the work of Belfiore (2006). In her work, it has been implemented a Scatter Search metaheuristic model to solve a HFVRPTWSD Case Study of a Brazilian Retail Industry. The problem consisted of serving 519 clients, but the dimension of the problem could vary between 50 ≤ n ≤ 400, depending on the demand requests.

The comparison with the real planning pointed out several important conclusions about the applicability of an approximate model in large scale real problems: the solution obtained represented a cost cut of € 974.016\(^{1}\) annually (the obtained results were projected for an average representative week), which corresponds a relative saving of 6.3%. The processing time of the algorithm was inferior to 3500 seconds for the busiest day (with higher number of clients), returning results in less than 20 minutes for problems sized up to n = 150.

### 4. Model Implementation and Model Validation

The model here described intends to solve the VRP as well as its most known variants, such as the Capacitated Vehicle Routing Problem (CVRP), the Heterogeneous Fleet Vehicle Routing Problem (HFVRP); Vehicle Routing Problem with Time Windows (VRPTW), where time window constraints are rigid or flexible; and the Vehicle Routing Problem with Split Deliveries (VRPSD), considered a relaxation of the original problem.

It also includes additional constraints like route length, clients’ accessibility constraints and multi-product delivery. The implemented model is inspired in a Tabu Search metaheuristic, previously approached in section 3.1. Therefore, the search process does not allow inverse non-improving movements after the previous attempted k movements – considered tabu.

Considering the initial goal of the model – to improve a previous planning – it is necessary an initial solution to proceed with its improvement. Additionally, two constructive heuristics were developed as an alternative to a pre-arranged initial solution import.

#### 4.1 Constructive Heuristics

The quality of the initial solution often affects a heuristic performance. The main purpose of the implemented construction heuristics is to feed the main algorithm with an initial solution, but it also meets performance comparison purposes.

The first of the construction heuristics was inspired in Clark&Wright (1964) Savings concept. It was not possible to cope with all of the Case Study’s constraints though, once the complexity of all requirements could prevent the algorithm from getting a complete solution. Thus, the implementation only complies with capacity constraints and maximum route length constraints (or traditional CVRP-compliant).

This heuristic couldn’t ensure full feasibility for the Case Study problem but produced almost feasible solutions. The initial solutions implemented were obtained through a couple of manual changes between clients or routes that

\(^{1}\) Conversion based on Brazil Central Bank website [http://www4.bcb.gov.br](http://www4.bcb.gov.br) for an exchange rate of 1€=2,66936R$ at March, 30 of 2006 (Belfiore thesis revision date)
did not meet all of the problem’s constraints. Below it’s presented a possible pseudo-code of the implemented algorithm:

**Algorithm 1: Savings Constructive Heuristic**

Savings matrix generation;
While there are uninserted clients Or there are unattempted Savings entries:
   - It’s chosen the biggest saving between two clients \( a \) and \( b \) And at least one of the clients wasn’t inserted in the solution;
   - If client \( a \) was already inserted:
     - If all constraints are met (capacity and route length constraints):
       - Client \( b \) is inserted immediately after or before client \( a \);
     - If client \( b \) was already inserted:
       - If all constraints are met (capacity and route length constraints):
         - Client \( a \) is inserted immediately after or before client \( b \);
     - If neither client \( a \) nor \( b \) was previously inserted:
       - If there is (at least) one available empty route And all constraints are met (capacity and route length constraints):
         - Both clients are inserted in the available empty route with larger capacity;
       - Else
         - It’s inserted client \( a \);

**End of cycle**

If “Number of inserted clients” counter is lower than the total number of clients (problem’s size):
While there are uninserted clients:
   - For each uninserted client \( j \):
     - If there’s a route \( i \) that satisfies: (vehicle \( i \) capacity - sum of visited clients’ demand) \( \leq \) client \( j \) demand And route \( i \) length \(<\) max route length - 1:
       - Client \( j \) is inserted in route \( i \);

**End of procedure.**

*The insertion of a client \( a \) (or \( b \)) is made after or before client \( b \) (or \( a \)) according to client \( b \) (or \( a \)) insertion in the beginning or in the end of the route (that is, the construction of a route follows a middle-to-extremity rule)*

A more complex algorithm was implemented to produce initial solutions, specifically to address Solomon’s instances that require time windows constraint feasibility. This heuristic follows a criterion of distance/time/both minimization between an inserted client and a list of candidates that comply with all constraints. The criterion is user-defined according to the specific problem addressed and applies to the choice of the next visited client. The route construction begins with the choice of the client that may be visited sooner, as a measure to start the routes as soon as possible. Below it is presented the algorithm’s pseudo-code:

**Algorithm 2: Local Construction Heuristic**

For each available vehicle:

If the number of inserted clients reaches problem’s size:
   - End cycle;
Else The first inserted client is the one with the lower time window;
For the remaining route positions:
   While the list of candidates isn’t void:
     - Get a list of candidates with all clients that may be inserted in the next route position (that is, compliant with problem’s constraints);
     - It is selected the one that minimizes the choice criterion with the previous visited client;
   If candidate list is void:
     - Move on to the next vehicle’s route construction;

