Logistics Research (2017) 10:7
DOI 10.23773/2017_7
Original Article

Tabu Search and Memetic Algorithms for a Real Scheduling and Routing Problem

Anass El-Yaakoubi [1] | Abdellah El-Fallahi [2] | Mohammad Cherkaoui [1] | Mohamed Reghioui Hamzaoui [2]

Received: 4 November 2016 / Accepted: 7 October 2017 / Published online: 20 November 2017

ABSTRACT

The Scheduling and Routing Problem (SRP) is a special case of the well-known vehicle routing problem, where a set of employees should perform an ensemble of services, at the clients' locations. In this work, we present a real SRP faced by a company of water and electricity distribution. Under some particular constraints, the company seeks to minimize the total distance traveled by its technicians. The services required by the clients consist of installation, maintenance and control operations. The problem is modeled as a mixed integer linear program. Three meta-heuristic algorithms based on tabu search methodology and memetic algorithm are proposed to solve it. These algorithms are tested on a set of real instances and others generated randomly. To measure the quality of the solutions obtained by our algorithms their results are compared to those given by CPLEX.

KEYWORDS Scheduling and routing problem· meta-heuristic algorithms· tabu search· memetic algorithm

Anass El-Yaakoubi
aelyaakoubi89@gmail.com

Abdellah El-Fallahi
aelfallahi@gmail.com

Mohammad Cherkaoui
cherkaoui66@gmail.com

Mohamed Reghioui Hamzaoui
m.reghioui@gmail.com

1 University AbdelMaleek Essaadi, National school of applied sciences of Tetouan
LIROSA, Mhannech II,
P.O. BOX 2222, Tetouan, Morocco

2 University AbdelMaleek Essaadi, National school of applied sciences of Tetouan
Research team MOSIL, Mhannech II, P.O. BOX 2222, Tetouan, Morocco

1. INTRODUCTION

In the classical Vehicle Routing Problem (VRP) client's request consists, generally, of an ensemble of goods. These goods are delivered to their corresponding clients using a fleet of vehicles based at a single or multiple depots. The VRP is originally introduced by Dantzig and Ramser [11] and aims to find a set of minimum cost to serve the clients respecting the constraints on the vehicles, clients, goods, and so on. Commonly, the cost of a route can be the total distance or the traveled time. During the latest years, different variants of VRP with different constraints have been tackled in the literature, such as the VRP with time windows by Homberger and Gehring [28], the multi-compartments VRP by El Fallahi et al. [17], the split delivery VRP by Silva et al. [41], the multi-depot VRP by Ho et al. [26], the periodic VRP by Hemmelmayr et al. [25], etc.
In order to solve the routing problems, a large number of techniques have been proposed in the literature. These techniques can be categorized into exact and heuristic algorithms. Many papers propose exact algorithms which obtain the optimal solutions. The most known exact algorithms for the VRP we find for example branch-and-bound by Zhang et al. [45], column generation by Dayarian et al. [12], constraints programming by Rousseau et al. [40], etc. Generally, the complexity of these algorithms grows exponentially with the problem size. For this reason, the heuristics methods are much more suitable to solve large scale combinatorial optimization problems. Thus, in the literature several approached algorithms have been proposed to find good, not necessary optimal, solutions in a reasonable running time of different combinatorial optimization problem like the VRP.
The approached algorithms can be divided into two main groups: heuristics and meta-heuristics. For the VRP, which belongs to the class of NP-hard problems as proven by Lenstra and Rinnooy [30], some of the well-known classical heuristics are the Clarke and Wright savings heuristic proposed by Clarke and Wright [9], the sweep algorithm by Gillett

& Miller [21] and the Fisher & Jaikumar algorithm [19]. On the other hand, many meta-heuristics have been applied to the VRPs, like tabu search by Nguyen et al [37], genetic algorithms by Baker and Ayechew [2], ant colony optimization by Bell and Mcmullen [3] and simulated annealing by Tavakkoli-Moghaddam et al. [43]. For an overview of the different variants of the VRP and the proposed solution methods in the literature, the reader can consult the paper published by Kumar and Panneerselvam [29].

The Scheduling and Routing Problem (SRP) under study is a special case of the VRP, in which the clients demand services and not goods. The services required by the clients can be maintenance operations, health care acts, installation of material and/or software in the telecommunication area, etc. A fleet of vehicles are used by employees to visit clients in order to carry out their services. Generally, in this type of real problems, the companies have to consider several constraints. These constraints are due to services diversification and the employees' competences. The main objective of the SRP is to allocate the employees to the services, and organize the routing of each employee at the same time, in order to optimize a fixed objective.

In this paper, we present a real SRP faced by a company of water and electricity distribution. The company seeks to determine an efficient scheduling and routing of its technicians in order to accomplish the clients' services. The required services include installation, maintenance and control operations. The scheduling must be performed according to the type and the time period of services required by the client, the qualification and the availability of technicians. Each client's service must be carried out by one technician. Therefore, every technician can accomplish various services by period, and the whole services fulfilled by a technician constitute its route. The objective is to minimize the total distance traveled by the technicians to perform all the services required.

To solve the SRP under study, we propose three meta-heuristic algorithms: a tabu search and two memetic algorithms. The first memetic algorithm is a classical genetic algorithm combined with a local search, and the second one is called memetic algorithm with wings.

This paper is organized as follows. Section 2 reviews the relevant literature on the SRP. Section 3 describes and formulates the SRP under study. As solution methods, three meta-heuristic algorithms based on tabu search methodology and genetic algorithm are presented in Section 3. The instances used to test the performance of the proposed algorithms and the obtained results are described and discussed in Section 4. The last section is devoted to the conclusions.

2. LITERATURE REVIEW

In this section, we review the literature of the SRP. The purpose is to highlight the important and relevant applications of SRP in the real-world. The SRPs addressed in the literature and the methods proposed to solve them are presented according to the areas of application.

Numerous practical applications of the SRP can be found in different areas. For example, in the medical area, the Home Health Care Problem (HHCP) is one of the most popular SRP. It consists of constructing routes and the roster for nurses providing some medical treatment for geographically spread patients. Cheng and Rich [7] treat a HHCP with the objective to minimize the amount of overtime and part-time work used. They consider that nurses take a lunch break within the nurse's lunch time window. As solution method, they present a two-phase algorithm. The first phase of the proposed algorithm is a parallel tour-building procedure, while the second phase seeks to improve the solutions obtained in the first phase.

Nowadays, some patients prefer to receive their medical treatments at home and the managers of the hospitals seek to increase the reception capacity of their institutes. For these reasons, the HHCP knows a remarkable development. Bertels and Fahle [4] deal with a HHCP in which the patients must be visited at their home within a time window respecting the work time limitations of nurses. Also, they consider the compatibility between services of patients and nurses specialties. The objective in their work is to minimize travel costs and maximize the satisfaction of patients and nurses. To solve the HHCP, the authors propose a combination of constraint programming and some heuristics and meta-heuristics as simulated annealing and tabu search among others.

In the services sector, the Domiciliary Care Problem (DCP) is an example of the SRP. The DCP differs from the HHCP in that it consists in carrying out the daily activities (i.e. cooking, bathing shopping, etc.) for elderly and/or disabled people by a set of home care workers [1,18,13]. Many solution methods for the DCP have been tackled in the literature like tabu search technique by Blais et al. [5], particle swarm optimization algorithm by Akjiratikarl et al. [1], a support system based on multi-agent system by Itabashi et al. [24], hyper-heuristics by Misir et al. [35].

In the maintenance area, Tricoire [44] presents a multi-period routing problem for service technicians faced by a company of water distribution and treatment. In his work the author considers two types of services; the first type is called appointment, while the second one is known as differential. The appointments are more critical than the differentials. In his paper, the author proposes two linear models for the SRP. The first model represents the case when the number of employees is insufficient to execute all services, while the second model deals with the opposite case. Then, he proposes some heuristics, a memetic algorithm used to solve the problem and near-optimal approaches based on column generation. Cordeau et al. [10] present a SRP known as Technician and Task Scheduling Problem (TTSP) faced by a company of telecommunications. In the TTSP, technicians perform a set of maintenance, installation tasks. The authors propose a construction heuristic and an adaptive large neighborhood search heuristic to solve the TTSP.

