

TWO META-HEURISTICS FOR SOLVING THE MULTI-VEHICLE MULTI-COVERING TOUR PROBLEM WITH A CONSTRAINT ON THE NUMBER OF VERTICES

Manel KAMMOUN

*MODILS, Faculty of Economic Sciences and Management of Sfax, Route de
l'aéroport km 4, Sfax 3018 Tunisie
kamounemanel@gmail.com*

Houda DERBEL

*MODILS, Faculty of Economic Sciences and Management of Sfax, Route de
l'aéroport km 4, Sfax 3018 Tunisie
derbelhouda@yahoo.fr*

Bassem JARBOUI

*Higher Colleges of Technology, Abu Dhabi, UAE
bassem.jarboui@yahoo.fr*

Received: January 2020 / Accepted: May 2020

Abstract: In this work we deal with a generalized variant of the multi-vehicle covering tour problem (m-CTP). The m-CTP consists of minimizing the total routing cost and satisfying the entire demand of all customers, without the restriction of visiting them all, so that each customer not included in any route is covered. In the m-CTP, only a subset of customers is visited to fulfil the total demand, but a restriction is put on the length of each route and the number of vertices that it contains. This paper tackles a generalized variant of the m-CTP, called the multi-vehicle multi-covering Tour Problem (mm-CTP), where a vertex must be covered several times instead of once. We study a particular case of the mm-CTP considering only the restriction on the number of vertices in each route and relaxing the constraint on the length (mm-CTP-p). A hybrid metaheuristic is developed by combining Genetic Algorithm (GA), Variable Neighborhood Descent method (VND), and a General Variable Neighborhood Search algorithm (GVNS) to solve the problem. Computational experiments show that our approaches are competitive with the Evolutionary Local Search (ELS) and Genetic Algorithm (GA), the methods proposed in the literature.

Keywords: Covering Tour Problem, Multi-Vehicle Multi-Covering Tour Problem, General Variable Neighborhood Search Algorithm, Genetic Algorithm, Variable Neighborhood Descent.

MSC: 90B85, 90C26.

1. INTRODUCTION

Logistics problems receive a lot of attention from the academic community due to their economic and social importance. Such problems consist of the distribution of goods between depots and users under certain constraints while satisfying the customer demands. Unfortunately, several constraints prevent the service to be quick and equitable such as the restrictions on time, budget, resource availability, which makes the visit of all customers almost impossible. In order to satisfy the demands within an acceptable distance from a visited customer, a covering concept was introduced. Finding the best location of a set of distribution centers among a set of candidate sites such that all customers can reach at least one center lead to formulation of the location of post boxes problem[1], and milk collection points [2]. Generally, the choice of appropriate stops such that all customers are within a reasonable distance from these stops minimize the total routing cost ([3],[4]).

Different optimization methods have been developed for the routing problems. But, Covering Tour Problem (CTP), including covering constraints, despite its importance in real word applications, got much less attention. The one-vehicle version of the problem (1-CTP) was introduced the first time in [5]. Then, in [6] the same problem was solved exactly by a branch-and-cut algorithm and a heuristic method was proposed. This problem was solved later in [7] where, the authors developed a two-commodity flow formulation and a scatter search algorithm.

Recently, many researchers expanded the scope of their research to the multi-vehicle version (mm-CTP). The problem has been solved in [8] exactly by introducing a three-index vehicle flow formulation and three heuristics. In addition, exact and approximate methods are proposed in [10] to solve the m-CTP without the length constraints on each route (m-CTP-p). The authors developed a branch-and-cut algorithm based on two-commodity flow formulation and a hybrid method, which combines the Greedy Randomized Adaptive Search Procedure (GRASP) and Evolutionary Local Search (ELS). More recently, in [12] and [11], a Variable Neighborhood Search (VNS) based on Variable Neighborhood Descent (VND) method was proposed to solve the m-CTP-p, and the results proved to be better than those in [10].

Most of the earlier studies focused on one single coverage of a customer, whereas in some situations, a customer must be covered several times to be completely served. We take into account multiple coverage in order to serve the entire demand of each customer. This problem is referred to as mm-CTP (multi-vehicle multi-covering Tour Problem). Pham et al. [13] address a particular case of the mm-CTP where the length constraints are relaxed. The problem is called mm-CTP-p, where p represents the maximum number of vertices in each route. They formulate the problem as an integer linear program and develop a branch-and-cut

algorithm as well as a Genetic Algorithm (GA). To evaluate the performance of their proposed GA, they use the Greedy Randomized Adaptive Search Procedure and Evolutionary Local Search (GRASP-ELS), proposed in [10], as a reference method where they change several components of the algorithm.

In this work, first we develop a Population-based approach to solve the mm-CTP-p. More precisely, we use a hybrid approach that combines GA and VND methods which we call (GA-VND). Second, we propose a General Variable Neighborhood Search algorithm (GVNS) as a solution-based approach. Therefore, several components of the VNS algorithm must be modified to deal with the considered problem. The two meta-heuristics were compared to each other and with other approaches from the literature [13].

The remainder of the paper is organized as follows. In section 2, we describe the problem statement. In sections 3 and 4 we outline the developed meta-heuristics GA-VND and GVNS in detail, respectively. In section 5, a comparative study is performed, and finally, we conclude with section 6.

2. PROBLEM DESCRIPTION

In this section, we give the formal description of the mm-CTP-p. The m-CTP is defined in [13] as an undirected graph $G = (V \cup W, E_1 \cup E_2)$. The vertex set is $(V \cup W)$, where $V = \{v_0, v_1, \dots, v_{(n-1)}\}$ is the set of n vertices that can be visited, $T \subset V$ is the set of vertices that must be visited, and $W = \{w_1, w_2, \dots, w_l\}$ is the set of vertices that must be covered. Let m be the number of identical vehicles located at the depot v_0 . A unit demand is associated with each vertex of $V \setminus v_0$ and the total demands are delivered by m vehicles with a limited capacity p . The edge set is $E_1 \cup E_2$, where $E_1 = \{(v_i, v_j) : v_i, v_j \in V, i < j\}$ and $E_2 = \{(v_i, v_j) : v_i \in V \setminus T, v_j \in W\}$. A length c_{ij} is associated with each edge of E_1 and a distance d_{ij} is associated with each edge of E_2 .

In the m-CTP, we must select a subset of vertices among all vertices in a graph such that covering constraints are reached. The m-CTP consists of minimizing the total cost so that the following constraints are respected:

- Each route starts and ends at the depot;
- Each mandatory location is visited exactly once while each optional location is visited at most once;
- Each unreachable location is covered by at least one route, i.e., it lies within a predefined distance of at least one visited location;
- The number of vertices on each route (excluding the depot) is less than a given value p ;
- The length of each route does not exceed a fixed value q .

In this article we deal with a special case of the mm-CTP in which the constraints on the length of each route are relaxed (i.e. $q = +\infty$). The problem is called mm-CTP-p. The mm-CTP is a generalization of the m-CTP version. Indeed, Pham et

al. [13] generalized the m-CTP, introducing a new problem that takes into account many real-world situations where the entire demand of some locations is too large and cannot be served by a single coverage. This generalization is applied in the same way as for the Covering Salesman Problem (CSP) [14]. In the generalized variant (mm-CTP), the authors define the coverage demand U_k as the number of times a customer w_k should be covered. It is clear that the mm-CTP is an NP-hard problem since it reduces to a m-CTP when the coverage demand equals to 1 ($U_k = 1 \forall w_k \in W$).

