International Journal of Computational Intelligence Systems
Vol. **14(1)**, 2021, *pp.* 1087–1107
DOI: https://doi.org/10.2991/ijcis.d.210226.001; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Research Article

Applying Metaheuristic for Time-Dependent Traveling Salesman Problem in Postdisaster

Ha-Bang Ban*

School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi, Vietnam

ARTICLE INFO

Article History
Received 09 Oct 2020
Accepted 07 Feb 2021

Keywords
TDTSP
TDTSP-PD
MA
GA
GRASP
VNS

ABSTRACT

The Time-Dependent Traveling Salesman Problem (TDTSP) is a generalization of the Traveling Salesman Problem (TSP) and Traveling Repairman Problem (TRP). In the TSP and TRP, the travel time to travel is assumed to be constant. However, in practice, the travel times vary according to several factors that naturally depend on the time of day. Therefore, the TDTSP is closer to several real practical situations than the TSP. In this paper, we introduce a new variant of the TDTSP, that is, the Time-Dependent Traveling Salesman Problem in Postdisaster (TDTSP-PD). In the problem, the travel costs need to be added debris removal times after a disaster occurs. To solve the TDTSP-PD, we present an effective population-based algorithm that combines the diversification power of the Genetic Algorithm (GA) and the intensification strength of Local Search (LS). Therefore, our metaheuristic algorithm maintains a balance between diversification and intensification. The results of the experimental simulation are compared with the well-known and successful metaheuristic algorithms. These results show that the proposed algorithm reaches better solutions in many cases.

1. INTRODUCTION

1.1. Definition

The problem is a generalization case of the Time-Dependent Traveling Salesman Problem (TDTSP), and at least as hard as the TDTSP. Therefore, it is also NP-hard problem. After that, we define the problem as follows:

Consider a complete graph $K_n = (V, E)$, a depot, $v_1 \in V$, a starting time $\tau \in R^+$, and a travel time function $f : E \times R^+ \to R^+$ in which $f\left(v_i, v_j, t\right)$ is the travel time from v_i to v_j when leaving v_i at time t in normal conditions. Moreover, there are some blocked edges destroyed by a disaster while the others do not affect by it. Let $r\left(v_i, v_j, t\right)$ be the time required to remove debris on edge $\left(v_i, v_j\right)$ at time t if edge $\left(v_i, v_j\right)$ is affected by disaster. Otherwise, this time is set to zero. Traveling from vertex v_i to v_j on edge $\left(v_i, v_j\right)$ takes $f\left(v_i, v_j, t\right) + r\left(v_i, v_j, t\right)$ times. Suppose that $T = \left(v_1, \ldots, v_k, \ldots, v_n\right)$ is a tour in K_n. The time of arrival at a vertex $v_k (1 < k \leq n)$ on T is the travel time from v_1 to v_k on T as followings:

$$t\left(v_k, T\right) = t\left(v_{k-1}, T\right) + f\left(v_{k-1}, v_k, t\left(v_{k-1}, T\right)\right) + r\left(v_{k-1}, v_k, t\left(v_{k-1}, T\right)\right) \qquad \forall k \in 1, 2, \ldots, k$$

where, start times must respect:

$$t\left(v_1, T\right) \leq \tau$$

The Time-Dependent Traveling Salesman Problem in Postdisaster (TDTSP-PD) is the problem of finding a tour $T = \left(v_1, v_2, \ldots, v_k, \ldots, v_n\right)$ which starts from the depot $\left(v_1 = s\right)$ and visits each vertex exactly once, and the returning time to the starting vertex, such that, $L(T) = t\left(v_n, T\right) + f\left(v_n, v_1, t\left(v_n, T\right)\right) + r\left(v_n, v_1, t\left(v_n, T\right)\right)$, is minimal.

1.2. Motivation

In recent years, the number of disasters that occur every year are 396. As a result, they affect about 95 million people without essential materials [1]. Therefore, timely delivery of necessary materials, as well as efficient clearing of debris, are related to effective, customer-oriented routing for vehicles. As we know, disasters destroy the infrastructure in cities that cause massive amounts of debris. There are different debris types, such as construction, vegetative, hazardous waste, white goods, freshwater, etc. [1]. Several studies on debris removal have been published in the literature, and they aim at the recovery phase of the disaster that makes the complete removal of debris. However, debris becomes a big issue in the response phase when roads completely or partially are blocked in relief logistics. Our goal is to reach destructed areas as soon as possible, while debris removal complete clearance is impossible because it takes several months to complete. Therefore, a sweeping operation can be deployed so that debris is moved aside and enough space for vehicles to pass.

The classic TDTSP [2–6] is a general variant of the Traveling Salesman Problem (TSP) [7] and Traveling Repairman Problem (TRP)

*Corresponding author. Email: BangBH@soict.hust.edu.vn

[8,9]. In the TSP and TRP, the travel time to travel is assumed to be constant. However, in practice, the travel times vary according to several factors that naturally depend on the time of day. Therefore, the TDTSP is more practical than the TSP and TRP. The TDTSP has many practical applications in scheduling time-dependent tasks [10–12], scheduling manufacturing system [6], network [13–15], timetables for university [16], schedules vehicles [16], and riding of amusement park attractions [17].

In the TDTSP [3–6], travel time function is defined in normal conditions. However, in disaster situations, travel costs need to be added to debris removal times. The novelty of the problem is to consider an extra effort. This extra effort is an additional time to sweep the debris and make enough space for vehicles. Hence, the TDTSP-PD problem is a single-vehicle routing problem that differs from the classic TDTSP in an important characteristic that is to use the blocked edges. Therefore, the vehicle must spend some extra time to unblock these edges. We can also understand this extra time as a fixed cost defined for edges.

Figure 1 describes an example of the TDTSP-PD. Dashed lines depict the blocked edges. The cost on each edge is the traveling time and debris removal time of the blocked edge. In Figure 1, a couple of values (2, 4) on edge (1, 2) mean that the traveling time from node 1 to 2 is 2 and its debris removal time is 4. For instance, a tour includes nodes 1-2-3-4-1. In this solution, the arrival time to node 2 is 6 (= 2 + 4), to node 3 is 10 (= 6 + 4), to node 4 is 20 (= 10 + 3 + 7), and to return node 1 is 24 (= 20 + 4). Therefore, the total cost is 24.

1.3. Literature Review

Though the TDTSP-PD is a natural extension of the classic TDTSP-PD, no publication can be found in the literature. After that, we describe several works to solve the TDTSP in the current. Some formulations for the TDTSP is described in [5]. In Picard *et al.* [10], Wiel *et al.* [19], and Bigras *et al.* [18], the travel time between any two positions depends on the period time of the day. Another variant of the problem is the Time-Dependent Vehicle Routing Problem (TDVRP) in [13,20], where a whole fleet must be routed instead of a single-vehicle.

The TDTSP is NP-hard because it is more challenging than the TSP and TRP [13]. In the TSP, the optimal solutions for large instances

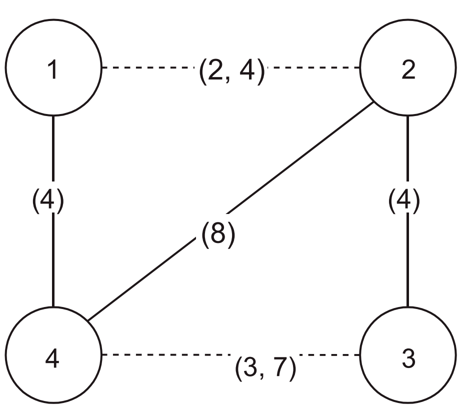

Figure 1 Example of Time-Dependent Traveling Salesman Problem in Postdisaster (TDTSP-PD).

[21] are found at a reasonable amount of time. Simultaneously, the exact algorithm [4] can solve only the TDTSP-instances with a few dozen customers. In the TSP, small changes in the structure of a metric space only affect local TSP structure changes. However, this can cause highly nonlocal changes in the structure of the TDTSP problem.

The direct works to the classic TDTSP provided in [2–6] are divided into two categories: 1) The exact algorithms [3,4,19,22] solve exactly instances with small sizes; 2) Several heuristic or metaheuristic algorithms [2,5,11,18,22,23] can produce good solutions fast for large instances. Moreover, some variants of the classic TDTSP [24,25] are known as the TDTSP Time Windows. The above algorithms are the state-of-the-art algorithms for the variants of the TDTSP. However, repairing times for broken roads and debris removal are not mentioned in these problems, and their corresponding algorithms are not easy to be adapted directly to the TDTSP-PD.