**End of cycle**;

If “Number of inserted clients” counter is lower than the total number of clients (problem’s size):
While there are uninserted clients:
   - For each uninserted client \( j \):
     - If there’s a route \( i \) that satisfies: (vehicle \( i \) capacity - sum of visited clients’ demand) \( \leq \) client \( j \) demand And Time Windows constraints are respected:
       - Client \( j \) is inserted in route \( i \);

**End of procedure.**

**4.2 Model Implementation**

After obtaining a feasible initial solution, the model proceeds to the solution optimization towards a predefined objective. The search through the range of possible solutions – search space – consists of a set of elementary operations that modify the current solution in every iteration, generating a list of candidates (neighborhood list). The operations may be of:

1. Random swap of route of a randomly selected client (default strategy);
2. Random switch of one or more clients between two randomly selected routes;
3. Deterministic swap between two clients of the same route;
4. Random swap of route of a client selected by a deterministic way;

Regardless of the neighborhood search strategy, all generated candidates (obtained from a certain solution called current solution) are submitted to a feasibility check to ensure full constraint compliance. If any constraint is violated, the solution is considered infeasible and a new one is generated.

The neighborhood search process is conditioned by the more or less restrictive characteristics of the problem, once the selected LISTSIZE candidates in each iteration must comply with all problem’s constraints. Since the neighborhood does not contain enough feasible different solutions for all situations, it was conceded a compromise to accept duplicated solutions within the generated candidate list.

The next step is to choose the fittest candidate that it is not considered tabu: the solution in the neighborhood list
with the inferior objective function value. Alternatively, it may be chosen the solution with the smaller number of routes or the smaller number of visited clients in a single route. In such case, the FORCE_MIN_TRUCKS criterion applies, as reviewed ahead.

The default strategy (movement 1) consists of randomly choosing a client, a random insertion route and a position in which that client is inserted, and transfer the client from the initial route into the position of the new designated route. It is also referred in the literature as a (1,0) k-interchange.

The randomness is assured by a random number generator and its output manipulation through rest and sum operators. Each time a run is executed, the random seeds are initialized with the run time, which makes results reproducibility impossible.

The default strategy is usually powerful enough to solve problems with modest dimension like the Case Study problem approached in the next section.

However, it was necessary to implement more powerful search strategies to deal with complex problems such as Christofides and Solomon instances, also reviewed in section 5. Therefore, movements (2) to (4) are allowed by enabling a DIVERSIFY parameter. Movements (2) and (3) are executed in every two search restarts for routes with more than two visited clients. In the odd search attempts the default strategy is followed.

Movement (2) consists of a switch of route between one to three clients, depending on a random generated variable that gives the number of clients switch to attempt, on the number of clients visited by each route and on the feasibility of the resulting routes.

Movement (3) is the unitary move of a 2-opt local search procedure, attempting every possible permutation within a certain route until an improvement is achieved. It is not an exhaustive algorithm because its execution stops when any improvement is found or if the route is already optimized. This algorithm is executed always after the occurrence of a movement (2) in both affected routes, in order to rearrange them.

Finally, movement (4) is similar to the default search strategy (movement 1), with the difference that the chosen client is included in the shortest route. This movement is conditioned by the parameter FORCE_MIN_TRUCKS, which forces the search to find solutions with fewer routes by systematically choosing the shortest route, that is, with fewer clients to visit.

The FORCE_MIN_TRUCKS strategy often drives the search to solutions that perform worse in terms of fitness. However, the following rearrangement of that solution may lead to unexplored better solutions. This strategy occurs if certain conditions are met in a specific period of the search process: if no better solutions were found in the latest $t$ seconds, it enforces the search of solutions with fewer routes for $2t$ time, being $t = \frac{1}{4}*STOP*MAX\_TIME$ and $STOP$ and $MAX\_TIME$ are termination criterions that will end the run after $MAX\_TIME$ seconds of execution or after $STOP*MAX\_TIME$ seconds of execution without improvement in the better solution found.

This strategy allows a deterioration of the Objective Function (up to 5% worse) if a new solution with a smaller number of routes is found. For that reason, it is considered an auxiliary objective function instead of a diversification procedure.

Diversification procedures were implemented as a set of rules that give the search process dynamic characteristics: The often used restart-search procedure occurs in every $it$ iterations, being $i$ a variable that takes values between $[RESTART\_IT; UPPER\_IT]$ with a growth given by $2^n$. The two parameters $RESTART\_IT$ and $UPPER\_IT$ are the boundaries that limit the extent of the search. The bigger the value taken by $i$, the more non-improving iterations are accepted. If that iteration limit is reached, search restarts from the best solution found so far.

The efficiency of the search is also affected by the parameters $LIST\_SIZE$ and $TABU\_SIZE$, that stand for the dimension of the candidate list and the amount of tabu movements to retain in the short term memory, respectively.

The implementation of Split Deliveries was decided upon the possible advantages previously shown in section 2, attending the Case Study’s characteristics. For many routes, the number of visited clients did not exceed one or two despite the truck’s capacity wasn’t fully used. Figure 1 illustrates an example of a solution improvement with Split Deliveries allowance as it is implemented.