Pham et al. [13] define the problem by an extended graph $\bar{G} = (\bar{V} \cup W, \bar{E}_1 \cup E_2)$ based on the original graph G , where $\bar{V} = V \cup \{v_n\}$, $V' = \bar{V} \setminus \{v_0, v_n\}$ and v_n is a copy of the depot v_0 , $\bar{E}_1 = E_1 \cup \{(v_i, v_n), v_i \in V'\}$ and $c_{in} = c_{0i}, \forall v_i \in V$. They discuss three variants of the problem: the binary version where a vertex is visited at most once, the problem with overnight where it is freely permitted to consecutively revisit a vertex, and in the third variant, each vertex can be visited by the tour more than once but only after passing through another vertex, called the variant without overnight. They propose graph transformations to reduce the last two variants and convert them into the binary one. Then, they solve the mm-CTP-p exactly by a branch-and-cut algorithm and approximately by two meta-heuristics. For the first approach they use the GRASP-ELS proposed in [10] as a reference method, and the second approach was adapted from the genetic algorithm proposed in [15]. They investigate the performance of their meta-heuristics on six variants of the mm-CTP (m-CTP-p, m-CTP, mm-CTP-p, mm-CTP, mm-CTP-o, mm-CTP-wo). The idea underlying the introduction of multiple coverage can be explained by the need to visit more vertices of V to satisfy the covering constraints. The number of time a vertex w_k of W must be covered is denoted by U_k . This number of coverages is calculated as follows:

$$U_k \in [1, \min(3, nb_k)]$$

where nb_k represents the maximum number of vertices in V which can cover w_k .

3. HYBRID GA-VND FOR THE mm-CTP-p

In this section, we develop a hybrid GA-VND for solving our problem. Below, we give the different steps of VND and GA. The evaluation function F of each solution is defined as the sum of the routing and covering costs as follows: $F(S) = \text{Routing Cost}(S) + \text{Covering Cost}(S)$. The fitness of a solution for our case represents only the total routing cost since all the vertices of W are covered without cost.

3.1. The Variable Neighborhood Descent procedure

In this work we use the VND method as a subordinate in our proposed VNS and GA meta-heuristics. The idea behind VND is to change neighborhoods in a deterministic way: VND starts with the first neighborhood structure and performs local search until no further improvements are possible to get a local optimum. Then, it

continues the local search with the second neighborhood structure. The key idea of this method is to explore the search area and to escape from the local minimum of a given neighborhood by changing to another neighborhood. This method consists of finding the best neighbor S_2 of a given solution $S_1, S_2 \in N_{k'}(S_1)$, where $N_{k'}$ represents the neighborhood structure. If the obtained solution S_2 is better than S_1 , then $S_1 \leftarrow S_2$ and $k' = 1$ otherwise, the VND continues the search with the next neighborhood $N_{k'+1}$. The chances to reach a global minimum increase when using several neighbourhood structures instead of one in the descent phase of the VND method. The different steps of the VND algorithm are summarized in Algorithm 1. The VND method used in our approaches consider the following

Algorithm 1 Steps of the VND algorithm

```

1: Input: initial solution  $S_1$  , neighborhood structures  $N_{k'}, k' = \{1, ..k'_{max}\}$  ;
2: Begin
3:  $k' = 1$  ;
4: repeat
5:   Find  $S_2$  the best neighbor of  $S_1$  using the neighborhood structure  $N_{k'}$ ;
6:   if  $F(S_2) < F(S_1)$  then
7:      $S_2 \leftarrow S_1$  ;
8:      $k' = 1$ ;
9:   else
10:     $k' = k' + 1$ ;
11:   end if
12: until ( $k' > k'_{max}$ ) ;
13: End ;

```

neighborhood structures:

- Neighborhood structure N_1 : Select a visited vertex from the solution and insert it to a new position in the same route.
- Neighborhood structure N_2 : Select a visited vertex from the solution and insert it to a new position in other routes.
- Neighborhood structure N_3 : Select two visited vertex from the same route and permute them.
- Neighborhood structure N_4 : Select two visited vertex from two different routes and permute them.

Generally speaking, the goal of using the VND procedure is to reduce the length of the tours and the value of the objective function. In fact, we need to make sure that only the minimum subset of T needed to cover W is kept in the tours. So, we propose to implement a move that consists of eliminating the redundant vertices from the solution and exchanging some visited vertices with another unvisited one to ensure the feasibility of the solution. From this step the mixed nature of the

VND appears. Thus, a mixed local search procedure is provided and referred to as mixed-VND. The different steps of the mixed-VND algorithm are summarized in Algorithm 2. In the next sections, we present the proposed approaches wherein the mixed-VND algorithm is integrated.

Algorithm 2 Steps of the mixed-VND algorithm

```

1: Input: initial solution  $S_0$ , neighborhood structure  $N_{K'}, k' = \{1, ..k'_{max}\}$ ;
2: Begin
3:  $S \leftarrow S_0$ ;
4: Let  $I$  the set of visited vertices in the solution  $S$ ;
5: Let  $R$  the set of unvisited vertices,  $R = V \setminus I$ 
6: Select a vertex  $i$  from  $I$ ,  $i \in I$ 
7: Let  $J_i$  a subset of  $R$  that keep the feasibility of the solution by changing vertex
   in  $I$  with vertex in  $J_i$ ;
8: for  $j \in J_i$  do
9:   Swap move between  $i$  and  $j$  to get  $S_1$ ;
10:   $S'_1 = VND(S_1)$ ;
11:  if  $F(S'_1) < F(S_1)$  then
12:     $S \leftarrow S'_1$ ;
13:    Update  $I$ ;
14:    Go to line 4;
15:  end if
16: end for
17: End;

```

3.2. The Genetic Algorithm

GA is an evolutionary algorithm developed by authors in [16]. Genetic algorithms prove their effectiveness for solving NP-hard problems. In this work we combine the GA with VND method to obtain hybrid approach called GA-VND. We attempt to improve the quality of the GA solution through successively using a set of neighborhood structures (mixed-VND). Before giving technical details of our hybrid approach, we briefly describe the basic concepts of GA. To apply the GA, we first need to find an appropriate genetic representation. The way the solutions are represented plays an important role in the performance of the GA. The GA appears attractive because of its population based nature. The initial population is defined by an initial set of random suitable solutions for the problem named chromosomes. In the genetic algorithm we begin first by initializing the population and then measuring the fitness function of each chromosome in the population. Then, a new population is created by applying different operators to the current population. The most well-known operators used are: i) **Selection** operator where we select two parents from a population according to their fitness, ii) **Crossover** operator applied with crossover probability where we cross over the parents to form the new offspring, and iii) **Mutation** operator where we mutate

the new offspring with a mutation probability. After applying those operators in that order ,a new offspring is produced, accepted and placed in a new population. We repeated these steps until the new population is completed. The new generated population is used again for a further run of GA until the stopping condition is satisfied. Finally, we return the best solution in the current population. The GA has a great flexibility cause its applying several operators to the solution. Figure 1 describes the flowchart of our GA-VND framework.