In the airports domain, the Manpower Allocation Problem (MAP) treated by Dohn et al. [14] is another example of the SRPs addressed in the literature. In the MAP, numerous teams carry out ground handling tasks including baggage handling and cleaning. The authors use column generation and Branch-and-Bound to solve the problem. In the security area, the Security Personnel Routing and Rostering (SPRR) presented by Misir et al. [36] can be seen as another example of the SRP in which security guards perform round of visits. As solution method, they propose a hyper-heuristic approach in their work. Chuin Lau and Gunawan [8] present a mathematical programming method for solving a problem of scheduling security teams to patrol. For additional information, we refer the reader to the recent survey by Castillo-Salazar et al. [6].

The SRP proposed in our work represents an extension of the problem treated by Tricoire [44]. Tricoire proposes two linear models; the first model represents the case when the number of resources is sufficient to execute all services, while the second model deals with the opposite case. In both cases the author differentiates between appointment and differentiable tasks, where the appointments are more critical than the differentiables. In our case, all the services required by the clients have the same importance, and then all the services have the same probability to be treated, and there is no separation between tasks. Also, in the problem treated by Tricoire, the meal location of each resource is chosen from a set of restaurants; in our case each resource has a predetermined meal location (restaurant or home).

In the SRP some clients need to be served within a time window, which can be a hard or soft constraints. The soft time window constraint may be violated, while the hard one has to be met. A resource arriving before the hard time window must wait, while starting the service after the time window is not allowed. Also, each resource in the SRPs, with multiple depots, can have a different starting and ending point (e.g. resource's home). Furthermore, the resource can use different transportation modes (heterogeneous fleet) like bicycles, public transport and cars to get to the client's location. Moreover, the resources can start and end their duty at the same point (central depot) or start from a point (home) and end at another one (central depot).

The most of SRPs addressed in the literature considered the skills of resources which are the qualifications needed to perform the services. Some SRPs have taken into account the working shift of resources, while others dealt with the daily capacity of resources which means the maximum number of working's hours per day. The availability of resources during the horizon of planning has been considered in some SRPs. Sometimes the number of resources is not enough to carry out all tasks, so the outsourcing of some tasks becomes necessary. In some SRPs, the services asked by customers must be synchronized, example in telecommunication area the hardware may be installed before the software.

The scheduling of a break (e.g. launch) for resources occurs in many SRPs. Some tasks must be performed within a number of validity's days in the planning horizon. In addition to the daily capacity of resources, some SRPs considered also the overtime. The priority of tasks has been also treated. In addition, some SRPs consider that all tasks must be scheduled, while in others cases some tasks can be uncovered. The splitting of some tasks between many resources is also dealt with. Performing some tasks needs more than one resource, so teams workers has been considered in some SRPs. Finally, some clients prefer to be served by certain resources, or the contrary. So, the preference resource/task has been considered in some SRPs. The characterization of the distinguishing elements of the cited problems is given in table one, in which the columns give the author's name of the cited works and rows present the different elements considered in each work.

3. PROBLEM DESCRIPTION AND FORMULATION

In this section, we present a detailed description of the SRP problem and we give the mathematical model corresponding.

3.1. Problem Description

The SRP is an abstract problem covering different applications that can occur in home care services, maintenance operations, security teams, etc. In our case, we deal with a real SRP faced by a company of water and electricity distribution. Every day, the company's technicians perform different types of services at the clients' locations. The offered services include installation, maintenance and control operations. A fleet of vehicles are used by the technicians to visit the clients. Each service should be assigned to only one available technician having the necessary skills to carry it out. Also, every service must be performed within a period of time.

Table 1: characterization of the relevant SRP in the literature

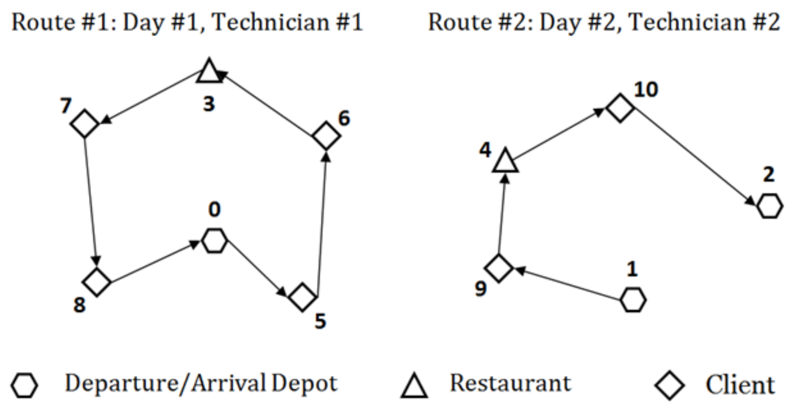
Characterizations / Authors names of cited works	Cheng and Rich (1998)	Bertels and Fahle (2006)	Tricoire (2006)	Akjiratikarl et al (2007)	Dohn et al (2009)	Eveborn et al (2009)	Cordeau et al (2010)	Misir et al (2010)	Misir et al (2011)	our problem
Hard time windows	X	X	X	X	X	X		X	X	X
Soft time windows		X								
Multiple depots	X		X	X				X	X	X
Possibility of starting and ending at the same point	X		X	X	X				X	X
Possibility of starting and ending at different point			X							X
Skills of resources is considered	X	X	X		X	X	X	X	X	X
Working shifts of resources	X	X			X			X		X
Working capacity of resources				X						
Availability of resources			X				X			X
Outsourcing	X				X		X		X	X
Synchronization of tasks					X		X	X		
Taking a break (break time is considered for lunch)	X	X	X			X				X
Validity days of tasks (within which the task must be done)			X							X
Overtime	X									
Priority of tasks (tasks more critical than others)			X				X			
Uncovered tasks (tasks can be unperformed)	X		X		X					
Split tasks					X					
Teaming (working in group)					X		X			
Preference (task/resource)		X				X		X		

The starting and ending locations can be the same, where all technicians start from the main center, or many locations (may be equal to the number of technicians) assuming each technician can start from their home. Also, each technician should take a meal at a restaurant or at home during their working shift. So, a triple (starting location, ending location, meal location) is known for each technician and these three locations can be the same for some workers. Given that we are considering a real problem, the service time for each service, including the meal time is estimated in advance using the data provided by the company. The daily working time of each technician shall not exceed the working shift.

The company seeks to minimize the total distance traveled by its technicians and affecting the right technician to each service. Therefore, the company aims to determine an efficient scheduling and routing of its technicians that carry out the clients' services over a planning horizon (e.g. one week).

Figure 1. Example of SRP

Figure one illustrates an example of SRP. In this figure, technician 1 takes route 1 on the first day of the planning horizon and serves four clients, denoted by 5-8, starting and ending the service at the same depot, denoted by zero, and taking a meal at a restaurant, denoted by 3. Technician two takes route two on the second day of the planning horizon and serves two clients, denoted by 9-10, starting the service from a departure depot node 1, ending at the arrival depot node 2, and taking a meal at a restaurant, denoted by 4.



Technicians: The technicians have different skills and qualifications. The skills can be related to the installation and maintenance of the electrical or water systems. Each technician is available for a certain number of days in the planning horizon. The availability days of technicians can range from one day to the whole planning horizon. In the remainder of this paper, we refer to the pair {technician, availability day} as resource. A working shift is associated to each resource and can vary from a half-day to a full-day. The working shift is represented with a time window.

The working shift is represented by a time window and associated to the starting and ending locations of the corresponding resource. Also, the resource must take a meal within a time window associated to the meal location. The duration of the meal is one hour, and it is defined as the service time of the meal location. The starting and ending locations require no service time. In this paper, the number of resources is considered as decision variable.