![Diagram](image)

**Figure 1: Search process movements with Split Deliveries**
If during the search process (movement 1 or 4) it is attempted to move client B from route 2 to route 3 but the resulting solution is found unfeasible due to capacity constraints, client B demand may be split in two if truck 3 is able to transport at least % of client’s request, being s = SPLIT_CRITERION usually defined as 50%. In such case, route 2 truckload is decreased by the amount of (Client B demand - truck 3 capacity - route 3 truckload) and route 3 truckload = truck 3 capacity.

After the occurrence of a split there is no immediate improvement of the found solution: solution (1b) is clearly worse than (1a) by d(B,C) - d(CD;C), for the same number of trucks. However, if parameter configuration is set properly, new movements will origin a better solution. An example of that is shown in figure (1c).

### 4.3 Model Validation

In order to get a notion of the algorithm’s power, it was run with two set of known benchmark instances in the literature: Christofides et al. (1979) and Solomon (1987). Below there’s a short presentation of the obtained results (Table 2 and 3).

The obtained results were considered very satisfactory, because it makes possible a future implementation on JM group’s operations, which have a bigger dimension than the analyzed Case Study Problem.

<table>
<thead>
<tr>
<th>Authors – Algorithms</th>
<th>% Average over the BKS for the 14 problems</th>
<th>Average processing time (minutes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taillard (1993) - TS Parallel</td>
<td>0,02%</td>
<td>21,30</td>
</tr>
<tr>
<td>Gendreau et al. (1994) - Taburoute</td>
<td>0,82%</td>
<td>55,59</td>
</tr>
<tr>
<td>Rochat-Taillard (1995) - Adaptive Memory TS</td>
<td>0,00%</td>
<td>16,19</td>
</tr>
<tr>
<td>Toth-Vigo (1998) - Granular TS</td>
<td>0,64%</td>
<td>3,84</td>
</tr>
<tr>
<td>Mester e Bräysy (2004) - AGES best</td>
<td>0,03%</td>
<td>7,72</td>
</tr>
<tr>
<td>- AGES fast</td>
<td>0,07%</td>
<td>0,27</td>
</tr>
<tr>
<td>Guerreiro - Tabu Search</td>
<td>3,97%</td>
<td>4,51</td>
</tr>
</tbody>
</table>

**Table 2 - Performance comparison with some of the most efficient algorithms for the Christofides et al. (1979) benchmark instances - Adapted from Laporte (2007)**