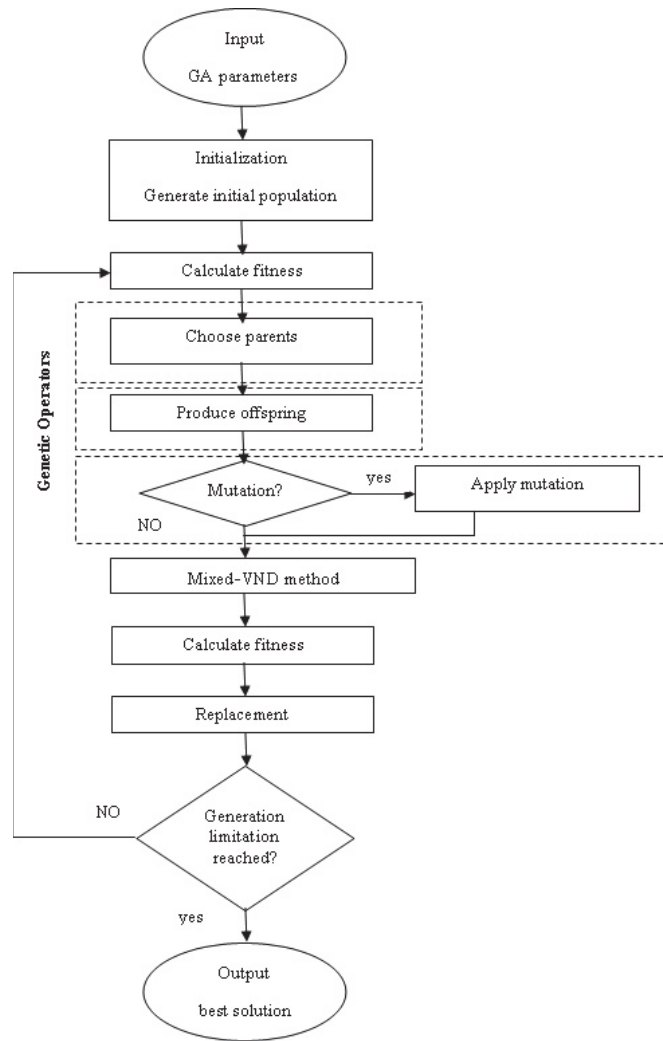


Figure 1: Flow Chart of solution of mm-CTP-p using Genetic Algorithm.

In the following we present our hybrid approach that incorporates the mixed-VND method into GA general scheme (Algorithm 3). Thereafter, we describe the

solution representation and discuss the genetic operators used therein.

Initial population and solution representation

The initial population is composed by a set of feasible solutions for the studied problem. In this work, an initial population was built by generating randomly a set of potential solutions for the mm-CTP-p. Each chromosome represents one individual, which is a possible solution to the problem and is presented by a vector. Let S_i be an individual from the population Pop , S_i is initially constructed by the random insertion of all customers that must be visited (T) in the vector and the random insertion of the remaining set of customers (unvisited customers of V) until all customers of W are covered a number of times equal to U_i .

Selection

This operator gives rise to a new generation by choosing good parents from the initial population that will be used in the next step to create new children. In our work, we applied the roulette wheel selection to select two parents to survive according to their probability distribution in order to create new offspring. The probability distribution of each individual is computed as follow:

$$P_i = 1 + (M - F(S_i))$$

where $M = \max(F(S_1), F(S_2), \dots, F(S_{psize}))$

After calculating the fitness $F(S_i)$ and the probability distribution P_i of each individual, we select two parents by repeating the following steps:

1. Calculate the cumulative probability P_c ; $P_c = \sum_{i=1}^{psize} P_i$
2. Generate a random number r in the range $[0; P_c]$;
3. Initialize t at P_0
4. For each P_i in Pop **if** $r < t$ **then**, $S \leftarrow S_i$ **otherwise**, $t = t + p_{i+1}$

Crossover

Replacement of a new child is done by taking information from two individuals from the population. More specifically, the new child inherits the characteristics from the selected parents. The crossover operator can be achieved in different ways. We consider a one point crossover by randomly choosing a crossing point cp from the first parent. The new child (S_{new}) is created by coping the first part (before cp) from the first parent, considered as the first part of this new solution. Then, we clean up the chromosome of the second parent, i.e., we remove the nodes taken from the first parent in order to construct the second part of the solution without redundant vertices. A sequence of nodes is obtained and used later for the creation of the second part of S_{new} . In fact, we select randomly a node from the updated sequence and insert it in S_{new} . We repeat this process until S_{new} becomes feasible. Figure 2 shows the reproduction of the new offspring using the crossover operator in the example of a solution of 3 vehicles with 15 vertices where only two vertices must be visited, $T = \{2,3\}$ and $cp = 3$. The chromosome of

parents contains a set of genes. A gene is an integer and represents the number of customers. For the first parent, Route1 is served by vehicle1 that visits the ordered list from the left, starting with customer 6, to the right, ending with the customer 10 before it goes back do the depot.

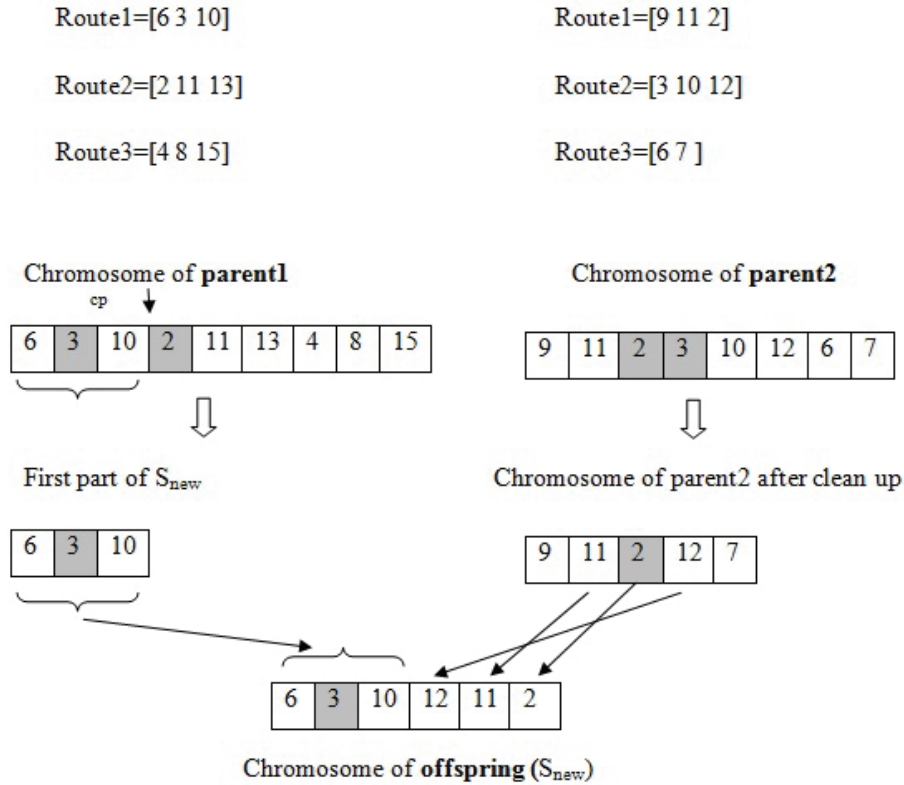


Figure 2: Example of crossover operator.

Mutation

The mutation operator is used to maintain the diversity of a population which enhance the solution so not to fail into local optima. There are many ways to mutate an individual. In our GA-VND approach we opted for a drop-insert operator. The considered operator consists of selecting randomly r vertices and removing them from the solution. Then, inserting randomly some vertices from the remaining set of vertices (vertices don't belong to the solution) until the covering constraint is respected. In this work, we do not mutate all the offspring resulted from the crossover operator. In fact, the mutation is performed according to the probability of mutation (P_m).

Mixed-VND method

This local search method is used to intensify the population since it is based on a descend method. More specifically, it consists of selecting a vertex from the mutated offspring and checking if the selected vertex can be removed from the solution without losing their feasibility. If it is the case, we remove this vertex from the solution and continue with other visited vertices. Otherwise, we try to swap the selected vertex with a set of unvisited vertices in order to find the best vertex and to insert it in the best (cheapest) position using different neighborhood structures. It is clear that this method can improve the offspring by eliminating redundant nodes, which optimize the solution.

Replacement

The replacement phase consists of removing an individual from the population and replacing it with the new obtained offspring. The key question here is which individual must be dropped from the population and replaced with the new one. In our approach we choose to replace it with the worst individual from the population. If the worst individual in the population is better than the new obtained offspring, we keep the best individual in the population without applying the replacement operator.