Related to the study on postdisaster road clearance and debris removal, Sahin *et al.* introduce the Debris Removal in the Response Phase problem that requests to reach a set of affected areas as soon as possible by traveling blocked roads due to debris [26]. The problem also considers an extra effort. This extra effort is an additional time to sweep the debris and makes space for vehicles. They developed mathematical models and heuristics to minimize the time to visit critical nodes. This objective is the same as minimizing the maximum latency. Another study related to debris removal is described in [27]. They propose mathematical models and heuristics to minimize 1) minimizing the maximum latency and 2) the total latency of the critical nodes. The experimental results show that their algorithm [26] obtains better solutions than Sahin *et al.* However, every time the salesman travels the same blocked edge in two problems, the debris removal time is added to the objective function. Thus, there is an over-calculation of the objective value. The TDTSP-PD problem is different from two above problems in four aspects: 1) the travel time to travel in [26,27] is a constant while in our problem it changes drastically that depends on certain time of the day; 2) the debris removal time in our problem does not recalculated on the same edges. Therefore, it does not cause a sub-optimal solution, especially when the number of blocked edges is high; 3) we prefer to minimize the arrival time instead of minimizing maximum latency and total latency; 4) in [26,27], they divide nodes into two types: critical and noncritical nodes. The feasible solution must visit critical nodes, while noncritical ones can be bypassed. Conversely, in the TDTSP-PD problem, all nodes are considered as critical nodes.

1.4. Our Novelties and Contributions

Currently, no work is published for the TDTSP-PD in literature. Our algorithm is the first metaheuristic to solve the problem. Conversely, several works for the TDTSP can be found in the literature. However, comparisons with the results in [6,12,18,22,26] would be meaningless because they were the algorithms a decade ago. In Ban [28] also show that their algorithm is much better than the algorithms in [6,12,18,22,23]. For this reason, we only choose the algorithms [3,28] for the TDTSP as a baseline in our research.

The success of a metaheuristic algorithm depends on the balance between intensification and diversification strategies. The constraint programming approach of Melgarejo [3] takes much time to find a good solution due to memory limits. On the other hand, the metaheuristic in [28] may implement a strong intensification strategy. However, its diversification cannot be sufficient. Therefore, it may get stuck into local optima. In this paper, we proposed the algorithm to overcome the above drawbacks. The main contributions of this work can be summarized as follows:

- From the algorithmic perspective, a good metaheuristic must ensure the balance between diversification and intensification. To the best of our knowledge, the work is the first population-based algorithm for the TDTSP-PD and its variants in the literature. To solve this challenging problem, we present the Memetic Algorithm (MA) for the TDTSP-PD. The MA [29] is a powerful population-based framework that balances the diversification of the Genetic Algorithm (GA) [30] and the intensification of Variable Neighborhood Search (VNS) [31].
- From the computational perspective, extensive numerical experiments on benchmark instances show the proposed algorithm's efficiency. Moreover, our algorithm reaches better solutions than the-state-of-art algorithms.

The rest of this paper is organized as follows. Section 2 presents the proposed algorithm. Computational evaluations are reported in Section 3. Sections 4 and 5 discuss and conclude the paper, respectively.

2. THE PROPOSED ALGORITHM

In this paper, an efficient metaheuristic proposed brings together the components of GA [30] and VNS [31]. We describe GA and VNS, respectively.

- The GA [30] is an evolutionary technique using the survival of the fitness idea. One encodes possible model behaviors into "genes." After each generation, the parents are allowed to mate and breed based on their fitness. In the process of mating, crossovers and mutations occur. The current population is discarded, and its offspring forms the next generation.
- The VNS is described in [31]. It is divided into two main components: The shaking and local search (LS). In the first, shaking implements the move to a random solution. The second consists of applying a LS to the solution and selecting the best one in a neighborhood set.

The MA algorithm for the TDTSP-PD is summarized in Algorithm 1. In the first step, the proposed algorithm generates an initial population. The algorithm then improves it by the VNS in the second step. In the last step, each iteration then implements a crossover and a local optimization. Specifically, the crossover selects and combines two randomly parent solutions to create an offspring. The LS then improves offspring.

Algorithm 1. The proposed algorithm

Input: v_1, V, $N_i(T)$ $(i = 1, \dots, k_{\max})$, P are a starting vertex, the set of vertices in K_n, the set of neighborhoods, the population, respectively.

Output: the best-found solution T^*.

{Step 1: Generating population}
$P \leftarrow$ **Init-Population**(v_1, V, n);
{Step 2: VNS}
for $(i = 1; i < |P|; i++)$ do
  $T_i \leftarrow \text{VNS}(T_i)$;
endfor
{Step 3: Crossover+VNS}
while (stop condition is not met) **do**
  Select parent *TP, TM* from *P*
  $TC =$ **Crossover**(TP, TM);
  $TC =$ **VNS**(TC);
  {Updating the population}
  $P =$ **Elitism-Selection**(TC, P);
endwhile
$T^* =$ Select the best individual in P;
return T^*;

Finally, an update of the population is implemented. The algorithm finishes after a fixed number of generations.

To make the algorithm's structure more detail, a flowchart of the MA algorithm is described in Figure 2. The algorithmic steps are described in the following sections.

2.1. Encoding and Evaluation

The simple encoding is used, in which a tour is represented as a list of n vertices $(v_1, \dots, v_k, \dots, v_n)$, where v_k is the k-th vertex to be visited.

2.2. Generating Population

We use two methods to create individuals. In the first method, we pick a number randomly between one and one hundred. If the number is less than β, the individual will be created randomly. Conversely, it will be the output result of the Greedy Randomized Adaptive Search Procedure (GRASP) [32]. Two methods are used for initialization to generate enough diversity of the population. The size of the population *SP* is the parameter determined in the experiments. Details of steps are given in Algorithm 2.

Algorithm 2. Init-Population

Input: v_1, V, α, β, SP are a starting vertex, the set of vertices in K_n, and the size of *Restricted Candidate List (RCL)*, threshold to control the ratio between random and greedy individuals, and the size of population, respectively.

Output: An initial solution T.

While $|P| < SP$ **do**
$T \leftarrow T \cup v_1$;
$type = \text{rand}(1)$;
if $(Type \leq \beta)$ **then**
  while $|T| < n$ **do**
    Select randomly vertex $v \in \{v_i | v_i \in V \text{ and } v_i \notin T\}$;
    $T \leftarrow \{v_i\}$;
  endwhile

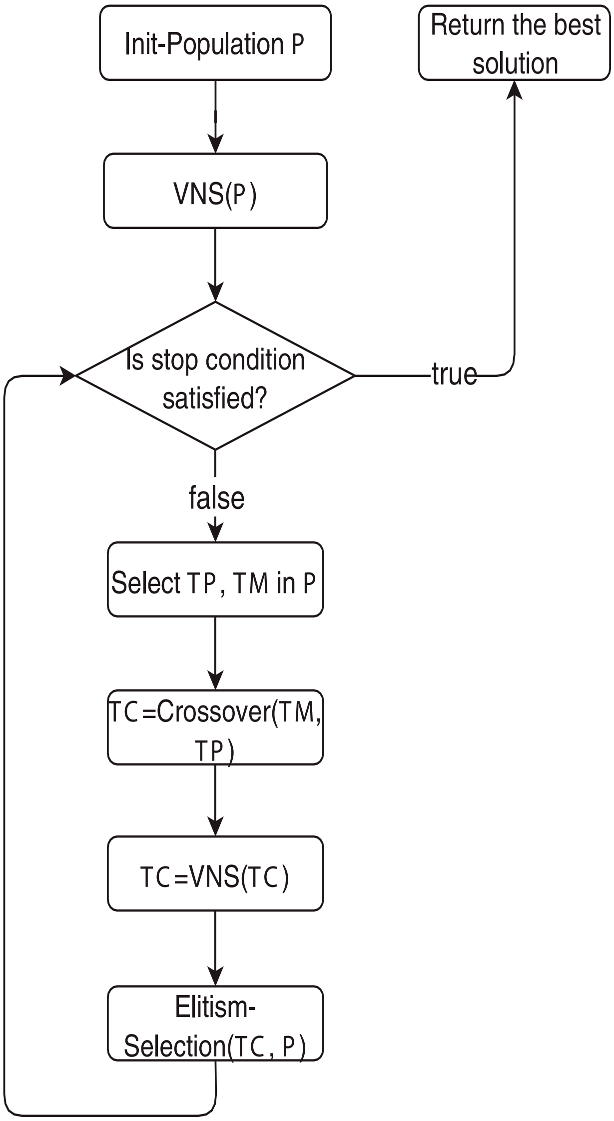

Figure 2 The flowchart of the Memetic Algorithm (MA) algorithm.

else
  while $|T| < n$ **do**
    $\{v_e$ is the last vertex in $T\}$
    Create RCL that includes α nearest vertices to v_e in V;
    Select randomly vertex $v \in \{v_i | v_i \in RCL \text{ and } v_i \notin T\}$;
    $T \leftarrow \{v_i\}$;
  endwhile
endif
$P = P \leftarrow \{T\}$;
return P;

2.3. Variable Neighborhood Search

2.3.1. Neighborhoods

We use neighborhoods widely applied in the literature to explore this problem's solution space [7]. Assume that T, and n are a tour, and its size, respectively. We describe more details about five neighborhoods ($k_{max} = 5$) as follows:

- shift (N_1) relocates a vertex to another position in T.
- swap-adjacent (N_2) tries to swap each pair of adjacent vertices in T.
- swap (N_3) tries to swap the positions of each pair of vertices in T.
- 2-opt (N_4) removes each pair of edges from the tour and reconnects them.
- Or-opt (N_5) reallocates three adjacent vertices to another position of T.