Services: The technicians perform different types of services at the clients' locations. The services include the installation, maintenance and control operations of electrical and water systems. Each service must be performed within validity days in the planning horizon. The validity days of a given service can range from one day to the whole planning horizon. Also, each service has a known service time and must start within a time window. The time window is defined by the earliest and the latest time in which the service must starts. If the resource arrives to the client’s location before the earliest time, it should wait until the beginning of the time window. Each client must be served by one and only one resource. The accordance between the resource and the client’s service is defined as: i. The compatibility between the necessary skills to accomplish this service and the technician’s skills,

ii. The consistency between the availability day of the technician and the validity days of the service.

3.2. Problem Formulation

In this work we have adapted the mathematical models proposed by Tricoire [44] taking into consideration the objective and the constraints of the SRP treated in our case. As we mentioned in section two, Tricoire proposes two linear models; the first model represents the case when the number of resources is sufficient to execute all services, while the second model deals with the opposite case. Since the number of resources is assumed to be sufficient in our case, then we are interested in the first mathematical model. Tricoire presents a general objective function which consists in minimizing the total travel cost. Therefore, the first modification is to replace the cost matrix with the distance matrix in the objective function.

Tricoire considers two types of services; the first type is called appointment, while the second one is known as differential. The appointments are more critical than the differentials. In this paper, all the services required by the clients have the same criticality. Since, we deal with only one type of services, then the two constraints impose the satisfaction of each type of services in the model of Tricoire have been replaced by only one constraint in our model. In the problem treated by Tricoire, the meal location of each resource is chosen from a set of restaurants, while we assume that each resource has a predetermined meal location. Therefore, we adapted the meal's constraint in Tricoire's model. In addition to the decision variables within the model of Tricoire, we consider one which gives the waiting time at each client. So, we added the waiting time decision variables (wik) (equations eight, nine and 10 in our model).

The SRP studied here is defined on a complete undirected graph G = (N, E), where N is the set of nodes and E gives the set of edges. The set of nodes N includes the customers, meal, starting and ending locations. The travel distance from a node i to another j is given by the Euclidean distance dij between them. The travel time from node i to node j is given by tij and used to check the time windows for each node. Tables two and three describe respectively the data and decision variables used to model the SRP.

Therefore, the adapted mathematical model of the SRP is given as follows:

$$Min \sum_{k\in\Delta}\sum_{i\in N}\sum_{j\in N} x_{ij}^{k} d_{ij} \tag{1}$$

$$s.t \sum_{i\in\Gamma\cup R} x_{d_k i}^{k} - \sum_{i\in\Gamma\cup R} x_{i a_k}^{k} = 0, \quad \forall k \in \Delta \tag{2}$$

$$\sum_{k\in\Delta} y_i^k = 1, \quad \forall i \in \Gamma \tag{3}$$

$$y_j^k - \sum_{i\in\Gamma\cup R\cup D} x_{ij}^{k} = 0, \quad \forall k \in \Delta, \forall j \in (\Gamma \cup R \cup A) \tag{4}$$

$$y_i^k - \sum_{j\in\Gamma\cup R\cup A} x_{ij}^{k} = 0, \quad \forall k \in \Delta, \forall i \in (\Gamma \cup R \cup D) \tag{5}$$

$$\sum_{j\in\Gamma\cup R} x_{d_k j}^{k} - \sum_{i\in\Gamma\cup D} x_{i a_k}^{k} = 0, \quad \forall k \in \Delta \tag{6}$$

$$y_i^k \leq o_{ik}, \quad \forall k \in \Delta, \forall i \in N \tag{7}$$

$$u_i^k + w_i^k + s_i + t_{ij} - M\left(1 - x_{ij}^k\right) \leq u_j^k, \quad \forall k \in \Delta, \forall i, j \in N \tag{8}$$

$$u_i^k + w_i^k + M\left(1 - y_i^k\right) \geq e_i, \quad \forall k \in \Delta, \forall i \in N \tag{9}$$

$$u_i^k + w_i^k - M\left(1 - y_i^k\right) \leq l_i, \quad \forall k \in \Delta, \forall i \in N \tag{10}$$

$$\sum_{i,j\in N} x_{ij}^k \leq |S| - 1, \quad \forall k \in \Delta, \forall S \subset N \text{ with } 2 \leq |S| \leq |N| - 2 \tag{11}$$

$$x_{ij}^k \in \{0,1\}, \quad \forall (i,j) \in E \tag{12}$$

$$y_i^k \in \{0,1\}, \quad \forall k \in \Delta, \forall i \in N \tag{13}$$

$$u_i^k \in \mathbb{R}^+, \quad \forall k \in \Delta, \forall i \in N \tag{14}$$

$$w_i^k \in \mathbb{R}^+, \quad \forall k \in \Delta, \forall i \in N \tag{15}$$

Table 2. Description of data

Notation	Description
Γ	Set of clients,
Δ	Set of resources,
D	Set of starting locations, the starting location associated with the resource k is noted by d_k,
A	Set of ending locations; the ending location associated with the resource k is noted by a_k. The starting (D) and ending(A) locations can be different
R	Set of meal locations; the meal location associated with the resource k is noted by r_k,
N	Set of nodes of the graph G representing the problem,
d_{ij}	Distance between the node i and j,
t_{ij}	Travel time between the node i and j,
s_i	Service time at the node i,
$[e_i, l_i]$	Time window at the node i, where e_i is the earliest time and l_i the latest time,
o_{ik}	Binary constant equal to 1 if the resource k and the service of the client i are compatible, 0 otherwise,
M	Large positive value.

Table 3. Description of decision variables

Variable	Description
y_i^k	Binary variable equal to 1 if the route associated with the resource k visits the node i, 0 otherwise,
x_{ij}^k	Binary variable equal to 1 if the edge (i, j) belongs to the route associated to the resource k, 0 otherwise,
u_i^k	Time at which the service begins in the node i using the resource k,
w_i^k	Waiting time at the node i using the resource k,

The objective function (1) gives the total distance traveled by the resources. Constraints (2) make sure that if the starting location is left by resource k, the ending location has to be visited and the other way round. The constraint (3) imposes that each client should be served by one resource. The constraints (4) and (5) ensure that each node has one incoming edge except the departure depot, and one outgoing edge except the arrival depot. The constraint (6) imposes visiting the restaurant for each resource. The compatibility between the resource and the client's service corresponds to the constraint (7). The constraints (8), (9) and (10) correspond to the respect of time windows. The constraint (11) is the sub-tour elimination constraint. The remaining constraints give the typology of the decision variables considered in this model. To solve the proposed model, we have developed three meta-heuristic algorithms based on tabu search methodology and genetic algorithm as described in the next section.

4. SOLUTION METHODS

It is well known that in the routing problems the high number of nodes and the complexity of real-life problem make the optimization problem NP-hard, and then the use of exact methods may be difficult to solve these problems in reasonable CPU times. Therefore, since the SRP is an NP-hard problem, we have proposed three meta-heuristics to solve it. Generally, the meta-heuristics algorithms produce good but not necessarily optimal solution. The first algorithm developed is based on tabu search methodology, while the two others algorithms are memetic algorithms. A detailed description of these algorithms is given in the following sections.

4.1. Tabu Search

In order to solve the SRP under study, we first propose a meta-heuristic algorithm based on tabu search (TS) methodology. The TS is a general heuristic procedure introduced by Glover [22, 23] to solve large scale optimization problems, for which the exact methods give no effective solutions. It has been applied successfully to solve several types of difficult combinatorial optimization problems [17, 37]. The strength of the TS resides in its capacity to escape the trap of local optimality, and therefore gives good solutions of hard combinatorial optimization problems.

TS algorithm starts from an initial solution, and then explores iteratively its neighborhood. Thus, the principal element that constitutes the TS methodology is neighborhood N(S) of an incumbent solution S. Indeed, each solution S' in the neighborhood N(S) is reached by a simple perturbation of the solution S. The neighborhood N(S) can contain solutions that deteriorate the quality of the current solution S. At each iteration, the best solution in N(S)\{S} is considered to continue the search. To the best of our knowledge, TS algorithm is one of the few optimization procedures that permit moves which deteriorate the quality of the current solution. Many types of moves for VRPs have been addressed in the literature, such as the λ-interchange by Osman [38], the k-opt by Lin [33], etc. In λ-interchange, maximum λ nodes may be interchanged between two different routes. The k-opt consists in crossing k edges in the same route or between two different routes.