It is worth mentioning that it was possible to reach the Best Known Score (BKS) ever obtained by a heuristic in several instances, as it is the case of instances C101, C108 and C202 in Solomon (1987); in several others, it was possible to overcome the BKS, as it is the example of instances 7, 9, 13 and 14 of Christofides et al. (1979) and instances R201, R205, R210, RC201, RC202 e RC207 of Solomon (1987).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>R1 (12 Inst. Average)</td>
<td>Nº of vehicles</td>
<td>12,38</td>
<td>12,58</td>
<td>12,33</td>
<td>12,67</td>
<td>12,39</td>
<td>14,25</td>
<td>13,3%</td>
</tr>
<tr>
<td>Proc. Distance</td>
<td>Total</td>
<td>1210,83</td>
<td>1197,42</td>
<td>1201,79</td>
<td>1200,33</td>
<td>1230,48</td>
<td>1273,24</td>
<td>6,3%</td>
</tr>
<tr>
<td>Proc. Time(s)</td>
<td>1800</td>
<td>2700</td>
<td>3600</td>
<td>2900</td>
<td>6887</td>
<td>348</td>
<td>-87,1%</td>
<td></td>
</tr>
<tr>
<td>C1 (9 Inst. Average)</td>
<td>Nº of vehicles</td>
<td>10,00</td>
<td>10</td>
<td>10,00</td>
<td>10,00</td>
<td>10,00</td>
<td>10,78</td>
<td>7,8%</td>
</tr>
<tr>
<td>Proc. Distance</td>
<td>Total</td>
<td>828,38</td>
<td>828,38</td>
<td>830,75</td>
<td>858,59</td>
<td>883,47</td>
<td>358</td>
<td>-88,8%</td>
</tr>
<tr>
<td>Proc. Time(s)</td>
<td>300</td>
<td>3200</td>
<td>2900</td>
<td>7315</td>
<td>358</td>
<td>-88,8%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RC1 (8 Inst. Average)</td>
<td>Nº of vehicles</td>
<td>11,92</td>
<td>12,33</td>
<td>11,95</td>
<td>12,12</td>
<td>12,00</td>
<td>14,63</td>
<td>18,6%</td>
</tr>
<tr>
<td>Proc. Distance</td>
<td>Total</td>
<td>1388,13</td>
<td>1377,39</td>
<td>1364,17</td>
<td>1388,15</td>
<td>1387,01</td>
<td>1466,55</td>
<td>6,5%</td>
</tr>
<tr>
<td>Proc. Time(s)</td>
<td>1800</td>
<td>2600</td>
<td>3600</td>
<td>2900</td>
<td>5632</td>
<td>470</td>
<td>-81,9%</td>
<td></td>
</tr>
<tr>
<td>R2 (11 Inst. Average)</td>
<td>Nº of vehicles</td>
<td>3,00</td>
<td>3,09</td>
<td>3,00</td>
<td>3,00</td>
<td>5,82</td>
<td>88,3%</td>
<td></td>
</tr>
<tr>
<td>Proc. Distance</td>
<td>Total</td>
<td>960,31</td>
<td>961,72</td>
<td>966,56</td>
<td>1046,56</td>
<td>983,15</td>
<td>2,2%</td>
<td></td>
</tr>
<tr>
<td>Proc. Time(s)</td>
<td>1800</td>
<td>9800</td>
<td>2900</td>
<td>3372</td>
<td>190</td>
<td>-98,1%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2 (8 Inst. Average)</td>
<td>Nº of vehicles</td>
<td>3,00</td>
<td>3</td>
<td>3,00</td>
<td>3,00</td>
<td>3,50</td>
<td>16,7%</td>
<td></td>
</tr>
<tr>
<td>Proc. Distance</td>
<td>Total</td>
<td>591,85</td>
<td>589,86</td>
<td>592,29</td>
<td>591,14</td>
<td>643,96</td>
<td>9,2%</td>
<td></td>
</tr>
<tr>
<td>Proc. Time(s)</td>
<td>1800</td>
<td>3600</td>
<td>2900</td>
<td>8187</td>
<td>371</td>
<td>-89,7%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RC2 (8 Inst. Average)</td>
<td>Nº of vehicles</td>
<td>3,33</td>
<td>3,62</td>
<td>3,38</td>
<td>3,38</td>
<td>6,00</td>
<td>65,7%</td>
<td></td>
</tr>
<tr>
<td>Proc. Distance</td>
<td>Total</td>
<td>1149,28</td>
<td>1119,59</td>
<td>1133,42</td>
<td>1220,28</td>
<td>1141,78</td>
<td>2,0%</td>
<td></td>
</tr>
<tr>
<td>Proc. Time(s)</td>
<td>1800</td>
<td>7800</td>
<td>2900</td>
<td>5798</td>
<td>197</td>
<td>-97,5%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total (56 Inst.)</td>
<td>Nº of vehicles</td>
<td>418</td>
<td>427</td>
<td>423</td>
<td>419</td>
<td>525</td>
<td>23,1%</td>
<td></td>
</tr>
<tr>
<td>Proc. Distance</td>
<td>Total</td>
<td>57582,87</td>
<td>57098,10</td>
<td>57423,75</td>
<td>59929,67</td>
<td>60063,08</td>
<td>5,2%</td>
<td></td>
</tr>
<tr>
<td>Proc. Time(s)</td>
<td>87300</td>
<td>281000</td>
<td>162400</td>
<td>342507</td>
<td>17794</td>
<td>-93,7%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 3 - Performance comparison with some of the most efficient algorithms for the Solomon (1987) benchmark instances – Adapted from Gambardella et al. (1999)**

2 Reaches Best Known Score (BKS) in 12 of 14 instances

3 Matches Taillard (1993) in 12 instances and overcomes BKS in 2 of 14 Christofides et al. (1979) instances
For the 41 instances where the global optima is known (Solomon, 2005) it was obtained on average results 9% above (worse than) the global optima and 4.5% above Rochat-Taillard (1995) best results.

For that matter, it is considered that the results obtained are good enough for the purpose of the model.

Regardless, the quality of the solutions found in Benchmark Instances was affected by the high variability of the results, partially explained by the random nature of the search process.

5. Results

The problem using the Case Study data (please refer to section 2) was originally formulated and solved by Figueiredo (2007), using an Exact method. In his work, a computational complexity analysis showed that the implemented Model couldn’t return results in a reasonable amount of time (maximum of 7200 seconds) for problems with dimension n > 10 due to the exponential growth of the additional variables and constraints.

It is necessary to point out that Figueiredo (2007) used several techniques in order to obtain practical results, such as subdividing the problem in clusters. Still, he didn’t reach optimality in most of the executed runs due to run time restrictions. The average integrality gap of the executed runs was of 4.5%. The formulated problem itself was severely conditioned by the necessary processing time, resulting in the simplification of the real planning. The subdivision of the problem added constraints that didn’t exist in the real problem, therefore affecting the quality of the solutions.

Nevertheless, Figueiredo (2007) obtained promising results with a reduction of 20% of the total distance and the necessary number of routes for the VRPSD scenario. The total route duration, another O.F. component, suffered a decrease of 4 to 7% for the simple VRPSD scenario. The difference between the two scenarios lies on the allowed number of visits per client: in both presented scenarios, it was allowed k-Split Deliveries, with k=2 for scenario ‘Simple VRPSD’ and k=Q-1 for scenario ‘VRPSD’, Q is client’s demand. (It’s not allowed to split unitary requests).