The VNS executes neighborhood procedures in turn and shaking techniques. At each iteration, the best neighboring solution is chosen from neighboring solutions generated from a neighborhood procedure. If it is better than the current best one, the procedure is repeated. Otherwise, the search goes to the next neighborhood procedure. The detail is described in Algorithm 3.

Algorithm 3. VNS

Input: T, $nloop$, k_{max} are an initial solution, the number of neighborhoods, the number of iterations, and the number of neighborhoods, respectively.
Output: the best solution T^*.
$level = 1$;
T = Variable Neighborhood Decent (VND) (T);
while ($level < nloop$) do
  $T' =$ Perturbation ($T, level, \rho$);
  $T'' = VND(T')$; {implement VND}
  if $(L(T'') < L(T)) \,\|\, (L(T'') < L(T^*))$ **then**
    $T = T''$;
    if $(L(T'') < L(T^*))$ then
      $T^* = T''$;
    endif
  endif
  if (T is equal T'') **then**
    $level = 1$;
  else
    $level{+}{+}$;
  endif
endwhile
return T^*;

2.3.2. Perturbation

The Perturbation mechanism is very important to escape local optima. When the mechanism has too small moves, the search can return to the previously visited solution space. On the other hand, large moves drive the search to undesirable space. To balance the strength of shaking (notation: $level$), we propose a new shaking technique based on the original double-bridge technique [33]. The detail is described in Algorithm 5.

Algorithm 4. VND

Input: k_{max} are an initial solution, and the number of neighborhoods, respectively

Output: the best solution T^*.

$k = 1$;
repeat
  Find the best neighborhood T' of $T \in N_k(T)$; {T' must be feasible solution}
  if $\left(L\left(T'\right) < L(T)\right) \,\|\, \left(L\left(T'\right) < L(T^*)\right)$ **then**
    $T = T'$; {centre the search around T' and search again in the first neighborhood}
    if $\left(L\left(T'\right) < L(T^*)\right)$ **then**
      $T^* = T'$;
    endif
    $k = 1$;
  else
    $k = k + 1$; {switch to another neighborhood}
  endif
until $k < k_{max}$;
$T^* = T'$;
return T^*;

The fitness function represents the evaluation of individuals. This evaluation function will check the cost of an individual. The less the cost is, the better the individual is.

Algorithm 5. Perturbation

Input: *level*, ρ are the tour, the parameter to control the strength of the perturbation, and threshold ratio, respectively.

Output: a new tour T.

$k = 1$;
while ($k < level$) do
  T' = double-brigde(T);
  $T'' \leftarrow \arg\min N_k\left(T'\right)$; {$T^*$ is the best solution}
  If $\left(L\left(T''\right) > (1-\rho) \times L(T^*)\right)$ **then**
    $T = T''$;
    break;
  else
    $k++$;
  endif
endwhile
return T;

2.4. Selection Operator

The selection phase is when the individuals are selected based on their fitness to mate and produce new offspring. In this work, the simple selection operator is used [34]. A group of *number of individuals (NG)* with a specified size is selected on a random basis. Then, two individuals that have the best fitness in the group will be chosen. The fitness difference provides selection pressure. Increased selection pressure can be provided by simply increasing the size of the group, as the winners from a larger size will, on average, have higher fitness than the winners of a small size.

2.5. Crossover

Crossover operators are significant as their exploratory force because of their ability to explore solutions over a wider search space area. Crossover is the process that mimics mating between two individuals to produce children. In Otman and Jaafar [35], the crossover operators are classified as follows: 1) The crossover operators focus heavily on the position of certain genes in the parents such as Partially Mapped Crossover (PMX), Cycle Crossover (CX), Position-Based Crossover (POS), etc.; 2) The crossovers create an offspring by selecting genes alternately from the parents while the repetition of genes is abandoned such as Edge Exchange Crossover (EXX), Edge Assembly Crossover (EAX), Heuristic Crossover (HGreX), etc.; 3) The crossovers inherit the order of genes from parent to the offspring such as Sorting Crossover (SC), Merging Crossover (MC), and Uniform Like Crossover (ULX), etc. We found no logical explanation of which one should bring better performance or better overall results. That means there is not the best crossover for all cases. In a pilot study, we found that the algorithm's performance is relatively insensitive to crossover operators. As testing our algorithm on all operators would have been computationally too expensive, we implement our numerical analysis on some selected operators for each type. In this work, the following operators are selected: type 1 (PMX), type 2 (EXX), and type 3 (SC) [35]. Using multiple crossovers makes the population more diverse than using only one crossover. Therefore, it can help the algorithm to prevent being trapped in a local optimum. The detail in this step is given in Algorithm 6.

Algorithm 6. Crossover

Input: TP, TM are the parent tours, respectively.

Output: A new child T.

If (*type*==1) then
{type 1 is chosen}
  TC = **PMX**(TP, TM); {PMX is chosen}
else if (*type*==2)
{type 2 is chosen}
  TC = **EXX**(TP, TM); {EXX is chosen}
else if (*type*==3)
{type 3 is chosen}
  TC = **SC**(TP, TM); {SC is chosen}
endif
return TC;

2.6. Updating the New Population

Each new solution generated by the VNS should be have appeared in the population. If it is better than the worst solution in the population, then it replaces the worst one. On the other hand, the population remains unchanged.

2.7. Stop Condition

The last aspect to discuss is the stop criterium of our algorithm. A balance must be made between computation time and efficiency. Here, the algorithm stops if no improvement is found after m loops.

3. EVALUATIONS

Our algorithm is implemented on a Pentium 4 core i7 2.40 GHz processor with 8 GB of RAM. In all experiments, parameters $\alpha, level, \rho, m, nloop, SP, \beta, NG$ are respectively set to 10, 5, 0.1, 50, 10, 50, 0.2, and 5, respectively. These parameters were chosen through empirical tests, and with them, the algorithm seems to produce good solutions at a reasonable amount of time compared to the other parameter values.

3.1. Instances

Our datasets are the TDTSP-benchmarks in [36]. Their instances are generated from 255 locations randomly chosen from a list of tours of drivers in Lyon. In this dataset, the number of time steps is 130. At the same time, each duration is 360 seconds. On hundred instances are generated for each problem size from 20 to 100. The duration of each visit is randomly selected in [60, 300] seconds. Due to the traffic jam, they create three travel time functions. In this experiment, 180 instances from 50 to 100 vertices are chosen.

To generate different scenarios in disaster situations, we assumed five levels of earthquake severity (*LES*), which varies from 1 to 5. Since the level is 1, there is a less severe earthquake. On the other hand, when the level is 5, it yields the highest severe earthquake. The higher and higher severe earthquake is, the more and more broken edges occur. Table 1 illustrates the broken edge ratios (*BER*) according to the severe earthquake. Repairing times $r\left(v_i, v_j, t\right)$ are calculated according to $r\left(v_i, v_j, t\right) = LES \times f\left(v_i, v_j, t\right)$ depending on the *LES* values and travel times. A similar generation for the dataset can be found in [26]. All instances are available at https://drive.google.com/file/d/1PboibMaMWfj11tH5J38u_3r45OXV3HeO/view?usp=sharing.

3.2. Metrics

We define the proposed algorithm's improvement with respect to the upper bound (*UB*) obtained by the Nearest Neighborhood Search [7]. The Nearest Neighborhood Search is not promising theoretically; however, it yields good enough solutions in practice.

$$gap\,[\%] = \frac{Best.Sol - UB}{UB} \times 100\%$$

Table 1 *LES* and corresponding *BER* values.

LES	*#BER*
1	0%–20%
2	20%–40%
3	40%–60%
4	60%–80%
5	80%–100%

BER, broken edge ratios; LES, levels of earthquake severity.

In addition, our solutions are also compared to the optimal or best solutions [3,4,28] for variants though it is not designed to solve them. The proposed algorithm runs on the same instances with the other algorithms. In all Tables, the same or improved results are highlighted in boldface and red, respectively. In addition, *OPT, Best.Sol, Aver.Sol*, and T correspond to the optimal solution, best solution, average solution, and average time in seconds of ten executions obtained by the proposed algorithm.