The core of the TS is embedded in its short-term memory or tabu list (TL). The TL is used to prevent cycling to the recent visited solutions by recording them. Nevertheless, since recording a whole solution is space consuming, a better way of doing consists in memorizing only the recent moves in the TL. Then the future search forbids the reverse moves for a certain number of iterations. The number of iterations during which a move is taboo is known as TL size. Generally, the size of the TL is fixed, but sometimes can be considered dynamic to better assist the search [16].

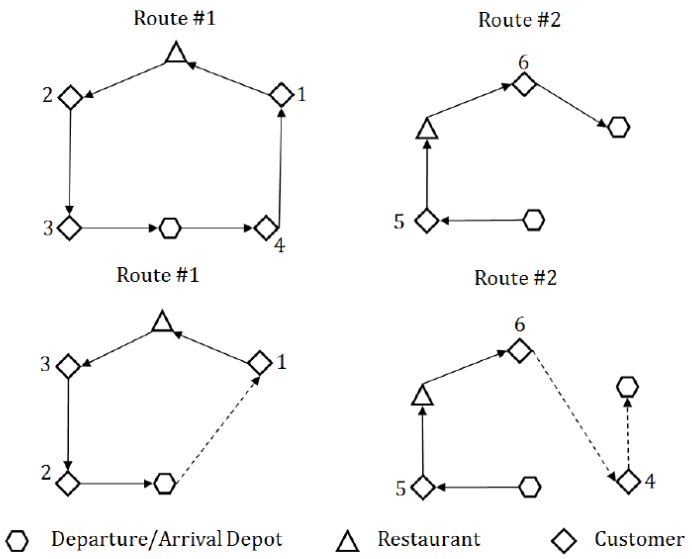
Initial Solution: The first stage of the TS consists in creating an initial solution using an adapted best insertion heuristic (BIH) proposed by Tricoire [44]. BIH is a construction heuristic which seeks to build a set of feasible routes by inserting each client into a route according to its insertion cost, starting by the client with the lowest cost. Since we use a limited number of resources, BIH starts by building an initial route for each resource including the starting, ending and meal nodes. Then, the insertion feasibility is examined for each unrouted client in a compatible route. The insertion of a given client, in a partially constructed route, is feasible if satisfies the necessary and sufficient conditions for time feasibility proved by Solomon [42]. In each iteration, the best possible insertion of a given client is performed. After the insertion of all clients, the unused routes are taken out. Algorithm one represents an overview of the BIH algorithm.

Moves and Neighborhood: In order to generate the neighboring solution, we use two types of moves, inter-route 1-interchange and intra-route 1-interchange. The inter-route 1-interchange move has the structure <C,R1,P1,R2,P2>, and consists of transferring the client C from the route R1 and the position P1 to the route R2 and the position P2. The intra-route 1-interchange move is a particular case of inter-route 1-interchange where R1 and R2 are the same route. Figure two represents an example of inter-route 1-interchange and intra-route 1-interchange moves. In figure two, client 4 is moved from route 1 to route 2, while the positions of clients 2 and 3 in route 1 are exchanged.

Algorithm 1 : general structure of the BIH

```
while there is an unrouted and feasible client do
    bestIns ← firstFeasibleIns()
    bestCost ← cost(bestIns)
    for each unrouted client i do
        for each compatible route j with i do
            for each position k in j do
                if (cost(i, j, k) < bestCost and
                    isInsFeasible(i, j, k) == True) then
                        bestIns ← ins(i, j, k)
                        bestCost ← Cost(bestIns)
                end if
            end for
        end for
    end for
    performIns(bestIns)
end while
```

Figure 2. Example of inter-route 1-interchange and intra-route 1-intechange moves



In the proposed TS, the neighborhood of a given solution S is represented by a list of feasible moves, called candidates list (CL(S)). Each element of the CL(S), if it applied, generates a new neighboring of the solution S. The CL considered in this work is a list of η elements. At each iteration, the CL(S) of the current solution S is created, and then the best non-tabu move is selected. To evaluate a move, we consider only the differential cost made by performing this move in the routes considered. The equation (16) illustrates the differential cost φ of a given move, where Ψ and Ω are respectively the sets of edges added and those removed when performing the move.

$$\varphi = \sum_{(i,j)\in\Psi} d_{ij} - \sum_{(i,j)\in\Omega} d_{ijj} \tag{16}$$

Tabu List: As we already mentioned, the TL is used to prevent return to already visited solutions. Generally, it records the recent visited solutions. Nevertheless, since recording a whole solution is space consuming, a better way of doing consists in memorizing a set of attributes able to characterize the solution. The TL considered in this work is a list of θ elements. In each iteration, the reverse of the performed move is added to the TL at the first position; the opposite of a given move M=<C,R1,P1,R2,P2> is M'=<C,R2,P2,R1,P1>. After adding a new element to the TL, its latest element becomes free, because its size is constant.

To avoid move aside some important moves, we have considered the aspiration criteria that allows tabu move when improves the best known solution. The equation (17) gives the necessary and sufficient aspiration condition, where Ccurrent, Cbest and φ are respectively the cost of the current solution, the cost of the best solution and the differential cost made by the tabu move. The search stops after αmax consecutive iterations without improving the best solution. Algorithm two gives a general structure of the proposed TS.

$$(C_{current} - \varphi) - C_{best} < 0 \tag{17}$$

Memetic Algorithm

The second proposed algorithm to solve the SRP is a memetic algorithm (MA). MA is a genetic algorithm [27] hybridized with a local search, to make it more efficient. Unlike tabu search, which is a single solution based method, MA is a population-based method. It repeatedly modifies a subset of the search space, called population, whose elements are represented by artificial chromosomes. The chromosome is a sequence of genes; each gene is encoded into a digital data and gives some information about the problem.

Algorithm 2 : general structure of the TS

```
currentSolution ← initialSolution
bestSolution ← initialSolution
bestCost ← solCost(initialSolution)
tabuList ← Ø
while count < iterationsNumber do
    candidatesList ← Ø
    createCandidatesList(currentSolution)
    bestMove ← selectBestMove(candidatesList)
    performMove(bestMove, currentSolution)
    updateTabuList(tabuList, bestMove)
    if solCost(currentSolution) < bestCost then
        bestSoltuion ← currentSolution
        bestCost ← solutionCost(currentSolution)
        count ← 0
    else
        count ← count + 1
    end if
end while
```

Different encoding techniques have been addressed in the literature. The most used encoding in the transportation area, we find binary encoding used by Zhu [46], where every gene is a bit (0 or 1), and permutation encoding presennted by Eiben and Smith [15], where every gene is an integer. After the encoding step, a fitness value is computed for each chromosome to evaluate the quality of its corresponding solution. To improve the quality of the current population, MA performs, generally, three operations; selection, crossover and mutation or local search. At each iteration, these operations evolve the actual population towards a new better one.

There are several commonly used selection operators in MA. The most popular selection operators used in the VRP are roulette wheel selection used by Gen and Chengi [20] and tournament selection presented by Michalewicz [34]. At the beginning, we select two parents from the actual population to be processed by the crossover and mutation operations. Then, the crossover uses the selected parents to reproduce new solutions, called offspring. As first step of the crossover, we chose random swap nodes in each parent, and then we exchange subsequences around these nodes to generate new offspring. There are many different ways that crossover can be applied depending on the number of crossover points, like one-point crossover used by Potvin and Bengio [39] and two swap nodes presented in Baker and Ayechew [2].

After the crossover is performed, mutation takes place. It is used to improve the quality of the new offspring, and also to escape the trap of local minima by applying a simple transformation to the chromosomes representing the offspring. When the mutation involves complex moves, it is called local search, and thus genetic algorithm becomes the memetic algorithm. As local search moves, we can cite the insertion, λ-interchange, or-opt and k-opt among others.