Algorithm 3 Hybrid GA-VND high level overview

```

1: Initialize:  $p_{size} = 10$  ,  $P_m = 0, 1$  ,  $nb_{gen} = 1000$ ; /*Global parameters*/
2: Generate Pop; /*First generation of individuals*/
3: Calculate the evaluation function  $F$  of each individual  $S_i$  in  $Pop$  ;
4: Initialize the best fitness  $F_{best}$ ;
5: Initialize the worst fitness  $F_{worst}$ ;
6: while the number of generation is not reached do
7:   SELECT two parents from  $Pop$  according to their probability distribution
8:   Apply CROSSOVER to the selected parents;
9:   Let  $S_{new}$  be the new obtained offspring;
10:  Apply MUTATION to  $S_{new}$  with probability  $P_m$ ;
11:  Apply mixed-VND to the mutated solution
12:  Let  $S_{new1}$  be the new obtained solution;
13:  if  $f(S_{new1}) < f(S_{worst})$  then
14:    Apply REPLACEMENT to Pop;
15:    Update the best fitness  $F_{best}$ ;
16:    Update the best fitness  $F_{worst}$ ;
17:  end if
18: end while;

```

4. THE GENERAL VARIABLE NEIGHBORHOOD SEARCH ALGORITHM

The well known Variable Neighborhood Search algorithm (VNS) was developed in [17]. The strategy of this framework is based on a systemic change of

neighborhood structures in order to achieve optimal solutions.

The VNS approach consists of three steps: The first one is the shaking phase that guarantees the diversification of the search to help to escape local optimal solutions. The second one is the local search procedure where different neighborhood structures are involved to intensify the search and improve the quality of the solution. The set of neighborhood structures used for this phase can be different from the set used in the first phase. If a Variable Neighborhood Descent method (VND) is used as the improvement procedure in the local search phase, a GVNS algorithm is obtained. In this work we develop a GVNS algorithm where only one neighborhood structure is considered in the shaking phase and four neighborhood structures are used in the local search phase. The last step is the neighborhood change move where the switch is made to the neighborhood that will be explored and to the new current solution. Algorithm 4 shows the framework of the GVNS algorithm.

Algorithm 4 GVNS algorithm

```

1: Initialise:  $iter_{max}, k_{max}$ 
2: Generate initial solution  $S$ 
3: while the maximum number of iteration is not met do
4:   while  $k < k_{max}$  do
5:     Apply SHAKING to  $S$  using the set of neighborhood structure  $N_k, k =$ 
        $\{1, \dots, k_{max}\}$ ;
6:     Let  $S'$  the new obtained solution;
7:     Apply mixed-VND to  $S'$  using the set of neighborhood structure  $N_{K'}, k' =$ 
        $\{1, \dots, k'_{max}\}$ ;
8:     Let  $S''$  the new improved solution;
9:     Apply CHANGE NEIGHBOURHOOD to  $S$  and  $S''$ ;
10:    Let  $S$  the new current solution;
11:   end while;
12: end while;

```

First we start by generating an initial feasible solution. To build the initial solution, we insert all the vertices of T at random routes. Then, we insert randomly vertices from V to cover a maximum number of vertices in $V \cup W$. To allow the diversification of this solution, a perturbation procedure was performed and a new solution S' was created.

The perturbation (shaking) phase is performed by the neighborhood structure N_5 that consists of selecting randomly l visited vertices and removing them from the solution. The removed vertices do not include any mandatory location, i.e., we remove only vertices that belong to $V \setminus T$. The new solution S' is constructed by inserting at random some unvisited vertices from V until the covering constraints are respected. To better understand the perturbation procedure, an illustration of the neighborhood structure N_5 is provided on the following simple instance:

The set of vertices include one vertex that must be visited $T = \{v_1\}$; fifteen vertices that may be visited and used to cover five vertices where $V = \{v_2, \dots, v_{15}\}$

Table 1: The coverage possibility

Vertices	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	nb_k
w_1	×				×										2
w_2		×		×						×				×	4
w_3				×	×	×				×	×	×			6
w_4									×						1
w_5	×		×						×						3

Table 2: The different solutions to the example instance in SHAKING phase.

Solutions	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_{11}	v_{12}	v_{13}	v_{14}	v_{15}	R
S	×	×	×	×	×					×	×					$\{v_5, v_7, v_8, v_9, v_{13}, v_{14}, v_{15}\}$
S_1	×	×									×	×				$\{v_5, v_7, v_8, v_9, v_{13}, v_{14}, v_{15}, v_3, v_4, v_6, v_{10}\}$
S_2	×	×		×							×	×				$\{v_7, v_8, v_9, v_{13}, v_{14}, v_{15}, v_3, v_4, v_6, v_{10}\}$
S'	×	×		×					×	×	×					$\{v_7, v_8, v_9, v_{13}, v_{14}, v_{15}, v_3, v_4, v_6\}$

and $W = \{w_1, w_2, w_3, w_4, w_5\}$. Tables 1 and 2 show respectively the possibility of coverage of each vertex of W and the visited vertices of different solutions during the perturbation phase. For example w_1 can be covered by two vertex $\{v_2, v_6\}$. In this example, four vertices ($l = 4$) were randomly removed from the initial solution S using the neighborhood structure N_5 . An infeasible solution S_1 was resulted and the remaining set of vertices R containing the unvisited vertices was updated. The vertex v_5 was selected randomly from R and inserted in the solution. After this insertion move the obtained solution S_2 was still infeasible since the coverage demand was not completely fulfilled (see Table 3). This insertion process was repeated until the total demand was served and a feasible solution S' was met (Fig. 3). Table 3 summarizes the coverage demand of each vertex of W during the SHAKING procedure. Let nb_{w_i} be the number of time a vertex w_i of W is covered in the solution. A solution is considered feasible if nb_{w_i} is better than U_i in other word $nb_{w_i}/U_i \geq 1$. After a shaking phase has been performed, the local search procedure explores gradually the neighborhood of a the solution S' in order to find the neighbor solution S'' that improves the objective function. We consider the mixed-VND algorithm described in the previous sections as the local search phase in our GVNS algorithm.

The change step is applied according to the fitness of the solution S and the

Table 3: The coverage demand of different solutions.

Vertices	w_1	w_2	w_3	w_4	w_5
S	2/1	2/2	3/3	1/1	3/2
S_1	1/1	1/2	2/3	0/1	1/2
S_2	1/1	2/2	3/3	0/1	1/2
S'	1/1	2/2	3/3	1/1	2/2

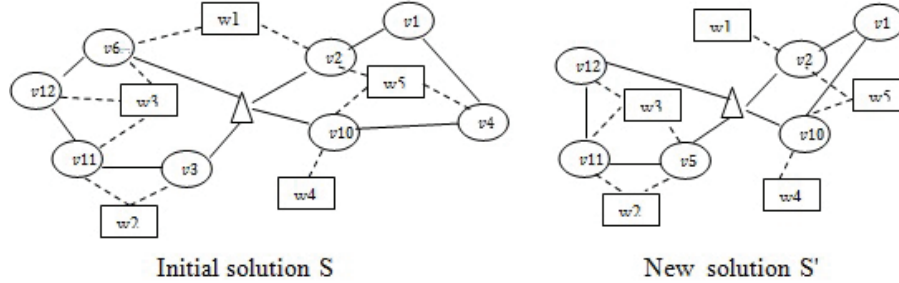


Figure 3: An example of a move with neighborhood N_5 .

obtained solution S'' . If $F(S'') < F(S)$ then, we change the current solution ($S \leftarrow S''$) and continue the search with the first neighborhood structure ($k \leftarrow 1$), otherwise, we keep the current solution S and continue with the next neighborhood structure ($k \leftarrow k + 1$).

5. EXPERIMENTAL RESULTS

Our proposed algorithms are coded in C++ programming language and run on a computer with an Intel Core i 5-4200U and 2.3 Ghz processor and 6 GB memory.