3.3. Comparison with UB

Tables A1–15 compare the results of the proposed algorithm with the UB. The values in Table 2 are the average ones calculated from Tables A1–15 in Appendix.

In Table 2, for each dataset, the average gap between the UB and our best solution varies between −14.05% and −18.10%. Obviously, the improvement is large and significant. These results further indicate the proposed algorithm reaches good solutions fast.

3.4. Comparison the Difference between the Objective Values of the TDTSP and TDTSP-PD

In Tables A1–15 in the Appendix, we also see that the higher and higher the value of *LES* is, the larger and larger the objective function cost is. It is suitable in practical situations because debris removal time for a high level of earthquake severity consumes more time than the lower level cases.

Table 5 shows the difference between the objective value of the TDTSP and TDTSP-PD with five values of LES. he results show that when *LES*'s value is small (*LES* = 1), the difference between them is not large because debris removal operation takes a little time to unblocked roads in this case. Conversely, the difference becomes large with the *LES*'s other values (*LES* = 2, 3, …, 5). It shows that most of the time in these cases is used to clear debris other than travel on the roads.

3.5. Comparison with TS+VNS

Recently, Tabu Search (TS) [28] has been successfully applied to solve the TDTSP. To compare directly with the TS+VNS, we adapt the TS+VNS algorithm for the TDTSP to the TDTSP-PD case. Fortunately, Ban [28] support us to access their code. We also remain the parameter settings unchanged.

In Table 2, the results show that the robustness of the proposed algorithm. More precisely, the average gap between the two algorithms is −2.44%. In 900 instances, our algorithm obtains better solutions in 407 cases and the same solutions in 203 cases (see more detail in Tables A1–15 in the Appendix). To indicate a clear dominance of the proposed algorithm over the TS+VNS, Wilcoxon's test [37] is used. Table 3 shows Wilcoxon signed ranks test results between the TS+VNS and MA in 900 cases. The results indicate that the MA shows a significant improvement over the TS+VNS with a level of significance $\alpha = 0.05$.

Table 2 The average experimental results for TDTSP-PD with instances that are proposed by [25].

Instances	Travel Time Function 1			Travel Time Function 2			Travel Time Function 3		
	TS+VNS	MA		TS+VNS	MA		TS+VNS	MA	
	gap [%]	gap [%]	*T*	gap [%]	gap [%]	*T*	gap [%]	gap [%]	*T*
INST-30-x (LES=1)	−9.9	−10.66	4.12	−7.75	−8.21	4.14	−7.14	−9.14	4.14
INST-30-x (LES=2)	−5.38	−5.38	4.14	−4.3	−12.58	4.14	−3.56	−11.79	4.14
INST-30-x (LES=3)	−5.35	−19.12	4.26	−5.8	−18.27	4.26	−16.55	−21.01	4.26
INST-30-x (LES=4)	−9.55	−18.84	4.27	−1.8	−20.74	4.26	−5.19	−23.52	4.26
INST-30-x (LES=5)	−16.99	−23.49	4.32	−16.3	−23.19	4.32	−24	−17.62	4.32
INST-50-x (LES=1)	−6.36	−7.65	70.85	−3.74	−6.3	70.49	−4.6	−6.64	71.39
INST-50-x (LES=2)	−4.85	−10.66	70.76	−1.17	−9.95	70.87	−0.43	−8.87	70.91
INST-50-x (LES=3)	−16.62	−16.56	71	−13.71	−13.74	70.83	−15.59	−15.92	70.71
INST-50-x (LES=4)	−24.07	−25.93	71.33	−22.69	−23.17	70.96	−21	−21.31	71.34
INST-50-x (LES=5)	−28.16	−29.61	70.42	−25.91	−27.1	71.8	−26.82	−28.62	70.51
INST-100-x (LES=1)	−7.27	−9.12	222.2	−7.09	−8.23	222.6	−7.72	−9.12	222.6
INST-100-x (LES=2)	−15.94	2.68	222.1	−14	5.32	221.8	−14.91	3.59	222.4
INST-100-x (LES=3)	−26.68	−28.68	222.5	−25.24	−27.7	222.6	−21.42	−24.2	221.9
INST-100-x (LES=4)	−29.85	−33.59	222.5	−28.37	−31.88	221.9	−26.06	−32.22	221.9
INST-100-x (LES=5)	−34.7	−34.84	222.5	−32.92	−32.97	222.3	−31.05	−31.67	222.5
Aver	−16.11	−18.10		−14.05	−17.25		−15.07	−17.20	

MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; VNS, Variable Neighborhood Search.

Table 3 Wilcoxon signed ranks test results between the TS+VNS and MA for the TDTSP-PD and TDTSP with a level of significance $\alpha = 0.05$.

Problems	R^+	R^-	*p* -value
TDTSP-PD	109552	−53299	<0.0001
TDTSP	2215	−474	<0.0001

TDTSP, Time-Dependent Traveling Salesman Problem; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; VNS, Variable Neighborhood Search.

Table 4 Comparison for the running time by seconds between TS+VNS and MA for the TDTSP-PD.

Instance	TS+VNS	MA
INST-30-x	1.32	4.22
INST-50-x	5.89	70.94
INST-100-x	38.51	222.28
Aver	15.24	99.14

MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; VNS, Variable Neighborhood Search.

To compare their time complexity, three perspectives [23] can be considered as follows:

- The theoretical time complexity: The time complexity of the MA mainly spends to explore in VNS. In VNS, the time complexity of the Or neighborhood is not less than those of the other neighborhoods. Assume that, when k is its maximum number of runs in the algorithm, the MA requires $O\left(m \times |SP| \times k \times n^3\right) \sim O\left(n^3\right)$ times. The theoretical time complexity of the TS+VNS is $O\left(k \times n^3\right) \sim O\left(n^3\right)$ times. Thus, the theoretical time complexity of the two algorithms is the same.
- The time complexity by CPU times: Ban [28] support us to access their code. Therefore, two algorithms are run on the same computer languages, platforms, and compilers. It is convenient to compare the running time of them by CPU times. In Table 4, the running time of the proposed algorithm grows moderate compared to the TS+VNS. The result is understandable because a population-based metaheuristic often consumes more time than a single-based one.
- The time complexity by function evaluations: For several expensive optimization problems, the function evaluation reflects an algorithm's time complexity. In the TDTSP-PD, the evaluation function is not too expensive when it consumes $O(n)$ time. The fundamental evaluations do not completely dominate the internal workings of the algorithm. However, to help us evaluate two algorithms' time complexity more exactly, the maximum number of fitness function evaluations is also mentioned. We count the maximum number of fitness evaluations such that the MA obtains the best solution. After that, we run the TS+VNS with the same maximum number of fitness evaluations. The results of the TS+VNS still remains unchanged. The TS+VNS cannot find any better solutions in all instances though in many cases it may be run with the additional number of fitness evaluations. The TS+VNS might have a strong search for intensification capacity. Their diversification technique may not be sufficient to bring the search to unexplored search space regions. Therefore, it can get stuck into local optima. The additional number of fitness evaluations does not help it to improve the solution quality. Due to the random nature, our population-based algorithm can explore a more extensive solution space. As a result, the chance of finding the optimal solution is higher.

In summary, on average, the MA consumes more time than the TS+VNS. Nevertheless, the fact that their running time difference is quite moderate. In addition, the TS+VNS does not improve solution quality though the additional number of fitness evaluations may be allowed. In that case, we can still say that our MA is beneficial.

3.6. Experimental Results on TDTSP Instances

From Tables 6 and 7, we know that most algorithms are developed for a specific variant that does not apply to the other variants. Our algorithm still runs well to the TDTSP, although it was not designed for solving it. In comparison with the best-known solution in [3,28], our algorithm's solutions obtain better solutions than the Melgarejo's algorithm [3], and Ban's algorithm [28] in many cases. The improvement is significant since it can be observed that our algorithm is capable of finding the new best solutions in 69 instances. To have more clear statistical comparison, the Wilcoxon's test is still used. Table 3 shows that the MA shows a significant improvement over the TS+VNS with a level of significance $\alpha = 0.05$. However, the proposed algorithm consumes more time compared to Melgarejo's [3] and Ban's algorithm [3,28] (see Table 8).

Table 5 The difference objective value between TDTSP-PD and TDTSP with the values of *LES*.

Instances	Travel Time Function 1	Travel Time Function 2	Travel Time Function 3
	diff [%]	diff [%]	diff [%]
INST-50-x (LES=1)	16.06	14.24	70.49
INST-50-x (LES=2)	37.12	37.12	70.76
INST-50-x (LES=3)	75.28	71.00	68.11
INST-50-x (LES=4)	160.44	151.33	153.71
INST-50-x (LES=5)	184.22	176.50	174.14
INST-100-x (LES=1)	22.80	24.83	222.6
INST-100-x (LES=2)	68.85	74.08	76.64
INST-100-x (LES=3)	100.63	105.76	111.50
INST-100-x (LES=4)	157.90	154.80	155.41
INST-100-x (LES=5)	256.84	243.78	252.63

MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; VNS, Variable Neighborhood Search.