Encoding and Initial Population: The solution in the proposed MA is represented by an artificial chromosome. The chromosome is a list of routes which forms a given solution. Each gene in the chromosome is an integer number that represents a node number. In one route, the sequence of genes is the order of visiting nodes. The route must start with the starting node, finish with the ending node and contains the meal node. Therefore, the empty route (or the route of an unused resource) includes only these three nodes. Figure three illustrates a chromosome representing a solution to a SRP used in the MA. In figure three, routes R1 and R2 represent two used routes, while route R3 is an unused route. Nodes 1-3 represent the starting locations, nodes 4-6 are the meal locations, nodes 7-9 are the ending locations and nodes 10-15 are the clients.

Figure 3. A chromosome representing a solution to a SRP

R1:	1	12	14	4	10	13	7
R2:	2	11	5	15	8		
R3:	3	6	9				

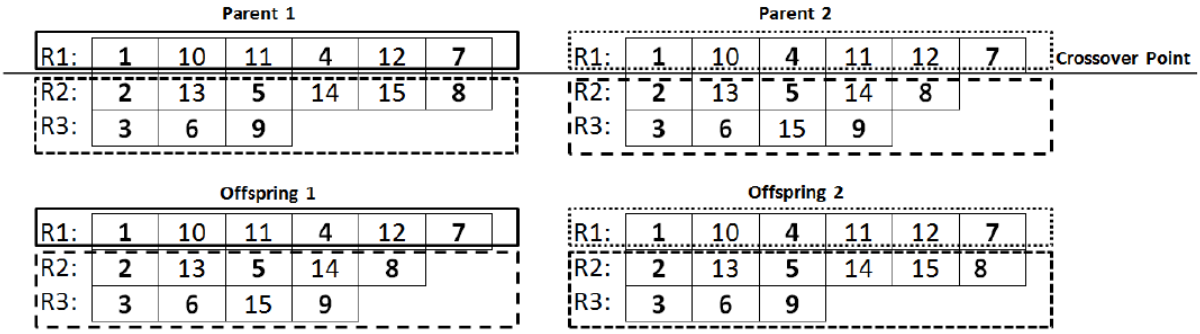
The first step of the proposed MA consists in generating randomly an initial population of psize individuals. Each individual in the population represents a feasible solution of the SRP. At the beginning, an empty route is built for all resources; an empty route contains only and exclusively the starting, ending and meal nodes. Then, for each unrouted client, a feasible route and an insertion position are chosen. This process is repeated until the insertion of all clients. Generally, the random processes used guarantee the generation of diverse solutions. Afterwards, we compute the fitness of each solution in the initial population as exposed in the following section.

Fitness and Selection: For each solution in the population, we compute a fitness value which is the total traveled distance of all used resources. The equation (18) gives the formula of the fitness value, where r, m, m', l and l' are respectively the list of routes, the number of used routes, the number of unused routes, the length of the used route and the length of the unused route. To improve the solutions in the initial population, we use the classical crossover and mutation operations described in the following section. Before applying the crossover and mutation, we select two parents from the population using the tournament selection. The tournament selection operator consists in choosing randomly two solutions from the population and then selects the best of them according to its fitness value. The selection operator is repeated twice to select two parents to continue the search.

$$\sum_{i=1}^{m}\sum_{j=1}^{l} d_{r[i][j-1],r[i][j]} - \sum_{i=1}^{m'}\sum_{j=1}^{l'} d_{r[i][j-1],r[i][j]} \qquad (18)$$

Crossover and Mutation: In their classical version, crossover operators exchange the subsequences of genes around the swap nodes of the selected parents. In this work, we propose a crossover operator that exchanges the sub-lists of routes around the crossover point. The crossover operator in the proposed MA starts by choosing random swap nodes from the list of routes that represents the whole solution. The first offspring inherits the sub-list of the routes before the crossover point from the first parent, and the sub-list of the routes after the crossover point from the second parent. The second offspring inherits the sub-list of the routes before the crossover point from the second parent, and the sub-list of the routes after the crossover point from the first parent. Figure four represents an example of the crossover operation used in this paper. In figure four, nodes 1-3 are the starting nodes, nodes 4-6 represent the ending locations, nodes 7-9 are the meal locations and nodes 10-15 are the clients.

Figure 4. Crossover operation



After the crossover operation is performed, some adjustment operations take place. The objective of these operations is to repair the infeasible solutions generated by the crossover. These infeasibilities can be caused by the duplication of clients in the same solution and/or the loss of some clients in the new solutions. In the case of duplication, we remove the repeated client from the position with greatest cost. In the case of absence, the best insertion of the client is performed. The adjustment operations are performed respecting the compatibility and the time window constraints. After the generation of two offspring by the crossover operation, the local search operation is applied to the new offspring. As local search process of the proposed MA, we use the inter-route and the intra-route 1-interchange moves described in Section 3.1. The local search first consists in creating a list of lsize random feasible moves. Then, the move with the best differential cost is chosen and applied. After some experiments, we noted that the application of one simple move gives a small effect on the fitness value and the algorithm's convergence is very slow. So, we apply nbmv moves.

Replacement Strategy and Stopping Criteria: In the proposed MA, we use a Replace-Worst strategy as replacement operator. It consists of replacing two worst individuals in the current population with the obtained offspring. The Replace-Worst is an effective replacement strategy that speeds the convergence of the algorithm keeping the diversity of the population. The MA stops when a maximum number βmax of generations is reached without improving the best solution. Algorithm three gives an overview of the the MA.

4.3. Memetic Algorithm with Wings

The last proposed solution method is also a memetic algorithm, named memetic algorithm with wings (MAW). MAW is based on the same operators used in the proposed MA, including selection, crossover, mutation and replacement operators.

In each generation of MAW, two parents are selected and used to produce two offspring. The process of producing the new offspring is repeated using the same parents until a maximum number γmax of iterations is reached. Then, the two best offspring found during this process replace the two worst individuals in the current population. This additional process guarantees the quality of the new generated solutions. Algorithm four gives an overview of the proposed MAW. In the next section, the instances used to test the performance of the proposed algorithms and the obtained results are described.

Algorithm 3 : general structure of the MA

```
currentPopulation ← generateInitialPopulatio()
currentGeneration ← 0
bestOverallSolution ← bestSolution(currentPopulation)
bestCost ← solutionCost(bestOverallSolution)
while currentGeneration < maxGenerations do
    P1, P2 ← selectParents(currentPopulation)
    O1, O2 ← crossover(P1, P2)
    O1, O2 ← repairOffspring(O1, O2)
    O1, O2 ← mutation(O1, O2)
    I1, I2 ← worstSolutions(currentPopulation)
    remove(I1, I2)
    add(O1, O2)
    bestSolution ← bestSolution(currentPopulation)
    if solutionCost(bestSolution) < bestCost then
        bestOverallSolution ← bestSolution
        bestCost ← solutionCost(bestSolution)
        currentGeneration ← 0
    Else
    currentGeneration ← currentGeneration + 1
    end if
end while
```

5. NUMERICAL EXPERIMENTS

To test the quality of TS, MA and MAW, we consider three sets of instances. The first set consisting of random small-sized instances (10 to 30 clients) that can be solved exactly by CPLEX optimization software. The small-sized instances are considered to compare the results obtained by the proposed algorithms with those provided by CPLEX. The second set of medium-sized instances (50 to 52 clients) is formed by a real instances proposed by the company. The third set is composed by random large-sized instances (100 to 300 clients) created randomly in order to compare and measure the performance of the proposed algorithms on large scale optimization problems

5.1. Experimental Environment and Parameter Settings

All tests are performed on a laptop computer equipped with an Intel Dual-Core processor with 2.30 GHz and 4 GB RAM, operating Windows 7 Home Premium. TS, MA and MAW were implemented in the C++ programming language. The parameters used to obtain the final results are presented in table four. The presented values are the result of intensive studies we conducted to fine-tune the proposed algorithms.