The mm-CTP-p is NP-hard and the exact methods that have been provided in [13] to deal with this problem give optimal solutions only for 69 out of 80 instances with 50 vertices of V . For the remaining set of the instances ($|V| > 50$) CPLEX solver needs more computational time, since the running time of the proposed branch- and-cut algorithm is limited to two hours for each instance [13]. For this reason, we opted for approximated algorithms to address the problem.

5.1. Test instances and parameter settings

We tested our algorithms on benchmark instances generated in [10]. There are four sets of instances with 100 vertices and two sets with 200 vertices. The KroA100, KroB100, KroC100 and KroD100 represent the small size instances whereas KroA200 and KroB200 represent the large size instances. The instances are labeled X-T-n-W-p, where X is the name of the instance, T is the set of vertices that must be visited, n is the number of vertices in V and p represents the maximum number of vertices in each route. For further information about the considered instances see [13].

The choice of the parameter values of our GA-VND and GVNS algorithms have an impact on the quality of the solution. Table 4 summarizes the different parameters considered in our proposed algorithms.

5.2. Comparative study

In this section, a computational study is carried out to compare our proposed approaches with best known solutions. Tables 7 and 8 report the numerical

Table 4: Parameters of our GA-VND and GVNS algorithms.

Parameter	Description	Value
p_{size}	Size of population	10
nb_{gen}	Number of generation	1000
P_m	The probability of mutation operator	0.1
r	Number of dropped nodes in the mutation operator	Random
N_K	The set of neighborhood structure in the SHAKING phase	–
$N_{K'}$	The set of neighborhood structure in the mixed-VND	–
k_{max}	Maximum number of neighborhood structure in the N_K	4
k'_{max}	Maximum number of neighborhood structure in the $N_{K'}$	1
$iter_{max}$	Maximum number of iteration in the GVNS	1000
l	Number of removed vertices in the SHAKING phase	Random

Table 5: Stability of our GVNS and GA-VND algorithms.

Methods	Same cost	GVNS better	GA-VND better
GVNS	2	4	0
GA-VND	2	0	0

results of our algorithms and a comparison with existing method in the literature (GRASP-ELS and GA-VLG). Column Data is the name of instance. Column Best presents the best cost of the solution over 10 runs. Column Avg. present the average cost of the solution over 10 runs. The column Time is the total running time in seconds of 10 runs . Column σ^2 show the variance of solution costs over 10 runs. The columns GRASP-ELS and GA-VLG show the results of [13]. The columns GVNS and GA-VND report the results of our proposed approaches. The columns Gap_1 and Gap_2 report the standard deviation from the best and the average solution between the method in the literature and our GVNS and GA-VND respectively. To the best of our knowledge, the work in [13] is the only work in the literature that deals with a generalized variant of the m-CTP multiple coverage.

Table 5 shows that our GVNS algorithm performs better than our GA-VND algorithm. Indeed, it finds four better solutions in term of best and average cost in the instances A2-1-100-100-*-250 where $p = \{4,5,6\}$. It also provides two solutions with better variance where the two algorithm provides the same best cost (A2-1-100-100-8-250, B1-1-50-50-5-250). The GA-VND provides only two solutions with better variance where it provides the same best cost than GVNS (A2-20-100-100-6-250, A2-20-100-100-8-250). We investigate the stability of the proposed algorithms by calculating the variance of solution costs of each instance over 10 runs. Then, we compare the results with GRASP-ELS and GA-VLG meta-heuristics. Table 6 summarizes this result where the column headings are as follows: Same Cost column presents the number of problem instances that the algorithm (GVNS , GA-VND , GRASP-ELS, GA-VLG) has better variance than the others while

they provide the same best cost. A comparison between our GVNS and GRASP-ELS is presented where GVNS better and GRASP- ELS better columns are the same as Same Cost column, GVNS provides better solutions than GRASP-ELS does or vice versa, respectively. Also, in the same way, we compare the variance of our GVNS algorithm with the GA-VLG algorithm where each of them has better best cost than the other does (Columns GVNS better and GA -VLG better). Table 6 shows also a comparison between our GA-VND algorithm and the two meta-heuristics (GRASP-ELS, GA-VLG) in the same way as for GVNS.

Table 6 shows that the GVNS method has 16 times better (smaller) variance than GRASP-ELS, 6 times better variance than GA-VLG, and 5 times better variance than both of them when these three algorithms provide the same best cost. However, results shows that the GA-VND method has 5 times better variance than GRASP-ELS, 5 times better variance than GA-VLG, and 4 times better variance than both of them. It is clear from this table that, when the same best cost is reported, our algorithms are much more stable. Indeed, better solution cost and variance are better than GRASP-ELS in six and four instances respectively. But, the GVNS achieves smaller variance than the GRASP-ELS and GA-VLG in only one instance where each of them has better solution cost than the GVNS. The GA-VND achieves smaller variance than the GRASP-ELS and GA-VLG in two and three instances respectively where each of them has better solution cost than the GA-VND. Our approaches provide the majority of the best known solutions in a reasonable computational time compared with GRASP-ELS algorithm whereas the GA-VLG seems to be faster than all other methods (see Fig. 4)

Table 7 shows a comparison of the best cost solutions between the different algorithms. We indicate the better cost founded by our proposed algorithms in bold. Table 8 details the variance of solution costs of each instance over 10 runs.

Table 7: Comparison of best cost solution on mm-CTP-p

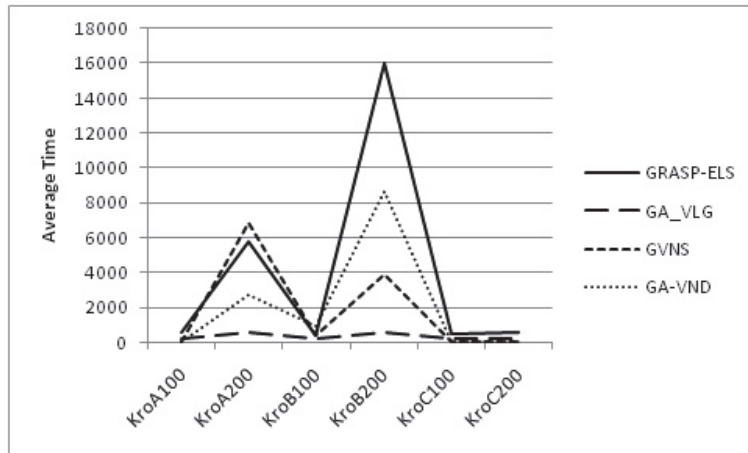
Data	GRASP-ELS			GA-VLG			GVNS	GA-VND
	Best	Gap ₁	Gap ₂	Best	Gap ₁	Gap ₂	Best	Best
A1-1-25-75-4-250	17774.00	0.00	0.00	17774.00	0.00	0.00	17774.00	17774.00
A1-1-25-75-5-250	15793.00	0.00	0.00	15793.00	0.00	0.00	15793.00	15793.00
A1-1-25-75-6-250	14628.00	0.00	0.00	14628.00	0.00	0.00	14628.00	14628.00
A1-1-25-75-8-250	12590.00	0.00	0.00	12590.00	0.00	0.00	12590.00	12590.00
A1-1-50-50-4-250	21473.00	0.00	0.00	21473.00	0.00	0.00	21473.00	21473.00
A1-1-50-50-5-250	18680.00	0.00	0.00	18680.00	0.00	0.00	18680.00	18680.00
A1-1-50-50-6-250	17481.00	0.00	0.00	17481.00	0.00	0.00	17481.00	17481.00
A1-1-50-50-8-250	14380.00	0.00	0.00	14380.00	0.00	0.00	14380.00	14380.00
A1-10-50-50-4-250	25340.00	0.00	0.00	25340.00	0.00	0.00	25340.00	25340.00
A1-10-50-50-5-250	21712.00	0.00	0.00	21712.00	0.00	0.00	21712.00	21712.00
A1-10-50-50-6-250	20125.00	0.00	0.00	20125.00	0.00	0.00	20125.00	20125.00
A1-10-50-50-8-250	17603.00	0.00	0.00	17603.00	0.00	0.00	17603.00	17603.00
A1-5-25-75-4-250	13082.00	0.00	0.00	13082.00	0.00	0.00	13082.00	13082.00
A1-5-25-75-5-250	11969.00	0.00	0.00	11969.00	0.00	0.00	11969.00	11969.00
A1-5-25-75-6-250	11746.00	0.00	0.00	11746.00	0.00	0.00	11746.00	11746.00
A1-5-25-75-8-250	9081.00	0.00	0.00	9081.00	0.00	0.00	9081.00	9081.00
A2-1-100-100-4-250	25051.00	0.10	0.00	25026.00	0.00	-0.10	25026.00	25051.00
A2-1-100-100-5-250	21626.00	0.00	-0.20	21626.00	0.00	-0.20	21626.00	21670.00
A2-1-100-100-6-250	19119.00	0.06	0.00	19108.00	0.00	-0.06	19108.00	19119.00
A2-1-100-100-8-250	16226.00	0.10	0.10	16209.00	0.00	0.00	16209.00	16209.00
A2-1-50-150-4-250	23601.00	0.00	0.00	23601.00	0.00	0.00	23601.00	23601.00
A2-1-50-150-5-250	20439.00	0.00	0.00	20439.00	0.00	0.00	20439.00	20439.00