The implemented model was run in four different scenarios and compared with 4 days of the Case Study’s real planning. Split Deliveries were not allowed in scenarios C1 and C2, while scenarios C3 and C4 were run with allowance of a maximum of two visits per client. The scenarios C1 and C3 were run with Objective Function (OF1) while scenarios C2 and C4 were run with OF2.

\[
\begin{align*}
\text{OF1:} & \quad \sum_{i=1}^{m} \sum_{j=0}^{n} \sum_{k=1}^{n} x_{ijk} d_{ij} + \sum_{i=1}^{m} \sum_{j=0}^{n} x_{ijk} v_{ij} + \sum_{i=1}^{m} \sum_{j=0}^{n} x_{ijk} t_{ij} + c_{i} \sum_{i=1}^{m} \sum_{j=0}^{n} x_{ijk} c_{i} \\
\text{OF2:} & \quad w_{d} \sum_{i=1}^{m} \sum_{j=0}^{n} x_{ijk} d_{ij} + w_{v} \sum_{i=1}^{m} \sum_{j=0}^{n} x_{ijk} v_{ij} + w_{t} \sum_{i=1}^{m} \sum_{j=0}^{n} x_{ijk} t_{ij} + w_{c} \sum_{i=1}^{m} \sum_{j=0}^{n} x_{ijk} c_{i}
\end{align*}
\]

with \( w_{d} + w_{v} + w_{t} + w_{c} = 1 \).

The obtained results were also compared with Figueiredo (2007) main results, for performance comparison purposes.

OF1 is a multi-objective function, proposed by Figueiredo (2007) for this Case Study, and its use lies on comparison purposes. It consists of a three-component minimization: total distance, used number of vehicles and client arrival time. Since the terms of OF1 are not properly normalized, the relative weight of the terms depends on the values taken by the respective variables.

\[
\begin{align*}
\text{OF1:} & \quad \sum_{i=1}^{m} \sum_{j=0}^{n} \sum_{k=1}^{n} x_{ijk} d_{ij} + \sum_{i=1}^{m} \sum_{j=0}^{n} x_{ijk} v_{ij} + \sum_{i=1}^{m} \sum_{j=0}^{n} x_{ijk} t_{ij} + c_{i} \sum_{i=1}^{m} \sum_{j=0}^{n} x_{ijk} c_{i}
\end{align*}
\]

The Case Study results show that this O.F. assumes relative weights of 99% for distance; less than 1% for number of vehicles and almost 0% for the client arrival time term.

Since the solutions obtained with this O.F. were severely conditioned by its distance minimization focus, as shown ahead, it was implemented a more flexible O.F (OF2) that consists of four terms: distance, number of vehicles, total route duration and sum of vehicle’s capacities, with associated weights \( w_{d}, w_{v}, w_{t}, w_{c} \) respectively. The later only makes sense in a Heterogeneous Fleet VRP, with trucks with different associated costs, or this term’s minimization would lead to the same results given by the number of vehicles’ term.

OF2 minimization leads to a progressive improvement of the initial solution through a permanent comparison of its four components: initial distance (\( d_{ij} \)), initial number of vehicles (\( v_{ij} \)), initial total route duration (\( t_{ij} \)) and initial sum of vehicle’s capacities (\( c_{i} \)). The choice of the initial solution while using this O.F. must be careful, though, since a better potential improvement of one of its components may lead the search process to solutions that minimize mostly that component, regardless of the other components. There lies the explanation for component weights attribution. In the following runs (scenarios C2 and C4), the utilized weights were: \( w_{d} = 50\% \), \( w_{v} = 10\% \), \( w_{t} = 30\% \), \( w_{c} = 10\% \).

The obtained results were divided into three components for comparison purposes: distance, necessary number of vehicles and total route duration. Additionally, it is presented the discriminated times that make the total route duration for one of the planned days.

For the distance component the results obtained are shown in table 4. The comparison with the real planning for the sum of the analyzed days shows that it were reached savings up to 24% in total travelled distance (scenario C3). Moreover, even without Split Deliveries allowance (scenarios C1 and C2), the obtained results are competitive with Figueiredo’s (2007) best results.
As pointed out in the beginning of this section, it was possible to overcome Figueiredo's (2007) best results using a metaheuristic due to problem simplifications and run time restrictions to in the Exact Model. Therefore, for the component “Number of vehicles” it is also presented the best results in table 5.

The reduction of required vehicles in this particular problem was quite relevant because the average distance between the distribution center and any client was by far superior to the average distance between any two clients. Therefore, any route suppression made possible substantial savings.

The combination of the previous results allows calculating the average length per route. As shown in table 5, the slight variation shows that it is possible to make a better planning with fewer routes with similar route length.

Finally, the route duration component presented the biggest challenge. Results are shown in table 6.

The runs with OF1 (scenarios C1 and C3) were the most penalized due to its nearly zero weighting of time component.

If we consider the average route duration, all of the considered scenarios present worse solutions. The increase of the routes’ duration may affect the daily operation since most of the vehicles are used in multiple routes during one day.

To better understand what caused such bad performance in this component, despite the rest of the components improvement, table 7 shows a discrimination of the operation times for day of planning 12.

In table 7, the sum of routes’ duration is presented in its time components: Variable Service Time (VST), Fixed Service Time (FST), travel time and total delays. The variable service time doesn’t change within the different scenarios because it only depends on the delivered quantities, which are the same for the same problem regardless of the obtained solution.