Algorithm 4 : general structure of the MAW

```
currentPopulation ← generateInitialPopulatio()
currentGeneration ← 0
bestOverallSolution ← bestSolution(currentPopulation)
bestCost ← solutionCost(bestOverallSolution)
while currentGeneration < maxGenerations do
    currentIteration ← 0
    P1, P2 ← selectParents(currentPopulation)
    while currentIteration < maxIterations do
        O1, O2 ← crossover(P1, P2)
        O1, O2 ← repairOffspring(O1, O2)
        O1, O2 ← mutation(O1, O2)
        B1, B2 ← updateBestOffsping(O1, O2)
        currentIteration ← currentIteration + 1
    end while
    I1, I2 ← worstSolutions(currentPopulation)
    remove(I1, I2)
    add(B1, B2)
    bestSolution ← bestSolution(currentPopulation)
    if solutionCost(bestSolution) < bestCost then
        bestOverallSolution ← bestSolution
        bestCost ← solutionCost(bestSolution)
        currentGeneration ← 0
    Else
    currentGeneration ← currentGeneration + 1
    end if
end while
```

5.2. Instances Description

To test the quality of the proposed algorithms, we have considered three sets of instances. The first set is constituted of five small-sized instances generated randomly and named A1, A2, A3, A4 and A5. The second set is composed of eight real-life medium-sized instances given by the company and named B111, B121, B122, B112, B211, B221, B222 and B212. The third set is formed of six random large-sized instances named C11, C21, C31, C12, C22 and C32. Table five presents the content of each instance. The number of clients, the length of the planning horizon (in days), the number of technicians, and the numbers of resources, are respectively given in columns Clients, Length, Technicians and Resources. For all instances, we assume that all technicians are available every day in the planning horizon. Therefore, the number of resources is equal to the number of technicians multiplied by the length of the planning horizon.

Table 4. Overview of the parameters used to obtain the final results

Algorithm	*Parameter*	*Set of Instances* A	B	C
TS	θ	7	35	35
	η	14	70	140
	α_{max}	300000	30000	30000
MA	p_{size}	20	20	20
	β_{max}	10000	1000	500
	l_{size}	50	100	200
	nb_{mv}	10	15	25
MAW	p_{size}	20	20	20
	β_{max}	330	20	10
	l_{size}	50	100	200
	nb_{mv}	10	15	25
	γ_{max}	30	50	60

In all random instances, the node's coordinates are randomly chosen on a square map measuring 1000 arbitrary units. The set of generated nodes includes the nodes representing the clients, the starting, ending and meal locations of each resource. The Euclidean distance is considered as the distance between the nodes. The travel time are computed by applying a given factor f to the distances. Equation (19) represents the formula used to calculate f, where nbkm, nbunits and avspeed are respectively the measurement of the map in kilometers, the measurement of the map in units and the average speed of the vehicle. If we suppose that 1000 units in the map is equivalent to 41 kilometers and the average speed of the vehicles is 35 km/h, then the factor f is equal to 0.07.

Table 5. Content of instances

Instance	Clients	Length	Technicians	Resources
A1	10	2	5	10
A2	15	2	7	14
A3	20	2	10	20
A4	25	2	12	24
A5	30	2	15	30
B111	50	1	48	48
B121	52	1	50	50
B122	52	1	50	50
B112	50	1	48	48
B211	50	1	6	6
B221	52	1	5	5
B222	52	1	5	5
B212	50	1	6	6
C11	100	1	15	15
C21	200	1	30	30
C31	300	1	45	45
C12	100	2	10	20
C22	200	2	20	40
C32	300	2	30	60

$$f = \frac{nb_{km} \times av_{speed} \times 60}{nb_{unit}} \qquad (19)$$

The service time, the time window and the validity days of the clients are randomly generated. The distribution of the service time and the time window is provided in table six, where the column Per represents the percentage of clients with the service time or time window given in the column Value. The validity days of each client are also randomly generated, and range between one day and the whole planning horizon; some clients are not available over the entire planning horizon, then they demand to be served in a given days of the week these days constitute their validity days. The service time of the meal node is one hour, while the starting and ending nodes require no service time. The time window of the meal, starting and ending nodes correspond to the working shift of the resource associated with them. All resources are assumed to work a full-day shift and have the necessary skills to perform all services; that means for each service requested, we can find a technician with the adequate skills to achieve it, and not each technician can execute all tasks.

Table 6. Distribution of the service time and the time window

Time Window		Service Time	
Per (%)	Value (min)	Per (%)	Value (min)
25	[0,300]	50	5
25	[300,540]	25	15
50	[0,540]	25	25

The real-life instances have the same structure as the random ones. They represent different situations faced by the company. The time windows of the clients in B121 and B221 are long, while they are short in B122 and B222. The locations used by the resources for starting and ending the duty and taking the meal in B111 and B211 are different from those in B112 and B212. Different working shifts are considered in the real instances, including the half-day and full-day shifts.

5.3. Results

The obtained results are reported in table seven, table eight and table nine. Table seven indicates the results obtained by CPLEX, TS, MA and MAW. The first column Instance gives the names of the instance used, columns clients, BestCost, Res and Time indicate respectively the number of clients and the best solution found by all approaches, the number of resources used to serve the clients and the running time. For the solutions obtained with CPLEX, the objective function value given in column Cost corresponds to the optimal solution, or the best upper bound found within 7200 seconds. The value with a star represents the best upper bound found by CPLEX. For the three meta-heuristic algorithms, columns Min(Cost), Max, Average and SD represent respectively the minimum, maximum, the average and the standard deviation of four executions of each algorithm for each instance. The column Dev denotes the gap to the best solution found by either the meta-heuristics or CPLEX.

The results show clearly the efficiency of all algorithms to solve the small-sized instances A1 and A2 giving an optimal solution. For the others small-sized instances, CPLEX was not able to provide the optimal solution and the result showed in table seven for the instances A3, A4 and A5 represents the best feasible solution given by CPLEX. For these last three instances, the proposed algorithms found better solution than the best bound provided by CPLEX within an acceptable time. MA found the best solution of the instances A3 and A4, while MAW provided the best solution of the instance A5.

For some medium-sized instances, CPLEX was not able to provide a feasible solution. For these instances, MA gives the best solution for the instances B121, B122 and B221, while the best solution for the others is provided by TS. For large-sized instances, TS provided the best solution of the instances C11, C12 and C22, while the best solution of the instance C31 is found by MA. MAW provided the best solution for the instances C21 and C32. Also, we notice that the proposed algorithms reduce the number of resources used to serve the clients. For example, they were able to reduce the number of resources used on the instance A3 from 4 resources, obtained by CPLEX, to only 3 resources obtained by meta-heuristics. TS found ten best solutions, MA provided eight best solutions and MAW obtained five best solutions.

Analyzing table seven, we observe that TS algorithm gives best average of the averages values (column Average) that equal to 10354.68 over the instances considered, while memetic algorithm obtains 10983.31 as average value and the average value of the MAW is equal to 11841.2. Also, TS has the smallest average of standard deviation 278.58, MA obtains 1620.99 and MAW obtains 1418.32. The results presented in tables seven confirm the performance of the TS algorithm comparing it with MA and MAW proposed in this work. Also, TS shows better stability than MA and MAW according to the standard deviation average of each algorithm.

Table eight gives the lower bound and the upper bound found using CPLEX without running time limit. this table shows that CPLEX is unable to give an upper bound for the six large scale instances (C11 to C32) and also for the B122 instance even if contains only 52 customers.

Table 8: lower and upper bound given by CPLEX without limiting running time

Instance	Clients	Resources	LB	UB
A1	10	2	3733,76	3733,76
A2	15	2	4286,47	4286,47
A3	20	3	6700,55	7348,75
A4	25	3	4739,46	5738,16
A5	30	4	5069,06	6738,19
B111	50	28	61,72	427,10
B112	50	9	61,72	166,60
B121	52	6	740,58	2561,50
B122	52			
B211	50	6	67,6	261,50
B212	50	6	67,7	143,90
B221	52	3	1040,09	1467,00
B222	52			
C11	100			
C12	100			
C21	200			
C22	200			
C31	300			

Generally, the SRP is a particular problem faced by a private company, and in the literature we didn't find instances of this problem with which we can compare our results, for this reason we have compared the results obtained by the proposed algorithms with those obtained by CPLEX using small instances; when CPLEX don't give optimal solution we have considered the upper bounds if it is exist. Otherwise, and with the objective to measure the quality of our algorithms over large instances we have adapted the TS and MA, which clearly perform MAW, to solve the classical VRP with time windows that is close to the SRP using twelve random instances from Solomon library.