Table 6 The experimental results for TDTSP with INST-50-x instances that are proposed by [25].

Instances	Travel Time Function 1					Travel Time Function 2					Travel Time Function 3				
	PAM	BA	MA			PAM	BA	MA			PAM	BA	MA		
			Best .Sol	*Aver .Sol*	*T*			*Best .Sol*	*Aver .Sol*	*T*			*Best .Sol*	*Aver .Sol*	*T*
INST-50-1	22795	22846	**22537**	22537	84.05	22108	22108	**22010**	22010	83.76	21678	21718	**21678**	21678	83.16
INST-50-2	23421	23713	23421	23421	83.74	22661	23233	**22475**	22475	81.14	22676	22287	**22108**	**22108**	81.78
INST-50-3	22684	22684	**22431**	22431	80.60	21777	21777	**21777**	21777	80.32	21382	21217	21354	**21354**	84.99
INST-50-4	24396	24833	24406	24406	82.63	23579	23610	**23553**	23553	83.84	22949	22976	22949	22949	81.12
INST-50-5	20960	20960	**20960**	20965	81.63	20877	20877	**20877**	20877	83.36	20553	20610	20658	**20658**	83.26
INST-50-6	22074	22396	**21966**	21966	82.73	21380	21795	21795	21795	83.58	21276	20950	**21046**	**21046**	83.02
INST-50-7	23241	23241	**23241**	23126	81.99	22645	22645	22650	22650	83.21	22308	22371	22464	**22464**	81.94
INST-50-8	23274	23274	**23274**	23274	82.08	22558	22558	22576	22576	82.10	22182	22014	**22005**	**22005**	80.71
INST-50-9	22549	22549	**22549**	22549	80.90	22015	22015	**22015**	22015	81.95	21669	21913	21669	21669	80.13
INST-50-10	22556	22831	**22195**	22195	81.28	21861	22249	**21655**	21655	84.08	21928	21427	21429	**21429**	82.11
INST-50-11	23775	23893	**23530**	23530	80.10	23015	23063	**22974**	22974	81.59	22230	22490	22357	**22357**	80.92
INST-50-12	24487	24610	**24035**	24035	84.62	23792	23969	23792	23792	84.07	23568	23131	**23100**	**23100**	83.63
INST-50-13	23432	23432	**23148**	23148	83.27	22585	22585	**22578**	22578	83.95	22190	22121	**22095**	**22095**	81.85
INST-50-14	23452	23678	**23389**	23389	84.66	22677	22737	22729	22729	84.26	22103	22238	22211	**22211**	84.21
INST-50-15	22473	22473	22829	22829	80.82	21838	21838	21869	21869	82.53	22057	21490	**21440**	**21440**	83.67
INST-50-16	23538	23538	**23509**	23509	84.61	23032	23035	**22962**	22962	83.18	22463	22461	22463	22463	82.86
INST-50-17	24028	24100	24028	23862	83.97	23273	23311	**23055**	23055	84.75	23207	22644	**22753**	**22753**	80.88
INST-50-18	21181	21539	21181	21181	82.89	20785	20785	20832	20832	82.22	20681	20483	**20483**	20483	84.79
INST-50-19	24477	24477	24501	24501	82.20	23899	23899	**23821**	23821	80.30	23332	23274	**23268**	**23268**	81.33
INST-50-20	22528	23170	**23170**	23170	81.29	22180	22499	**22071**	22071	84.33	21984	21655	21691	21691	84.62
aver					82.50					82.93					82.55

MA, Memetic Algorithm; TDTSP, Time-Dependent Traveling Salesman Problem.
The bold values in Table 6 means that our solutions are better or at least as well as the other algorithms (PAM and BA).

Table 7 The experimental results for TDTSP with INST-100-x instances that are proposed by [25].

Instances	Travel Time Function 1					Travel Time Function 2					Travel Time Function 3				
	PAM	BA	MA			PAM	BA	MA			PAM	BA	MA		
			Best .Sol	*Aver .Sol*	*T*			*Best .Sol*	*Aver .Sol*	*T*			*Best .Sol*	*Aver .Sol*	*T*
INST-100-1	39532	39249	39249	39205	231.12	39336	38245	**37801**	37801	231.38	36884	36884	**36431**	36431	230.69
INST-100-2	36015	36015	**36015**	35563	231.87	35065	35143	**34419**	34419	231.24	34793	33987	34793	34793	231.09
INST-100-3	36807	36807	36869	36869	230.44	37240	35729	**35330**	35330	232.26	37000	34477	**34470**	34470	230.91
INST-100-4	39631	39631	**39284**	39284	233.20	38954	38240	**38027**	38027	231.14	39558	37765	**36811**	36811	230.21
INST-100-5	38396	37377	**36757**	36757	230.90	35852	35852	**35685**	35685	234.02	34580	34580	35056	35056	230.53
INST-100-6	37957	38312	**37444**	37444	230.23	36508	36508	**36163**	36163	234.93	34950	34950	35243	35243	233.08
INST-100-7	41457	40371	**40460**	40460	233.62	37971	39081	38425	38425	230.15	36395	36395	37217	37217	234.70
INST-100-8	39302	39302	39847	39847	231.74	39325	39257	**38085**	38085	232.68	37250	37250	**37064**	37064	231.77
INST-100-9	38189	38189	38189	37622	233.30	36813	36813	36813	36813	230.44	35808	35808	**35250**	35250	232.05
INST-100-10	40021	40021	**40021**	39181	231.92	37958	37958	**37658**	37658	234.01	37185	37185	**36589**	36589	234.92
INST-100-11	42661	40486	40681	40681	233.14	40937	40501	**40501**	40501	234.95	39210	39210	**38657**	38657	234.73
INST-100-12	41026	41122	**40559**	40559	230.11	38727	38727	38982	38982	230.33	37730	37730	38163	38163	233.38
INST-100-13	39439	39439	**39423**	39423	234.55	38188	38188	**37660**	37660	234.70	37023	37023	**36422**	36422	234.94
INST-100-14	38089	38089	**38089**	37151	234.00	36358	36358	**36008**	36008	230.09	35553	35553	**34884**	34884	233.83
INST-100-15	39815	39826	**39189**	39189	233.73	39706	38599	**37652**	37652	233.42	38680	38680	**38680**	38680	231.68
INST-100-16	39128	39128	39324	39324	234.07	38964	37923	**37623**	37623	233.92	37010	37239	**36242**	36242	233.31
INST-100-17	41595	41595	**41508**	41508	231.92	40181	39928	**39563**	39563	232.67	39438	39438	**39071**	39071	231.22
INST-100-18	39582	39582	**39144**	39144	233.09	39486	39302	39486	39486	234.43	37977	37977	**37135**	37135	231.48
INST-100-19	40032	40032	**39457**	39457	232.88	38459	38459	38459	38459	234.50	38071	37949	**36556**	36556	233.40
INST-100-20	40723	40163	**39402**	39402	232.65	38212	38212	**37765**	37765	233.13	36619	36619	**36619**	36619	232.64
aver					232.42					232.72					232.53

MA, Memetic Algorithm; TDTSP, Time-Dependent Traveling Salesman Problem.
The bold values in Table 7 means that our solutions are better or at least as well as the other algorithms (PAM and BA).

Table 8 Comparison for the running time by seconds between TS+VNS and MA for the TDTSP.

Instance	TS+VNS	MA
INST-50-x	5.70	82.66
INST-100-x	37.10	232.56
Aver	21.4	157.61

MA, Memetic Algorithm; TDTSP, Time-Dependent Traveling Salesman Problem; TS, Tabu Search; VNS, Variable Neighborhood Search.

Table 9 Comparison with the optimal solution of the TDTSP-instances that are proposed in [4].

Instances	*OPT*	*Best.Sol*	*Aver.Sol*	*T*
dantzig42	12528	**12528**	12528	42.5
att48	209320	**209320**	209320	41.5
eil51	10178	**10178**	10178	62.6
berlin52	143721	**143721**	143721	60.4
st70	20557	**20557**	20557	78.4
KroA100	983128	**983128**	983128	221.5
KroB100	986008	**986008**	986008	224.2
KroC100	961324	**961324**	961324	224.3
KroD100	976965	**976965**	976965	223.2

TDTSP, Time-Dependent Traveling Salesman Problem.
The bold values in Table 9 means that our solutions are better or at least as well as the other algorithms (PAM and BA).

Table 9 shows the results of the proposed algorithm on instances with up to 100 vertices [4]. These solutions are compared with the optimal values published in [4]. Our algorithm obtains the optimal solutions for all instances in a short time.