<table>
<thead>
<tr>
<th>Distance (km)</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>Total</th>
<th>% JM Planning</th>
<th>% JM Planning</th>
</tr>
</thead>
<tbody>
<tr>
<td>JM Planning</td>
<td>3391</td>
<td>2984</td>
<td>3085</td>
<td>3014</td>
<td>12474</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Simple VRPSD</td>
<td>3163</td>
<td>2481</td>
<td>2429</td>
<td>2577</td>
<td>10650</td>
<td>-14,6%</td>
<td>-14,6%</td>
</tr>
<tr>
<td>Figueiredo,</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VRPSD</td>
<td>2937</td>
<td>2366</td>
<td>2309</td>
<td>2448</td>
<td>10060</td>
<td>-19,4%</td>
<td>-19,4%</td>
</tr>
<tr>
<td>Initial</td>
<td>3217</td>
<td>2739</td>
<td>2457</td>
<td>2924</td>
<td>11337</td>
<td>-9,1%</td>
<td>-9,1%</td>
</tr>
<tr>
<td>Solution</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1 TS Guerreiro OF1</td>
<td>2922</td>
<td>2376</td>
<td>2248</td>
<td>2487</td>
<td>10033</td>
<td>-19,6%</td>
<td>-19,6%</td>
</tr>
<tr>
<td>C2 TS Guerreiro OF2</td>
<td>3070</td>
<td>2386</td>
<td>2320</td>
<td>2479</td>
<td>10255</td>
<td>-17,8%</td>
<td>-17,8%</td>
</tr>
<tr>
<td>C3 TS Guerreiro SD OF1</td>
<td>2740</td>
<td>2251</td>
<td>2198</td>
<td>2248</td>
<td>9437</td>
<td>-24,3%</td>
<td>-24,3%</td>
</tr>
<tr>
<td>C4 TS Guerreiro SD OF2</td>
<td>2808</td>
<td>2384</td>
<td>2237</td>
<td>2257</td>
<td>9686</td>
<td>-22,4%</td>
<td>-22,4%</td>
</tr>
</tbody>
</table>

Table 4 - Total travelled distance by scenario and planned day

<table>
<thead>
<tr>
<th>Number of required vehicles</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>Total</th>
<th>% JM Planning</th>
<th>Average route length (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JM Planning</td>
<td>29</td>
<td>25</td>
<td>26</td>
<td>26</td>
<td>106</td>
<td>-</td>
<td>117,68</td>
</tr>
<tr>
<td>Simple VRPSD (Figueiredo,</td>
<td>27</td>
<td>21</td>
<td>21</td>
<td>22</td>
<td>91</td>
<td>-14,2%</td>
<td>117,03</td>
</tr>
<tr>
<td>VRPSD (Figueiredo, 2007)</td>
<td>25</td>
<td>20</td>
<td>20</td>
<td>21</td>
<td>86</td>
<td>-18,9%</td>
<td>116,98</td>
</tr>
<tr>
<td>Initial Solution</td>
<td>28</td>
<td>23</td>
<td>21</td>
<td>25</td>
<td>97</td>
<td>-8,5%</td>
<td>116,88</td>
</tr>
<tr>
<td>(C1) TS Guerreiro OF1</td>
<td>25</td>
<td>20</td>
<td>19</td>
<td>21</td>
<td>85</td>
<td>-19,8%</td>
<td>118,04</td>
</tr>
<tr>
<td>(C2) TS Guerreiro OF2</td>
<td>26</td>
<td>20</td>
<td>20</td>
<td>21</td>
<td>87</td>
<td>-17,9%</td>
<td>117,87</td>
</tr>
<tr>
<td>(C3) TS Guerreiro SD OF1</td>
<td>23</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>80</td>
<td>-24,5%</td>
<td>117,96</td>
</tr>
<tr>
<td>(C4) TS Guerreiro SD OF2</td>
<td>23</td>
<td>20</td>
<td>19</td>
<td>19</td>
<td>81</td>
<td>-23,6%</td>
<td>119,58</td>
</tr>
</tbody>
</table>

Table 5 – Number of required vehicles by scenario and planned day

<table>
<thead>
<tr>
<th>Total Route Duration (min)</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>Total</th>
<th>% JM Planning</th>
<th>Average route duration (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>JM Planning</td>
<td>4920</td>
<td>4454</td>
<td>4100</td>
<td>4436</td>
<td>17910</td>
<td>-</td>
<td>168,96</td>
</tr>
<tr>
<td>Simple VRPSD (Figueiredo,</td>
<td>4716</td>
<td>3976</td>
<td>3863</td>
<td>4008</td>
<td>16563</td>
<td>-7,5%</td>
<td>182,01</td>
</tr>
<tr>
<td>VRPSD (Figueiredo, 2007)</td>
<td>4986</td>
<td>4190</td>
<td>4029</td>
<td>4012</td>
<td>17217</td>
<td>-3,9%</td>
<td>200,20</td>
</tr>
<tr>
<td>Initial Solution</td>
<td>5178</td>
<td>5832</td>
<td>5334</td>
<td>5658</td>
<td>22002</td>
<td>22,8%</td>
<td>226,82</td>
</tr>
<tr>
<td>(C1) TS Guerreiro OF1</td>
<td>5748</td>
<td>5682</td>
<td>4842</td>
<td>4794</td>
<td>21066</td>
<td>17,6%</td>
<td>247,84</td>
</tr>
<tr>
<td>(C2) TS Guerreiro OF2</td>
<td>4578</td>
<td>3870</td>
<td>3720</td>
<td>3924</td>
<td>16092</td>
<td>-10,2%</td>
<td>184,97</td>
</tr>
<tr>
<td>(C3) TS Guerreiro SD OF1</td>
<td>6916</td>
<td>6407</td>
<td>6213</td>
<td>6546</td>
<td>26082</td>
<td>45,6%</td>
<td>326,03</td>
</tr>
<tr>
<td>(C4) TS Guerreiro SD OF2</td>
<td>4801</td>
<td>3867</td>
<td>4318</td>
<td>3791</td>
<td>16777</td>
<td>-6,3%</td>
<td>207,12</td>
</tr>
</tbody>
</table>