Table 7 – (continued)

Data	GRASP-ELS			GA-VLG			GVNS	GA-VND
	Best	Gap ₁	Gap ₂	Best	Gap ₁	Gap ₂	Best	Best
A2-1-50-150-6-250	18410.00	0.00	0.00	18410.00	0.00	0.00	18410.00	18410.00
A2-1-50-150-8-250	15565.00	0.41	0.41	15502.00	0.00	0.00	15502.00	15502.00
A2-10-50-150-4-250	25702.00	0.00	0.00	25702.00	0.00	0.00	25702.00	25702.00
A2-10-50-150-5-250	21503.00	0.00	0.00	21503.00	0.00	0.00	21503.00	21503.00
A2-10-50-150-6-250	20250.00	0.00	0.00	20250.00	0.00	0.00	20250.00	20250.00
A2-10-50-150-8-250	16676.00	0.00	0.00	16676.00	0.00	0.00	16676.00	16676.00
A2-20-100-100-4-250	38074.00	0.00	0.00	38074.00	0.00	0.00	38074.00	38074.00
A2-20-100-100-5-250	32646.00	0.19	0.19	32583.00	0.00	0.00	32583.00	32583.00
A2-20-100-100-6-250	28490.00	0.00	0.00	28490.00	0.00	0.00	28490.00	28490.00
A2-20-100-100-8-250	24615.00	0.09	0.09	24593.00	0.00	0.00	24593.00	24593.00
B1-1-25-75-4-250	17417.00	0.00	0.00	17417.00	0.00	0.00	17417.00	17417.00
B1-1-25-75-5-250	15891.00	0.00	0.00	15891.00	0.00	0.00	15891.00	15891.00
B1-1-25-75-6-250	14260.00	0.00	0.00	14260.00	0.00	0.00	14260.00	14260.00
B1-1-25-75-8-250	11538.00	0.00	0.00	11538.00	0.00	0.00	11538.00	11538.00
B1-1-50-50-4-250	19966.00	0.00	0.00	19966.00	0.00	0.00	19966.00	19966.00
B1-1-50-50-5-250	17113.00	0.00	0.00	17113.00	0.00	0.00	17113.00	17113.00
B1-1-50-50-6-250	15989.00	0.00	0.00	15989.00	0.00	0.00	15989.00	15989.00
B1-1-50-50-8-250	14027.00	0.00	0.00	14027.00	0.00	0.00	14027.00	14027.00
B1-10-50-50-4-250	20075.00	0.00	0.00	20075.00	0.00	0.00	20075.00	20075.00
B1-10-50-50-5-250	17986.00	0.00	0.00	17986.00	0.00	0.00	17986.00	17986.00
B1-10-50-50-6-250	15924.00	0.00	0.00	15924.00	0.00	0.00	15924.00	15924.00
B1-10-50-50-8-250	13672.00	0.00	0.00	13672.00	0.00	0.00	13672.00	13672.00
B1-5-25-75-4-250	17079.00	0.00	0.00	17079.00	0.00	0.00	17079.00	17079.00
B1-5-25-75-5-250	15110.00	0.00	0.00	15110.00	0.00	0.00	15110.00	15110.00
B1-5-25-75-6-250	14707.00	0.00	0.00	14707.00	0.00	0.00	14707.00	14707.00
B1-5-25-75-8-250	11319.00	0.00	0.00	11319.00	0.00	0.00	11319.00	11319.00
B2-1-100-100-4-250	40974.00	0.00	0.00	40974.00	0.00	0.00	40974.00	40974.00
B2-1-100-100-5-250	34848.00	0.00	0.00	34848.00	0.00	0.00	34848.00	34848.00
B2-1-100-100-6-250	30829.00	-0.06	-0.06	30849.00	0.00	0.00	30849.00	30849.00
B2-1-100-100-8-250	25804.00	0.00	0.00	25804.00	0.00	0.00	25804.00	25804.00
B2-1-50-150-4-250	23288.00	0.00	0.00	23288.00	0.00	0.00	23288.00	23288.00
B2-1-50-150-5-250	20039.00	0.00	0.00	20039.00	0.00	0.00	20039.00	20039.00
B2-1-50-150-6-250	18046.00	0.00	0.00	18046.00	0.00	0.00	18046.00	18046.00
B2-1-50-150-8-250	15668.00	0.00	0.00	15668.00	0.00	0.00	15668.00	15668.00
B2-10-50-150-4-250	25967.00	0.00	0.00	25967.00	0.00	0.00	25967.00	25967.00
B2-10-50-150-5-250	22359.00	0.00	0.00	22359.00	0.00	0.00	22359.00	22359.00
B2-10-50-150-6-250	19792.00	0.00	0.00	19792.00	0.00	0.00	19792.00	19792.00
B2-10-50-150-8-250	17106.00	0.00	0.00	17106.00	0.00	0.00	17106.00	17106.00
B2-20-100-100-4-250	53590.00	0.00	0.00	53590.00	0.00	0.00	53590.00	53590.00
B2-20-100-100-5-250	45209.00	0.00	0.00	45209.00	0.00	0.00	45209.00	45209.00
B2-20-100-100-6-250	39184.00	0.00	0.00	39184.00	0.00	0.00	39184.00	39184.00
B2-20-100-100-8-250	32513.00	0.00	0.00	32512.00	0.00	0.00	32513.00	32513.00
C1-1-25-75-4-250	13012.00	0.00	0.00	13012.00	0.00	0.00	13012.00	13012.00
C1-1-25-75-5-250	11666.00	0.00	0.00	11666.00	0.00	0.00	11666.00	11666.00
C1-1-25-75-6-250	9820.00	0.00	0.00	9820.00	0.00	0.00	9820.00	9820.00
C1-1-25-75-8-250	9818.00	0.00	0.00	9818.00	0.00	0.00	9818.00	9818.00
C1-1-50-50-4-250	20294.00	0.00	0.00	20294.00	0.00	0.00	20294.00	20294.00
C1-1-50-50-5-250	17378.00	0.00	0.00	17378.00	0.00	0.00	17378.00	17378.00
C1-1-50-50-6-250	16365.00	0.00	0.00	16365.00	0.00	0.00	16365.00	16365.00
C1-1-50-50-8-250	13900.00	0.00	0.00	13900.00	0.00	0.00	13900.00	13900.00
C1-10-50-50-4-250	26931.00	0.00	0.00	26931.00	0.00	0.00	26931.00	26931.00
C1-10-50-50-5-250	23544.00	0.00	0.00	23544.00	0.00	0.00	23544.00	23544.00
C1-10-50-50-6-250	20818.00	0.00	0.00	20818.00	0.00	0.00	20818.00	20818.00
C1-10-50-50-8-250	18154.00	0.00	0.00	18154.00	0.00	0.00	18154.00	18154.00
C1-5-25-75-4-250	13738.00	0.00	0.00	13738.00	0.00	0.00	13738.00	13738.00
C1-5-25-75-5-250	13575.00	0.00	0.00	13575.00	0.00	0.00	13575.00	13575.00
C1-5-25-75-6-250	10826.00	0.00	0.00	10826.00	0.00	0.00	10826.00	10826.00
C1-5-25-75-8-250	10556.00	0.00	-0.09	10556.00	0.00	-0.09	10556.00	10566.00
D1-1-25-75-4-250	18127.00	0.00	0.00	18127.00	0.00	0.00	18127.00	18127.00
D1-1-25-75-5-250	15972.00	0.00	0.00	15972.00	0.00	0.00	15972.00	15972.00