3.7. Comparison of the Best-Found TDTSP-PD-Solution with the Best-Found TDTSP-Solution Using the TDTSP-PD's Objective Function

This experiment compares the best-found TDTSP-PD-solution with the best-found TDTSP-solution using the TDTSP-PD's objective function. Because running our algorithm in all instances is too expensive, we run our numerical analysis on some selected instances.

Table 10 shows that good TDTSP solutions are generally not a good solution to the TDTSP-PD in the same instances. On average, the best solution founds by our algorithm is about 19.18% better than the best-found TDTSP-solution using the TDTSP-PD's objective function. Therefore, the methods designed for the TDTSP instances may not be adapted easily to solve the TDTSP-PD. The results demonstrate that developing a suitable algorithm for the TDTSP-PD is necessary.

3.8. Diversification and Intensification Balance

In the proposed algorithm, crossover operators are a very useful technique to make jumps in the solution space. Therefore, they help our algorithm to have a diversification of the solution space. However, they do not make an exhaustive search. To get an intensive search, the VNS can handle this aim easily. The ability of multiple crossovers to maintain diversification is indicated in [38]. In this

Table 10 Comparison of the best-found TDTSP-PD-solution with the optimal TSPTD-solution using the TDTSP-PD-objective function.

Instances	*LES* = 1			*LES* = 2			*LES* = 3		
	The Best TDTSP	The Best TDTSP-PD	%diff	The Best TDTSP	The Best TDTSP-PD	%diff	The Best TDTSP	The Best TDTSP-PD	%diff
INS-50-1	25565	24551	−4.0	25895	25485	−1.6	34955	29433	−15.8
INS-50-2	27615	27090	−1.9	28307	27898	−1.4	45781	35806	−21.8
INS-50-3	29946	28186	−5.9	34385	32493	−5.5	64614	56623	−12.4
INS-50-4	29344	29102	−0.8	36251	34384	−5.2	32379	29183	−9.9
INS-50-5	24501	24451	−0.2	27906	26851	−3.8	56925	43905	−22.9
INS-100-1	49685	46945	−5.5	59258	51483	−13.1	108940	76245	−30.0
INS-100-2	46631	44430	−4.7	87455	59104	−32.4	109775	70151	−36.1
INS-100-3	43882	43543	−0.8	82772	61801	−25.3	159057	80186	−49.6
INS-100-4	50947	50026	−1.8	83083	71059	−14.5	50987	50654	−0.7
INS-100-5	48216	47447	−1.6	62233	59836	−3.9	157749	98291	−37.7
aver			−2.7			−10.7			−23.7
	LES = 4			*LES* = 4					
	The Best TDTSP	The Best TDTSPPD	%diff	The Best TDTSP	The Best TDTSPPD	%diff			
INS-50-1	89092	64111	−28.0	144347	81046	−43.9			
INS-50-2	113361	74348	−34.4	39505	36006	−8.9			
INS-50-3	114362	89086	−22.1	84600	60948	−28.0			
INS-50-4	46200	45174	−2.2	30878	29728	−3.7			
INS-50-5	42712	38994	−8.7	93639	68421	−26.9			
INS-100-1	69091	53042	−23.2	254973	153335	−39.9			
INS-100-2	223815	128683	−42.5	345151	203116	−41.2			
INS-100-3	199050	103161	−48.2	353613	235556	−33.4			
INS-100-4	152242	102419	−32.7	411166	238187	−42.1			
INS-100-5	164619	93392	−43.3	92912	59984	−35.4			
aver			−28.5			−30.3			

LES, levels of earthquake severity; TDTSP, Time-Dependent Traveling Salesman Problem; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster.

experiment, to study the capacity of the VNS to exploit the search space, an experiment about the distribution of locally optimal solutions.

The metric distance between two tours is defined as the minimum number of transformations from one to another. We define the distance to be n minus the number of vertices with the same position on both tours. We have selected the instance (INST-30-1) with three traveling time matrices. Running the instance with a time limit of 5 seconds, we obtain distinct local optima. Then, we build a matrix M (r columns and r rows) in which each element M_{ij} represents the distance between the solution.

T_i and T_j. Finally, we map the r points from the n-dimensional space into the Euclidean space R^2. Figures 2–4 describe these points in the Euclidean space R^2.

In Figures 3–5, the initial solution (blue point) appears to be central to all other local minima, and the distances between distinct local optima and initial solution are quite large, which implies that the VNS exploits a broad region of the solution space.

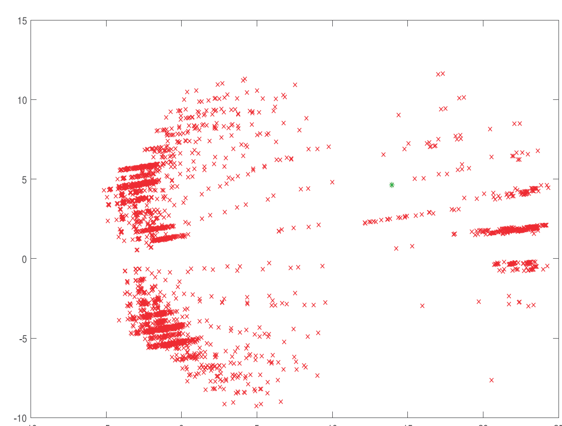

Figure 3 The solution distribution of INS-30-1 with traveling time function 1.

4. DISCUSSIONS

For NP-hard problems, there are three popular approaches to solve the problem, namely, 1) exact algorithms, 2) approximation algorithms, 3) heuristic (or metaheuristic) algorithms. Firstly, the exact algorithms guarantee to find the optimal solution and take exponential time in the worst case, but they often run much faster in practice. The best exact algorithms solve the TDTSP with the instances with up to 50 vertices [6,24,36]. Secondly, an α-approximation algorithm produces a solution within some factor of α of the optimal solution. However, the best approximation ratio is often far from the optimal solution. Thirdly, metaheuristic algorithms perform well in practice and validate their empirical performance on an experimental benchmark of interesting instances. The TDTSP-PD is NP-hard; therefore the metaheuristic algorithm is a natural approach to solve large instances in a short time.

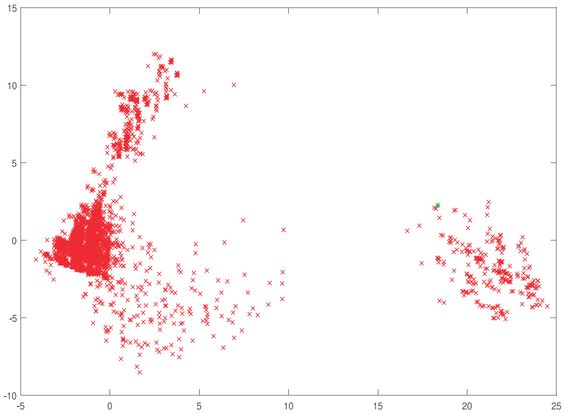

Figure 4 The solution distribution of INS-30-1 with traveling time function 2.

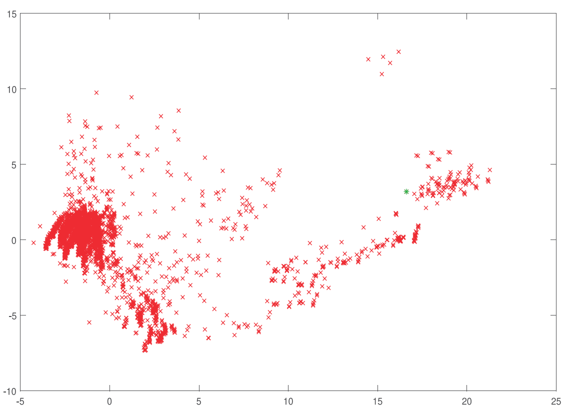

Figure 5 The solution distribution of INS-30-1 with traveling time function 3.

Generally speaking, metaheuristic algorithms can get stuck into local optimum because there is a lack of the balance between diversification and intensification in which diversification means generating diverse solutions to explore the search space on a global scale. In contrast, intensification means to focus on the search in a local region by exploiting the information that a current good solution is found in this region. While the algorithms in [3,4,28] might have a strong search intensification, their diversification mechanisms may not be sufficient. Due to the random nature, population-based algorithms improve on the chance of finding a globally. The same idea can be found in [39–41].