Table nine reports the best results obtained by TS and MA for twelve random instances of Solomon's TWVRP, considering four executions of each method, are very close to the best known solution. The instances proposed by Solomon are classified into three classes that start by R, C and RC, we have selected randomly four instances of each class to measure the quality of the TS and MA algorithms. The column Instance gives the name of Solomon's instance considered; column Bestsol presents the best solution known for each instance as presented in http://web.cba.neu.edu/~msolomon/problems.htm , and columns TS and MA indicate respectively the solution obtained by TS and MA. The columns D(TS/Bestsol) and D(MA/Bestsol) give respectively the deviation between the solution obtained by TS and MA to the best known solution. The deviation between a method A and BestSol is given by Equation (19). The largest gap between the solutions obtained by our algorithms and the best known solution for the twelve instances is around 3% as we can see in the columns D(TS/BestSol) and D(MA/BestSol).

$$\frac{A-BestSol}{BestSol} * 100. \qquad (19)$$

Table 9: results of TS and MA using a set of twelve random instances of Solomon TWVRP

Instance	*BestSol*	*TS*	*MA*	*D(TS/bestSol)*	*D(MA/BestSol)*
R101	1645,79	1662,25	1686,93	1,00	2,50
R106	1251,98	1283,28	1264,50	2,50	1,00
R110	1096,72	1118,65	1127,43	2,00	2,80
R206	906,14	911,78	909,98	0,62	0,42
C101	828,94	838,47	836,89	1,15	0,96
C105	828,94	828,94	828,94	0,00	0,00
C108	828,94	851,98	853,69	2,78	2,99
C203	591,17	607,60	601,45	2,78	1,74
RC103	1261,67	1268,90	1283,78	0,57	1,75
RC106	1424,73	1440,40	1463,20	1,10	2,70
RC108	1139,82	1174,01	1166,04	3,00	2,30
RC204	798,41	821,48	824,45	2,89	3,26

CONCLUSION

In this paper, we have presented a real-life SRP faced by a company of water and electricity distribution. The SRP has been modeled by a MILP, to solve this model we have developed three meta-heuristics algorithms TS, MA and MAW. To measure the quality of the proposed algorithms we have compared their results with the optimal or the best known solution given by the CPLEX optimizer for small and medium size problems. For large instances, for which CPLEX cannot find feasible solution, we have compared the results obtained by TS, MA and MAW between them. Also, we have adapted our algorithms to solve the classical TWVRP considering six instances from the Solomon ones. The results obtained by our algorithms over the twelve instances of Solomon show that the proposed algorithms are able to give a good solution. Also, the results presented in table seven indicate that TS provides better solutions within a reasonable running time by comparison with MA and MAW considering the average time of four executions of each method. Moreover, analyzing the standard deviation of the results obtained by each method we observe that TS presents better stability than MA and MAW.

Compliance with Ethical Standards:

The authors declare that they have no conflict of interest.

REFERENCES

Akjiratikarl, C., Yenradee, P., & Drake, P. R. (2007). Pso-based algorithm for home care worker scheduling in the UK. Computers & Industrial Engineering, 53(4), 559–583.

Baker, Barrie M., & M. A. Ayechew. (2003). A genetic algorithm for the vehicle routing problem. Computers and Operations Research, 30 (5), 787–800.

Bell, John E., & Patrick R. Mcmullen. (2004). Ant colony optimization techniques for the vehicle routing problem. Advanced Engineering Informatics, 1 (18), 41–48.

Bertels, Stefan, & Torsten Fahle. (2006). A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem. Computers and Operations Research, 33 (10), 2866–2890.

Blais, M., Lapierre, S. D., & Laporte, G. (2003). Solving a home-care districting problem in a urban setting. Journal of the Operational Research Society, 54(11), 1141–1147.

Castillo-Salazar, J. A., Landa-Silva, D., and Qu, R. (2014). Workforce scheduling and routing problems: literature survey and computational study. Annals of Operations Research.

Cheng, E., & Rich, J. L. (1998). A home health care routing and scheduling problem. Technical report CAAMTR98-04. Rice University.

Chuin Lau, H., & Gunawan, A. (2012). The patrol scheduling problem. In Proceedings of the 9th international conference on the practice and theory of automated timetabling (PATAT 2012), Son, Norway (pp. 175–192).

Clarke, G., & J. W. Wright. (1964). Scheduling of vehicles from a central depot to a number of delivery points. Operations Research, 12 (4), 568–581.

Cordeau, J. F., Laporte, G., Pasin, F., & Ropke, S. (2010). Scheduling technicians and tasks in a telecommunications company. Journal of Scheduling, 13(4), 393–409.

Dantzig, G. B., & J. H. Ramser. (1959). The truck dispatching problem. Management Science, 6 (1), 80–91.

Dayarian, Iman, Teodor Gabriel Crainic, Michel Gendreau, & Walter Rei. (2015). A column generation approach for a multi-attribute vehicle routing problem. European Journal of Operational Research, 241 (3), 888–906.

Dohn, A., Rasmussen, M. S., Justesen, T., & Larsen, J. (2008). The home care crew scheduling problem. In K. Sheibani (Eds.), Proceedings of the 1st international conference on applied operational research. (ICAOR'08), Yerevan, Armenia. Lecture Notes in Management Science, (Vol. 1, pp. 1–8).

Dohn, A., Kolind, E., & Clausen, J. (2009). The manpower allocation problem with time windows and jobteaming constraints: A branch-and-price approach. Computers & Operations Research, 36(4), 1145–1157.

Eiben, A. E., and James E Smith. (2003). Introduction to evolutionary computing. Springer.

El-Fallah, Abdellah, & Rafael Martí. (2003). Computational modeling and problem solving in the networked world: Interfaces in computer science and operations research, eds. Hemant K. Bhargava & Nong Ye, 79–96. Boston, MA: Springer. ISBN 978-1-4615-1043-7.

El-Fallahi, Abdellah, Christian Prins, & Roberto Wolfler Calvo. (2008). A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. Computers and Operations Research, 35 (5), 1725–1741.

Eveborn, P., Rönnqvist, M., Einarsdóttir, H., Eklund, M., Lidén, K., & Almroth, M. (2009). Operations research improves quality and efficiency in home care. Interfaces, 39(1), 18–34.

Fisher, Marshall L., & Ramchandran Jaikumar. (1981). A generalized assignment heuristic for vehicle routing. Networks, 11 (2), 109–124.

Gen, Mitsuo, & Runwei Chengi. (1997). Genetic algorithms and engineering design. Wiley Interscience.

Gillett, Billy E., & Leland R. Miller. (1974). A heuristic algorithm for the vehicle-dispatch problem. Operations Research, 22 (2), 340–349.

Glover, Fred. (1989). Tabu search - part {I}. ORSA Journal on Computing, 1 (3), 190–206.

Glover, Fred. (1990). Tabu search - part {II}. ORSA Journal on Computing 2, 2 (1), 4–32.

Itabashi, G., Chiba, M., Takahashi, K., & Kato, Y. (2006). A support system for home care service based on multi-agent system. Proceedings of the fifth international conference on information, communications and signal processing, 2005 (pp. 1052–1056).

Hemmelmayr, Vera C., Karl F. Doerner, & Richard F. Hartl. (2009). A variable neighborhood search heuristic for periodic routing problems. European Journal of Operational Research, 195 (3), 791–802.

Ho, William, George T. S. Ho, Ping Ji, & Henry C. W. Lau. (2008). A hybrid genetic algorithm for the multi-depot vehicle routing problem. Engineering Applications of Artificial Intelligence, 21 (4), 548–557.

Holland, John H. (1992). Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control and artificial intelligence. Cambridge, MA, USA: MIT Press.

Homberger, Jorg, & Hermann Gehring. 2005. A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. European Journal of Operational Research, 162 (1), 220–238. Logistics: From Theory to Application.

Kumar, S., & R. Panneerselvam. (2012). A survey on the vehicle routing problem and its variants. Intelligent Information Management, 4 (3), 66–74.