Table 7 – (continued)

Data	GRASP-ELS			GA-VLG			GVNS	GA-VND
	Best	Gap ₁	Gap ₂	Best	Gap ₁	Gap ₂	Best	Best
D1-1-25-75-6-250	14532.00	0.00	0.00	14532.00	0.00	0.00	14532.00	14532.00
D1-1-25-75-8-250	12700.00	0.00	0.00	12700.00	0.00	0.00	12700.00	12700.00
D1-1-50-50-4-250	23275.00	0.00	0.00	23275.00	0.00	0.00	23275.00	23275.00
D1-1-50-50-5-250	20402.00	0.00	0.00	20402.00	0.00	0.00	20402.00	20402.00
D1-1-50-50-6-250	18072.00	0.00	0.00	18072.00	0.00	0.00	18072.00	18072.00
D1-1-50-50-8-250	14930.00	0.00	0.00	14930.00	0.00	0.00	14930.00	14930.00
D1-10-50-50-4-250	30390.00	0.00	0.00	30390.00	0.00	0.00	30390.00	30390.00
D1-10-50-50-5-250	26284.00	0.00	0.00	26284.00	0.00	0.00	26284.00	26284.00
D1-10-50-50-6-250	23646.00	0.00	0.00	23646.00	0.00	0.00	23646.00	23646.00
D1-10-50-50-8-250	19986.00	0.00	0.00	19986.00	0.00	0.00	19986.00	19986.00
D1-5-25-75-4-250	18464.00	0.00	0.00	18464.00	0.00	0.00	18464.00	18464.00
D1-5-25-75-5-250	15767.00	0.00	0.00	15767.00	0.00	0.00	15767.00	15767.00
D1-5-25-75-6-250	14851.00	0.00	0.00	14851.00	0.00	0.00	14851.00	14851.00
D1-5-25-75-8-250	12705.00	0.00	0.00	12705.00	0.00	0.00	12705.00	12705.00

Figure 4: Average of Time for instances *KroA100*, *KroB100*, *KroC100*, *KroD100*, *KroA200* and *KroB200*.

6. CONCLUSION

In this paper, we deal with the generalized variant of the multi-vehicle covering tour problem (m-CTP) called the multi-vehicle multi-covering Tour Problem (mm-CTP). This variant has gained increased importance in recent years since it integrates covering issue into the routing problem. In order to solve this problem, we have developed two meta-heuristics, where the first one is a population-based approach (GA-VND), and the second is a solution-based approach (GVNS). In both meta-heuristics we integrate a mixed-VND algorithm, known as an effective local search method, in order to intensify the search process. The experiment results show that the developed meta-heuristics give good solutions in a reasonable execution time. The algorithms generate high-quality solutions for the majority of instances and seem to be more stable than GRASP-ELS and GA-VLG algorithms.

Table 6: Stability of the algorithms

Methods		Criteria							
		GVNS				GA-VND			
		Same cost		GVNS/GRASP-ELS	GVNS/GA-VLG	GA-VND/GRASP-ELS	GA-VND/GA-VLG		
		GVNS better	GRASP-ELS better	GVNS better	GA-VLG better	GA-VND better	GRASP-ELS better	GA-VND better	GA-VLG better
GVNS	GRASP-ELS	16							
	GA-VLG	6	6	1	0	1	-	-	-
	GRASP-ELS and GA-VLG	5							
GRASP-ELS	GVNS	0							
	GA-VLG	4	0	0	-	-	-	-	-
	GVNS and GA-VLG	0							
GA-VLG	GVNS	1							
	GRASP-ELS	13	-	-	0	0	-	-	-
	GVNS and GRASP-ELS	1							
GA-VND	GRASP-ELS	5							
	GA-VLG	5	-	-	-	-	4	2	0
	GRASP-ELS and GA-VLG	4							3
GRASP-ELS	GA-VND	0							
	GA-VLG	4	-	-	-	-	0	0	-
	GA-VND and GA-VLG	0							
GA-VLG	GA-VND	1							0
	GRASP-ELS	13	-	-	-	-	-	-	0
	GA-VND and GRASP-ELS	1							