To the best of our knowledge, a population-based algorithm has never been proposed for the TDTSP-PD in the literature. To tackle this computationally challenging problem, we present the first MA for the TDTSP-PD. The MA [29] is a robust population-based framework that combines the exploration from the GA and LS optimization's exploitation capacity. This work's main contributions can be summarized as follows: 1) from the algorithmic perspective, the proposed MA that brings the advantages of the GA and LS maintains the balance between diversification and intensification. In our algorithm, the GA is used to explore the promising solution areas that are yet refined while the VNS exploits them with the hope of improving a solution; 2) from the computational perspective, our algorithm obtains good solutions fast. Compared with the TS+VNS algorithm [28] in 900 instances, the proposed algorithm reaches better solutions in 407 cases and the same solutions in 203 cases. We also adapt the proposed algorithm for the TDTSP case. Our algorithm shows the proposed algorithm's highly competitive performance compared to the state-of-the-art algorithms for the TDTSP [3,28]. Moreover, the proposed algorithm finds the new best solutions in 69 out of 120 cases. It is a significant improvement because the algorithms in [3,28] are the best algorithms in current for the problem. Wilcoxon's test's statistical results also show the clear dominance of the proposed algorithm in comparison with the state-of-the-art algorithms in the literature. A research topic is increasing our algorithm's efficiency and running time to allow even larger problems to be solved in the future.

5. CONCLUSIONS

In this work, the TDTSP in Post Disaster is studied. As our main contribution, we propose a MA that combines the GA and LS to solve the problem. In our algorithm, the GA is used to explore the promising solution areas that are yet refined, while the VNS exploits them with the hope of improving a solution. The proposed algorithm maintains a balance between intensification and diversification. The performance of the proposed MA is evaluated on benchmark datasets for the TDTSP-PD and TDTSP. For the TDTSP-PD, the proposed algorithm finds high-quality solutions when the average gap from the UB is from 14.05% to 18.10%. Compared with the algorithm [28], the proposed algorithm obtains better solutions in 407 and the same results in 203 out of 900 cases. For the TDTSP, our algorithm can find the new best-known solutions for 69 TDTSP-instances. In addition, the instances with 100 vertices can be solved exactly in a short time. Wilcoxon's test's statistical results also show the clear dominance of the proposed algorithm in comparison with the state-of-the-art algorithms in the literature. A research topic is increasing our algorithm's efficiency and running time to allow even larger problems to be solved in the future.

CONFLICT OF INTEREST

The author declares there is no Conflict of Interest.

AUTHORS' CONTRIBUTION

The author designed and implemented the algorithms.

ACKNOWLEDGMENTS

This research is funded by Hanoi University of Science and Technology (HUST) under grant number T2020-PC-008.

REFERENCES

[1] Disaster Year in Review 2019, Cred Crunch, No. 58, Centre For Research On The Epidemiology Of Disasters (Cred), 2020, pp. 1–2, https://cred.be/sites/default/files/CC58.pdf

[2] R. Almahasneh, B. Tüű-Szabó, L.T. Kóczy, P. Földesi, Optimization of the time-dependent traveling salesman problem using interval-valued intuitionistic fuzzy sets, Axioms. 9 (2020), 1–14.
[3] P.A. Melgarejo, A Constraint Programming Approach for the Time Dependent Traveling Salesman Problem, Computer Science, INSA Lyon, 2016. English. https://hal.archives-ouvertes.fr/hal-01514369/document
[4] H.G. Abeledo, G. Fukasawa, R. Pessoa, A. Uchoa, The time dependent traveling salesman problem: polyhedra and branch-cut-and price algorithm, J. Math. Prog. Comp. 5 (2013), 27–55.
[5] L. Gouveia, S. Voß, A classification of formulations for the (time-dependent) traveling salesman problem, J. Eur. Oper. Res. 83 (1995), 69–82.
[6] A. Lucena, Time-dependent traveling salesman problem - the deliveryman case, J. Netw. 20 (1990), 753–763.
[7] D.S. Johnson, L.A. McGeoch, The traveling salesman problem: a case study in local optimization, in: E. Aarts, J.K. Lenstra (Eds.), Local Search in Combinatorial Optimization, Princeton University Press, Princeton, NJ, USA, 2003, pp. 215–310.
[8] A. Salehipour, K. Sorensen, P. Goos, O. Braysy, Efficient GRASP+VND and GRASP+VNS meta-heuristics for the traveling repairman problem, J. Oper. Res. 9 (2011), 189–209.
[9] M. Silva, A. Subramanian, T. Vidal, L. Ochi, A simple and effective metaheuristic for the minimum latency problem, J. Oper. Res. 221 (2012), 513–520.
[10] J.C. Picard, M. Queyranne, The time-dependent traveling salesman problem and its application to the tardiness problem in one machin scheduling, J. Oper. Res. 26 (1978), 86–110.
[11] K. Fox, Production Scheduling on Parallel Lines with Dependencies, Ph.D. Thesis, Johns Hopkins University, Baltimore, MD, USA, 1973.
[12] K. Fox, B. Gavish, S. Graves, The Time Dependent Traveling Salesman Problem and Extensions, Working paper No. 7904, Graduate School of Management,University of Rochester, Rochester, NY, USA, 1979.
[13] C. Malandraki, M. Daskin, Time dependent vehicle routing problems: formulations, properties and heuristic algorithms, J. Transp. Sci. 26 (1992), 185–200.
[14] A. Orda, R. Rom, Shortest-path and minimum delay algorithms in networks with time-dependent edge length, JACM. 37 (1990), 607–625.
[15] D. Simchi-Levi, O. Berman, Minimizing the total flow time of n jobs on a network, IEEE. Trans. 23 (1991), 236–244.
[16] N. Balakrishnan, A. Lucena, R.T. Wong, Scheduling examinations to reduce second-order conflicts, Comput. Oper. Res. 19 (1992), 353–361.
[17] L. Testa, A. Esterline, G. Dozier, Evolving efficient theme park tours, J. CIT. 7 (1999), 77–92.
[18] L. Bigras, M. Gamache, G. Savard, The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup time, J. Discrete Optim. 5 (2008), 685–699.
[19] R.V. Wiel, N. Sahinidis, An exact solution approach for the time-dependent traveling salesman problem, J. NRL. 43 (1996), 797–820.
[20] M.A. Figliozzi, The time dependent vehicle routing problem with time windows: benchmark problems, an efficient solution algorithm, and solution characteristics, J. Transp. Res. Part E Logist. Transp. Rev. 48 (2012), 616–636.
[21] K. Ruland, Polyhedral Solution to the Pickup and Delivery Problem, Ph.D. Thesis, Washington University, Saint Louis, MO, USA, 1995.
[22] J. Bentner, G. Bauer, G.M. Obermai, I. Morgensten, J. Schneider, Optimization of the time-dependent traveling salesman problem with Monte Carlo methods, Phys. Rev. E. 64 (2001), 36701-1–36701-8.
[23] V. Beiranvand, W. Hare, Y. Lucet, Best practices for comparing optimization algorithms, J. Optim. Eng. 18 (2017), 815–848.
[24] D.M. Vu, M. Hewitt, N. Boland, M. Savelsbergh, Time-dependent traveling salesman problem with time windows, J. Discret. Appl. Math. 261 (2019), 28–39.
[25] D.M. Vu, M. Hewitt, N. Boland, M. Savelsbergh, Dynamic discretization discovery for solving the time dependent traveling salesman problem with time windows, J. Transp. Sci. 54 (2020), 1–18.
[26] H. Sahina, B.Y. Karab, O.E. Karasanb, Debris removal during disaster response: a case for Turkey, J. Socio-Econ. Plan. Sci. 53 (2016), 49–59.
[27] N. Berktaş, B.Y. Kara, O.E. Karaşan, Solution methodologies for debris removal in disaster response, EURO J. Comput. Optim. 4 (2016), 403–445.
[28] H.B. Ban, An efficient two-phase metaheuristic algorithm for the time dependent traveling salesman problem, J. Rairo. 53 (2019), 917–935.
[29] J.K. Hao, Memetic algorithms in discrete optimization, in: F. Neri, C. Cotta, P. Moscato (Eds.), Handbook of Memetic Algorithms, Studies in Computational Intelligence, vol. 379, Springer, Berlin, Heidelberg, Germany, 2012.
[30] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, MA, USA, 1989.
[31] N. Mladenovic, P. Hansen, Variable neighborhood search, J. Oper. Res. 24 (1997), 1097–1100.
[32] T.A. Feo, M.G.C. Resende, Greedy randomized adaptive search procedures, J. Global Optim. 6 (1995), 109–133.
[33] O. Martin, S.W. Otto, E.W. Felten, Large-step Markov chains for the traveling salesman problem, J. Complex Syst. 5 (1991), 299–326.
[34] N.M. RazaliJohn, G.J. Geraghty, Genetic algorithm performance with different selection strategies in solving TSP, in Conference on ICCIIS'11, 2011, pp. 1134–1139.
[35] A. Otman, A. Jaafar, A comparative study of adaptive crossover operators for genetic algorithms to resolve the traveling salesman problem, J. Comput. Appl. 31 (2011), 49–57.
[36] Dataset for TDTSP: http://liris.cnrs.fr/christine.solnon/TDTSP.html
[37] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, J. Swarm Evol. Comput. 1 (2011), 3–18.
[38] E. Osaba, E. Onieva, R. Carballedo, F. Diaz, A. Perallos, An adaptive multi-crossover population algorithm for solving routing problems, in: G. Terrazas, F. Otero, A. Masegosa (Eds.), Nature Inspired Cooperative Strategies for Optimization (NICSO 2013), Studies in Computational Intelligence, vol. 512, Springer, Cham, Switzerland, 2013, pp. 113–125.
[39] H. Garg, A hybrid GSA-GA algorithm for constrained optimization problems, J. Inf. Sci. 478 (2019), 499–523.