Lenstra, J. K., & A. H. G. Rinnooy Kan. (1981). Complexity of vehicle and scheduling problems. Networks, 11 (2), 221–227.

Li, Y., Lim, A., & Rodrigues, B. (2005). Manpower allocation with time windows and job-teaming constraints. Naval Research Logistics, 52(4), 302–311.

Lim, A., Rodrigues, B., & Song, L. (2004). Manpower allocation with time windows. Journal of the Operational Research Society, 55(11), 1178–1186.

Lin, S. (1965). Computer solution of the traveling salesman problem. Bell System Technical Journal, 10 (10), 2245–2269.

Michalewicz, Z. (1995). Genetic algorithms + data structures = evolution programs. Springer.

Misir, M., Verbeeck, K., De Causmaecker, P., & Vanden Bergue, G. (2010). Hyper-heuristics with a dynamic heuristic set for the home care scheduling problem. In Proceedings of IEEE congress on evolutionary computation (CEC) (pp 18–23).

Misir, M., Smet, P., Verbeeck, K., & Vanden Bergue, G. (2011). Security personnel routing and rostering: A hyper-heuristic approach. Proceedings of the 3rd International Conference on Applied Operational Research, ICAOR11, Istanbul, Turkey (pp. 193–206).

Nguyen, Phuong Khanh, Teodor Gabriel Crainic, & Michel Toulouse. (2013). A tabu search for time-dependent multi-zone multi-trip vehicle routing problem with time windows. European Journal of Operational Research, 231 (1), 43–56.

Osman, Ibrahim Hassan. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problems. Annals of Operations Research, 41 (4), 421–451.

Potvin, Jean-Yves, & Samy Bengio. (1996). The vehicle routing problem with time windows part ii: Genetic search. INFORMS Journal on Computing, 8 (2), 165–172.

Rousseau, Louis-Martin, Michel Gendreau, & Gilles Pesant. (2002). Using constraint-based operators to solve the vehicle routing problem with time windows. Journal of Heuristics, 8 (1), 43–58.

Silva, Marcos Melo, Anand Subramanian, & Luiz Satoru Ochi. (2015). An iterated local search heuristic for the split delivery vehicle routing problem. Computers and Operations Research, 53, 234– 249.

Solomon, Marius M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research, 35 (2), 254–265.

Tavakkoli-Moghaddam, R., N. Safaei, M. M. O. Kah, & M. Rabbani. (2007). A new capacitated vehicle routing problem with split service for minimizing fleet cost by simulated annealing. Journal of the Franklin Institute, 344 (5), 406–425.

Tricoire, Fabien. (2006). Vehicle and personnel routing optimization in the service sector: application to water distribution and treatment (Thesis). Retrieved from

Zhang, Zizhen, Hu Qin, Wenbin Zhu, & Andrew Lim. (2012). The single vehicle routing problem with toll-by-weight scheme: A branch-and-bound approach. European Journal of Operational Research, 220 (2), 295–304.

K. Q. Zhu. (2003). A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time windows. Tools with Artificial Intelligence. Proceedings. 15th IEEE International Conference on, 176-183.

Table 7: The results obtained by CPLEX, TS, MA and MAW

Instance	Clients	BestCost	CPLEX				TS						MA						MAW					
			Res	Cost	Time(s)	Dev(%)	Res	Min (Cost)	Max	Average	Time(s)	Dev(%)	Res	Min (Cost)	Max	Average	Time(s)	Dev(%)	Res	Min (Cost)	Max	Average	Time(s)	Dev(%)
A1	10	3733,76	2	**3733,76**	175,00	0,00	2	**3733,76**	3733,76	3733,76	121,96	0,00	2	**3733,76**	4438,11	4092,29	221,83	0,00	2	**3733,76**	4488,88	4102,97	212,06	0,00
A2	15	4286,47	2	**4286,47**	2399,00	0,00	2	**4286,47**	4637,13	4422,90	333,26	0,00	2	**4286,47**	4773,57	4482,69	277,20	0,00	2	**4286,47**	5649,13	4972,58	337,05	0,00
A3	20	7266,09	4	8188,90*	7200,00	12,70	3	7332,66	7704,93	7571,32	383,27	0,92	3	**7266,09**	7782,97	7617,29	493,59	0,00	3	7419,12	8390,15	7912,44	460,49	2,11
A4	25	6513,70	5	8520,82*	7200,00	30,81	3	7031,35	7404,91	7258,67	558,11	7,95	4	**6513,70**	7064,42	6686,65	649,44	0,00	3	6995,12	7929,92	7295,81	592,56	7,39
A5	30	6597,71	4	10170,00*	7200,00	54,14	4	7874,34	8091,93	7966,93	606,12	19,35	4	7000,76	7446,65	7140,02	828,62	6,11	4	**6597,71**	7342,09	7013,16	1235,22	0,00
B111	50	96,60	28	427,10*	7200,00	342,13	5	**96,60**	98,20	97,48	307,43	0,00	5	100,60	114,20	106,75	368,87	4,14	5	103,00	112,10	108,45	972,19	6,63
B112	50	88,70	9	166,60*	7200,00	87,82	5	**88,70**	91,00	89,80	425,79	0,00	5	90,00	97,40	94,83	392,60	1,47	6	97,20	103,10	100,03	675,33	9,58
B121	52	1278,60	6	2561,50*	7200,00	100,34	4	1356,50	1366,60	1363,38	472,70	6,09	6	**1278,60**	1576,60	1401,23	457,47	0,00	6	1379,10	1492,50	1432,80	799,42	7,86
B122	52	1409,40	-	-	7200,00	-	7	1414,90	1432,20	1423,73	297,46	0,39	6	**1409,40**	1476,30	1443,45	389,71	0,00	7	1412,00	1717,00	1561,58	1035,95	0,18
B211	50	93,00	6	261,50*	7200,00	181,18	5	**93,00**	96,50	93,88	133,41	0,00	5	97,80	109,70	101,30	98,54	5,16	5	101,80	110,40	105,83	169,50	9,46
B212	50	86,80	6	143,90*	7200,00	65,78	5	**86,80**	89,30	87,63	91,36	0,00	5	88,40	93,80	90,30	101,99	1,84	4	89,20	100,20	94,40	183,63	2,76
B221	52	1319,90	3	1467,00*	7200,00	11,14	4	1330,50	1357,20	1349,35	77,13	0,80	4	**1319,90**	1558,30	1454,68	73,03	0,00	4	1389,50	1927,80	1649,53	135,28	5,27
B222	52	1425,70	-	-	7200,00	-	5	**1425,70**	1425,70	1425,70	122,92	0,00	5	1526,80	1665,90	1594,25	160,74	7,09	5	1453,10	2307,90	1857,60	262,90	1,92
C11	100	11144,10	-	-	7200,00	-	6	**11144,10**	12861,30	11593,75	303,61	0,00	9	13185,90	16565,00	14191,23	483,67	18,32	8	11980,70	18553,60	16020,58	1264,74	7,51
C12	100	16130,40	-	-	7200,00	-	9	**16130,40**	17075,30	16545,43	330,42	0,00	11	19923,00	22766,00	21180,53	398,21	23,51	11	18840,70	24123,50	22368,90	1091,53	16,80
C21	200	23133,70	-	-	7200,00	-	13	23383,20	25412,90	24530,15	599,26	1,08	15	23133,70	24793,50	23856,60	1386,51	0,00	16	**23133,90**	30301,80	27426,70	5545,46	0,00
C22	200	27300,80	-	-	7200,00	-	14	**27300,80**	29619,70	28893,35	1328,31	0,00	20	31471,30	32891,30	32240,95	1510,99	15,28	16	27429,40	39461,10	34573,33	7600,20	0,47
C31	300	32811,80	-	-	7200,00	-	20	36488,30	37792,30	37136,53	873,90	11,20	21	**32811,80**	38649,70	36941,58	4457,81	0,00	21	34566,00	45052,50	40586,38	20686,63	5,35
C32	300	37710,90	-	-	7200,00	-	20	40032,40	42074,70	41155,33	1032,26	6,16	27	42170,50	44168,50	43517,45	4653,24	11,83	25	**37710,90**	51496,30	45799,90	18192,69	0,00