Table 6 – Total route duration (in minutes) by scenario and planned day
The fixed service time varies with the amount of visits, which are 39 for the scenarios without Split Deliveries allowance. Thus, the scenarios with Split Deliveries are negatively affected in this component, which is representative of the problem since one of the major clients’ constraints consists of being visited by many vehicles.

The scenarios with minimization of Objective Function OF1 (C1 and C3) were severely penalized due to the occurrence of delays during a route’s progression. Those events were not possible in Figueiredo’s (2007) solutions because his planning consists on a reunion of four sub problems’ solutions (for computational complexity reduction purposes, as referred above). The division made split the clients into four clusters by geographical location and morning/afternoon time windows. That restricted the problem in ways that do not take place in the real planning but avoided the problem of dealing with delays caused by visiting clients with different time windows in the same route. The difference of a morning client’s due date and an afternoon client’s ready time corresponds to a delay that may vary up to several hours, as noted in table 7.

The chosen implementation tried to cope with this problem through the use of Objective Function OF2, whose minimization reduces the possibility of such occurrences. Finally, in table 8 it is presented the needed processing times to obtain the previous results.

This table points out that metaheuristics find very good solutions in a fraction of the time needed by an exact method. Furthermore, this is a good example of a median-sized problem that represents a small part of a real problem. Yet it required a division in four clusters to make possible to obtain results in acceptable time through an exact method. The total processing time for the problem in its full extent could take days, which is not compatible with the requirements of a daily operation. Therefore, the best solution for JM’s planning is scenario 4, which manages to successfully minimize all three components of the problem.

<table>
<thead>
<tr>
<th>Discrimination of operation times (min) - day 12</th>
<th>VST</th>
<th>FST</th>
<th>Travel Time</th>
<th>Total Delays</th>
<th>Total Route Duration</th>
<th>% JM Planning</th>
</tr>
</thead>
<tbody>
<tr>
<td>JM Planning</td>
<td>1048</td>
<td>860</td>
<td>3012</td>
<td>0</td>
<td>4920</td>
<td>-</td>
</tr>
<tr>
<td>Simple VRPSD (Figueiredo, 2007)</td>
<td>1048</td>
<td>1020</td>
<td>2648</td>
<td>0</td>
<td>4716</td>
<td>-4,1%</td>
</tr>
<tr>
<td>VRPSD (Figueiredo, 2007)</td>
<td>1048</td>
<td>1200</td>
<td>2738</td>
<td>0</td>
<td>4986</td>
<td>1,3%</td>
</tr>
<tr>
<td>Initial Solution</td>
<td>1048</td>
<td>780</td>
<td>2870</td>
<td>480</td>
<td>5178</td>
<td>5,2%</td>
</tr>
<tr>
<td>(C1) TS Guerreiro OF1</td>
<td>1048</td>
<td>780</td>
<td>2623</td>
<td>1297</td>
<td>5748</td>
<td>16,8%</td>
</tr>
<tr>
<td>(C2) TS Guerreiro OF2</td>
<td>1048</td>
<td>780</td>
<td>2750</td>
<td>0</td>
<td>4578</td>
<td>-7,0%</td>
</tr>
<tr>
<td>(C3) TS Guerreiro SD OF1</td>
<td>1048</td>
<td>1060</td>
<td>2538</td>
<td>2270</td>
<td>6916</td>
<td>40,6%</td>
</tr>
<tr>
<td>(C4) TS Guerreiro SD OF2</td>
<td>1048</td>
<td>940</td>
<td>2597</td>
<td>216</td>
<td>4801</td>
<td>-2,4%</td>
</tr>
</tbody>
</table>