Table 8: Comparison of the average solution on mm-CTP-p

Data	GRASP-ELS					GA-VLG					GVNS			GA-VND		
	Avg.	σ^2	Time	Gap ₁	Gap ₂	Avg.	σ^2	Time	Gap ₁	Gap ₂	Avg.	Time	σ^2	Avg.	Time	σ^2
A1-1-25-75-4-250	1774.00	0.00	56.77	0.00	0.00	1774.00	0.00	178.34	0.00	0.00	1774.00	0.86	0.00	1774.00	1.50	0.00
A1-1-25-75-5-250	15793.00	0.00	61.86	0.00	0.00	15793.00	0.00	177.52	0.00	0.00	15793.00	1.28	0.00	15793.00	2.41	0.00
A1-1-25-75-6-250	14628.00	0.00	59.14	0.00	0.00	14628.00	0.00	178.32	0.00	0.00	14628.00	1.14	0.00	14628.00	1.28	0.00
A1-1-25-75-8-250	12590.00	0.00	60.39	0.00	0.00	12590.00	0.00	179.16	0.00	0.00	12590.00	1.46	0.00	12590.00	2.83	0.00
A1-1-50-50-4-250	21473.00	0.00	860.54	0.00	0.00	21473.00	0.00	283.15	0.00	0.00	21473.00	5.73	0.00	21473.00	15.25	0.00
A1-1-50-50-5-250	18680.00	0.00	898.80	0.00	0.00	18680.00	0.00	287.94	0.00	0.00	18680.00	4.09	0.00	18680.00	4.97	0.00
A1-1-50-50-6-250	17481.00	0.00	944.39	0.00	0.00	17481.00	0.00	284.99	0.00	0.00	17481.00	46.37	0.00	17481.00	69.42	0.00
A1-1-50-50-8-250	14380.00	0.00	965.42	0.00	0.00	14380.00	0.00	278.43	0.00	0.00	14380.00	22.76	0.00	14380.00	26.62	0.00
A1-10-50-50-4-250	25340.00	0.00	1165.99	0.00	0.00	25340.00	0.00	298.60	0.00	0.00	25340.00	4.71	0.00	25340.00	13.54	0.00
A1-10-50-50-5-250	21712.00	0.00	1133.15	0.00	0.00	21712.00	0.00	309.51	0.00	0.00	21712.00	17.39	0.00	21712.00	37.19	0.00
A1-10-50-50-6-250	20125.00	0.00	1144.12	0.00	0.00	20125.00	0.00	300.33	0.00	0.00	20125.00	11.82	0.00	20125.00	15.26	0.00
A1-10-50-50-8-250	17603.00	0.00	1253.44	0.00	0.00	17603.00	0.00	308.99	0.00	0.00	17603.00	5.97	0.00	17603.00	12.21	0.00
A1-5-25-75-4-250	13082.00	0.00	20.83	0.00	0.00	13082.00	0.00	159.75	0.00	0.00	13082.00	1.91	0.00	13082.00	0.37	0.00
A1-5-25-75-5-250	11969.00	0.00	21.74	0.00	0.00	11969.00	0.00	159.81	0.00	0.00	11969.00	2.51	0.00	11969.00	0.43	0.00
A1-5-25-75-6-250	11746.00	0.00	21.10	0.00	0.00	11746.00	0.00	162.48	0.00	0.00	11746.00	2.65	0.00	11746.00	0.48	0.00
A1-5-25-75-8-250	9081.00	0.00	21.46	0.00	0.00	9081.00	0.00	155.13	0.00	0.00	9081.00	3.48	0.00	9081.00	0.50	0.00
A2-1-100-100-4-250	25058.20	60.96	3656.09	0.13	0.03	25033.60	134.84	538.48	0.03	-0.07	25026.00	14205.36	0.00	25051.40	2688.03	0.49
A2-1-100-100-5-250	21677.30	292.41	4140.76	0.09	0.03	21669.10	629.89	717.98	0.06	0.00	21656.80	19119.83	20.16	21670.00	3626.77	0.00
A2-1-100-100-6-250	19180.20	7823.96	4026.62	0.38	0.32	19108.00	0.00	565.29	0.00	-0.06	19108.00	4062.64	0.00	19119.00	2116.90	0.00
A2-1-100-100-8-250	16241.00	235.80	4051.89	0.08	-0.21	16266.40	3803.84	564.42	0.24	-0.06	16228.00	21948.22	7.21	16275.50	7864.36	43.53
A2-1-50-150-4-250	23613.60	635.04	798.37	0.05	0.05	23601.00	0.00	533.38	0.00	0.00	23601.00	59.76	0.00	23601.00	42.64	0.00
A2-1-50-150-5-250	20443.20	158.76	835.08	0.02	0.02	20483.40	3591.84	617.65	0.22	0.22	20439.00	269.39	0.00	20439.00	249.79	0.00
A2-1-50-150-6-250	18410.00	0.00	829.21	0.00	0.00	18410.00	0.00	493.03	0.00	0.00	18410.00	91.37	0.00	18410.00	74.82	0.00
A2-1-50-150-8-250	15593.30	145.41	768.58	0.59	0.59	15502.00	0.00	371.09	0.00	0.00	15502.00	7.39	0.00	15502.00	6.95	0.00
A2-10-50-150-4-250	25712.40	432.64	1072.54	0.04	0.04	25702.00	0.00	380.91	0.00	0.00	25702.00	7.02	0.00	25702.00	6.10	0.00
A2-10-50-150-5-250	21503.00	0.00	1046.22	0.00	0.00	21503.00	0.00	369.09	0.00	0.00	21503.00	9.55	0.00	21503.00	8.68	0.00
A2-10-50-150-6-250	20250.00	0.00	1126.13	0.00	0.00	20250.00	0.00	353.10	0.00	0.00	20250.00	13.22	0.00	20250.00	6.34	0.00
A2-10-50-150-8-250	16676.00	0.00	1091.21	0.00	0.00	16676.00	0.00	354.46	0.00	0.00	16676.00	9.14	0.00	16676.00	5.99	0.00
A2-20-100-100-4-250	38104.40	316.04	16115.14	0.08	0.08	38078.60	84.64	704.69	0.01	0.01	38074.00	117.16	0.00	38074.00	292.39	0.00
A2-20-100-100-5-250	32680.90	872.09	16736.41	0.30	0.30	32634.20	5179.16	825.72	0.16	0.16	32583.00	2801.43	0.00	32583.00	8435.40	0.00
A2-20-100-100-6-250	28576.00	3811.60	18798.71	0.03	0.30	28490.00	0.00	683.06	-0.28	0.00	28568.60	25043.30	64.18	28490.00	4410.48	0.00
A2-20-100-100-8-250	24652.90	555.49	16746.77	0.21	0.23	24605.10	351.09	901.37	0.02	0.04	24600.50	21527.09	11.46	24595.50	13049.48	7.50

REFERENCES

- [1] Labbé, M. and Laporte, G., “Maximizing user convenience and postal service efficiency in post box location”, *Belgian journal of operations research, statistics and computer science*, 26 (1986) 21—35.
- [2] Simms, J.C., “Fixed and mobile facilities in dairy practice”, *Veterinary Clinics North Am. Food Animal Practice*, 5 (1989) 591—601.
- [3] Hodgson, M.J. , Laporte, G. and Semet, F., “A covering tour model for planning mobile health care facilities in suhum district gana”, *Journal of Regional Science*, 38 (1998) 621—638.
- [4] Doerner, K.F. , Hartl and R.-F., “Health care logistics, emergency preparedness and disaster relief: New challenges for routing problems with a focus on the austrian situation”, *The Vehicle Routing Problem: Latest Advances and New Challenges*, Springer, Boston, MA (2008) 527—550.
- [5] Current, J., “Multiobjective Design of Transportation Networks”, *PhD dissertation. Johns Hopkins University* (1981).
- [6] Gendreau, M. , Laporte, G. and Semet, F., “The covering tour problem”, *Operations Research*, 45 (1997) 568—576.
- [7] Baldacci, R. , Hadjiconstantinou, E. and Mingozzi, A., “An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation”, *Operations Research*, 52 (2004) 723—738.
- [8] Hachicha, M. , Hodgson, M.J. , Laporte, G. and Semet, F., “Heuristics for the multi-vehicle covering tour problem”, *Computers and Operations Research*, 27 (2000) 29—42.
- [9] Jozefowicz, N., “A branch-and-price algorithm for the multi-vehicle covering tour problem”, *Networks*, 64 (3) (2014) 160—168.
- [10] Hà, M.H. , Bostel, N. , Langevin, A. and Rousseau, L.-M., “An exact algorithm and a metaheuristic for the generalized vehicle routing problem with flexible fleet size”, *Computers and Operations Research*, 43 (2014) 9—19.
- [11] Kammoun, M. , Derbel, H. , Ratli, M. and Jarboui, B., “An integration of mixed VND and VNS: the case of the multivehicle covering tour problem”, *International Transactions in Operational Research*, 23 (3) (2017) 663—379.
- [12] Kammoun, M. , Derbel, H. , Ratli, M. and Jarboui, B., “ A variable neighborhood search for solving the multi-vehicle covering tour problem.”, *Electronics Notes in Discrete Mathematics*, 47 (2015) 285—292.
- [13] Pham, T.A., Hà, M.H. and Nguyen, X.H., “Solving the multi-vehicle multi-covering tour problem”, *Computers and Operations Research*, 88 (2017) 258—278.
- [14] Golden, B., NajiAzimi, Z., Raghavan, S., Salari, M. and Toth, P., “The generalized covering salesman problem”, *INFORMS Journal on Computing*, 24 (2012) 534—553.
- [15] Vidal, T., Crainic, T. G., Gendreau, M. and Prins, C., “A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows”, *Computers and Operations Research*, 40 (1) (2013) 475—489.
- [16] Holland, J., *Adaptation in natural and artificial systems*, University of Michigan Press, Ann Arbor, MI., 1975.
- [17] Mladenovic, N. and Hansen, P., “Variable neighborhood search”, *Computers and Operations Research*, 24 (11) (1997) 1097—1100.