Table 7 - Discrimination of operation times by scenario for day 12

<table>
<thead>
<tr>
<th>Elapsed Processing Time to the found best solution</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>Total</th>
<th>% VRPSD simples</th>
<th>% VRPSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple VRPSD (Figueiredo, 2007)</td>
<td>21792</td>
<td>14575</td>
<td>15638</td>
<td>16039</td>
<td>68044,00</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>VRPSD (Figueiredo, 2007)</td>
<td>22136</td>
<td>15155</td>
<td>22705</td>
<td>22698</td>
<td>82694,00</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>(C1) TS Guerreiro OF1</td>
<td>25,39</td>
<td>29,88</td>
<td>15,23</td>
<td>15,92</td>
<td>86,42</td>
<td>0,13%</td>
<td>0,10%</td>
</tr>
<tr>
<td>(C2) TS Guerreiro OF2</td>
<td>9,11</td>
<td>11,38</td>
<td>13,35</td>
<td>55,16</td>
<td>89,00</td>
<td>0,13%</td>
<td>0,11%</td>
</tr>
<tr>
<td>(C3) TS Guerreiro SD OF1</td>
<td>144,39</td>
<td>34,18</td>
<td>25,44</td>
<td>12,44</td>
<td>216,45</td>
<td>0,32%</td>
<td>0,26%</td>
</tr>
<tr>
<td>(C4) TS Guerreiro SD OF2</td>
<td>18,49</td>
<td>4,17</td>
<td>74,47</td>
<td>59,30</td>
<td>156,43</td>
<td>0,23%</td>
<td>0,19%</td>
</tr>
</tbody>
</table>

Table 8 – Elapsed processing time to obtain the presented results

6. Conclusions and Future Developments

This work reinforces the sustained good performance of heuristics when compared with exact methods. The different conditions of execution of both methods make a comparison more difficult, mainly because Figueiredo (2007) didn’t reach global optima as desirable for comparison purposes. Still, the obtained solutions are a big improvement of every component of the real planning. The results (table 4 to 7 and graph 1) show that the best planning must include Split Deliveries and the incorporation of a time component in the Objective Function, to prevent delays in the planning phase.

The benchmark instances solving for validation purposes proved that the implemented model can produce reliable results, even though its efficiency and robustness could be enhanced to increase the competitiveness of the developed model. Yet, the best development that could apply to this Case Study would be an implementation of this Model in JM’s distribution operation. Such goal would need to include a cost impact analysis and more profound study on the company’s processes.
One of the main goals of this work was to build a model able to be implemented on JM's operations. In that sense, there was a concern to make a user friendly interface and to address all of the problem's constraints – not only the Case Study problem proposed by Figueiredo (2007). A successful implementation of this model would need to consider mixed cargo routes in order to optimize the planning and at the same time minimize the total number of visits to clients. The latter has major implications on clients' normal functioning due to the small number of employees working daily at each client, as a result of the group's labor cut policy (Figueiredo, 2007).

Still, the need to deal with variability, prediction of warehouse outcome and additional operations such as backhauling, cross-docking e transshipment makes JM daily planning a challenge that will continue to attract academic interest.

**Nomenclature**

**MAX_TIME** termination criterion: limit time per run

**STOP** termination criterion: stop execution after

**STOP*MAX_TIME** of run without improvements

**MAX_RESTARTS** termination criterion: stop execution after **MAX_RESTARTS**

**LISTSIZE** size of neighbourhood list

**TABUSIZE** size of tabu list

**SPLIT_CRITERION** necessary % of client's demand to make a split

**Appendix A: description of the implemented TS Algorithm**

Initial Solution – \( S_1 \)

Current Solution – \( S_c \)

Best Solution Found – \( S_b \)

Lower iteration Limit – **RESTART_IT**

Upper iteration Limit – **UPPER_IT**

Iteration limit – **lim_it** \( \in [\text{RESTART_IT}; \text{UPPER_IT}] \)

Search Restart Limit from solution \( S_b – \text{MAX_RESTARTS} \)

Restart counter – \( r \in [0, \text{MAX_RESTARTS}] \)

Iteration counter – \( i \in [0; \text{MAX_RESTARTS} * \text{UPPER_IT}] \)

Iteration without improvement counter – \( i_t \in [0; \text{lim_it}] \)

Max Run Duration – **MAX_TIME**

Max Run Duration without improvement – **STOP*MAX_TIME**, **STOP \in [0; 1]**

**Step 1: Initialization**

\( S_1 \) is obtained through one of the methods described in section 4.1 or it is imported

Set \( S_0 = S_2 \)

Set \( \text{lim_it} = \text{RESTART_IT} \)

Set \( i_t = 0; i_t = 0; r = 0; \)

**Step 2: Neighborhood Exploration**

While none of the termination criteria is met:

Get neighborhood through one of the movements described in section 4.2;

Choose the fittest non-tabu candidate or a solution with fewer routes up to 5% worse;

Make it the current solution \( S_c \);

If \( f(S_c) < f(S_b) \), make the current solution the new best known solution: \( S_b = S_c; i_t = 0; \)

Else increment the iteration without improvement counter: \( i_t = i_t + 1; \)

Increment movement that created \( S_c \) into tabu list;

If \( i_t >= \text{lim_it} \), make search restart from the best known solution:

\[ r = r + 1; S_c = S_1; \]

\[ i_t = 0; \text{lim_it} = \text{lim_it} * 2; \]

If \( \text{lim_it} >= \text{MAX_RESTARTS} \), reinitialize the search extent: \( \text{lim_it} = \text{RESTART_IT}; \)

**Step 3: End of run**

Print solution \( S_b \);

Export run statistics;

Free memory/Close temporary files;

End of run.

**References**