[40] H. Garg, A hybrid PSO-GA algorithm for constrained optimization problems, Appl. Math. Comput. 274 (2016), 292–305.
[41] H. Garg, A hybrid GA-GSA algorithm for optimizing the performance of an industrial system by utilizing uncertain data, in: P. Vasant (Ed.), Handbook of Research on Artificial Intelligence Techniques and Algorithms, IGIGlobal Publisher, 2015.

APPENDIX

Table A1 The experimental results for TDTSP-PD with INST-30-x instances that are proposed by [25] (*LES* = 1).

Instances	Travel time function 1							Travel time function 2							Travel time function 3						
	UB	*TS+VNS*		MA				*UB*	*TS+VNS*		MA				*UB*	*TS+VNS*		MA			
		Best. Sol	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	Gap [%]	*T*
INST-30-1	21150	20045	-5.22	19588	19588	-7.39	4.10	20745	20365	-1.83	19325	19325	-6.85	4.12	20932	19925	-4.81	18810	18810	-10.14	4.15
INST-30-2	20444	18003	-11.94	17482	17482	-14.49	4.09	18456	17443	-5.49	17682	17682	-4.19	4.20	19573	17482	-10.68	17308	17308	-11.57	4.15
INST-30-3	24495	21275	-13.15	20707	20707	-15.46	4.18	23383	20047	-14.27	19704	19704	-15.73	4.10	21637	19818	-8.41	19343	19343	-10.60	4.11
INST-30-4	21054	18420	-12.51	18093	18093	-14.06	4.09	20326	17800	-12.43	17655	17655	-13.14	4.09	20161	17622	-12.59	17489	17489	-13.25	4.10
INST-30-5	18815	16681	-11.34	16643	16643	-11.54	4.12	19515	16907	-13.36	16611	16611	-14.88	4.12	18247	16992	-6.88	16379	16379	-10.24	4.18
INST-30-6	21166	19237	-9.11	18649	18649	-11.89	4.19	20838	18879	-9.40	18293	18293	-12.21	4.11	20369	18334	-9.99	17991	17991	-11.67	4.14
INST-30-7	18168	17338	-4.57	17338	17338	-4.57	4.17	18592	17216	-7.40	17360	17360	-6.63	4.11	20167	17104	-15.19	17161	17161	-14.91	4.15
INST-30-8	21688	19069	-12.08	19069	19069	-12.08	4.13	20095	18746	-6.71	18746	18746	-6.71	4.15	19844	18076	-8.91	18076	20949	-8.91	4.12
INST-30-9	19305	17839	-7.59	17839	18833	-7.59	4.09	18709	17868	-4.50	17868	18476	-4.50	4.12	19442	16701	-14.10	16701	16701	-14.10	4.11
INST-30-10	20204	17900	-11.40	17900	17900	-11.40	4.08	19476	17422	-10.55	17422	17422	-10.55	4.18	19295	17544	-9.07	17544	17544	-9.07	4.11
INST-30-11	21576	19816	-8.16	20863	20863	-3.30	4.18	21409	19325	-9.73	19325	19325	-9.73	4.12	21251	20632	-2.91	19290	19290	-9.23	4.17
INST-30-12	25598	19900	-22.26	19900	19900	-22.26	4.08	21504	21191	-1.46	21191	21191	-1.46	4.20	21734	20956	-3.58	19034	19034	-12.42	4.12
INST-30-13	23356	20233	-13.37	20233	20233	-13.37	4.09	21301	21340	0.18	21340	21340	0.18	4.17	20189	24495	21.33	19279	19279	-4.51	4.12
INST-30-14	24970	22303	-10.68	22303	22303	-10.68	4.14	24581	21350	-13.14	21350	21350	-13.14	4.19	23153	18485	-20.16	21107	21107	-8.84	4.18
INST-30-15	19451	18034	-7.28	18034	18034	-7.28	4.10	17788	17537	-1.41	17537	17537	-1.41	4.19	17789	17409	-2.14	17409	17409	-2.14	4.19
INST-30-16	17432	17309	-0.71	16807	16807	-3.59	4.18	17282	17123	-0.92	16807	16807	-2.75	4.10	17091	16949	-0.83	16807	16807	-1.66	4.14
INST-30-17	20467	19919	-2.68	19156	19156	-6.41	4.15	21701	19528	-10.01	19390	19390	-10.65	4.12	20778	18971	-8.70	19236	19236	-7.42	4.08
INST-30-18	19026	18031	-5.23	17847	17847	-6.20	4.09	18890	17785	-5.85	18157	18157	-3.88	4.16	18709	17595	-5.95	17595	18121	-5.95	4.16
INST-30-19	23741	21599	-9.02	21440	21440	-9.69	4.10	23533	21351	-9.27	21516	21516	-8.57	4.09	22533	20398	-9.47	20398	21446	-9.47	4.13
INST-30-20	24452	19637	-19.69	19559	19559	-20.01	4.08	23412	19331	-17.43	19331	19331	-17.43	4.20	20929	18869	-9.84	19527	19527	-6.70	4.17
aver			-9.90			-10.66	4.12			-7.75			-8.21	4.14			-7.14			-9.14	4.14

LES, levels of earthquake severity; MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; UB, upper bound; VNS, Variable Neighborhood Search.

Table A2 The experimental results for TDTSP-PD with INST-30-x instances that are proposed by [25] (*LES* = 2).

Instances	Travel time function 1							Travel time function 2							Travel time function 3						
	UB	*TS+VNS*		MA				*UB*	*TS+VNS*		MA				*UB*	*TS+VNS*		MA			
		Best. Sol	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	Gap [%]	*T*
INST-30-1	31640	26968	-14.77	26968	26968	-14.77	4.11	30507	25578	-16.16	26322	26322	-13.72	4.16	30114	24301	-19.30	25714	25714	-14.61	4.18
INST-30-2	27108	23405	-13.66	23405	23405	-13.66	4.17	25853	22201	-14.13	22901	22901	-11.42	4.10	25313	21243	-16.08	21728	21728	-14.16	4.10
INST-30-3	21028	20480	-2.61	20480	20480	-2.61	4.14	20515	19737	-3.79	18490	18490	-9.87	4.13	19110	19111	0.01	18050	18050	-5.55	4.16
INST-30-4	31322	27816	-11.19	27816	27816	-11.19	4.12	28213	26137	-7.36	26449	26449	-6.25	4.13	26952	25147	-6.70	24778	24778	-8.07	4.14
INST-30-5	19081	19254	0.91	19254	19254	0.91	4.09	18095	18875	4.31	17036	17036	-5.85	4.09	18246	18253	0.04	16779	16779	-8.04	4.18
INST-30-6	20564	21934	6.66	21934	21934	6.66	4.08	19863	21042	5.94	19626	19626	-1.19	4.17	19754	20422	3.38	19042	19042	-3.60	4.13
INST-30-7	26113	24549	-5.99	24549	24549	-5.99	4.13	25800	23322	-9.60	22675	22675	-12.11	4.11	29294	22579	-22.92	22177	22177	-24.30	4.17
INST-30-8	39424	31111	-21.09	31111	31111	-21.09	4.18	31733	28718	-9.50	25092	25092	-20.93	4.11	30050	27271	-9.25	23894	23894	-20.49	4.12
INST-30-9	28348	23575	-16.84	23575	23575	-16.84	4.18	26467	22534	-14.86	23117	23117	-12.66	4.15	26923	22046	-18.11	22350	22350	-16.99	4.12
INST-30-10	33256	29238	-12.08	29238	29238	-12.08	4.17	29964	28287	-5.60	26615	26615	-11.18	4.13	29147	26848	-7.89	25381	25381	-12.92	4.11
INST-30-11	36049	33322	-7.56	33322	33322	-7.56	4.10	34261	30794	-10.12	27901	27901	-18.56	4.19	31529	29482	-6.49	26679	26679	-15.38	4.15
INST-30-12	36156	30818	-14.76	30818	30818	-14.76	4.17	33794	28832	-14.68	26431	26431	-21.79	4.19	32857	27136	-17.41	25557	25557	-22.22	4.19
INST-30-13	26237	29278	11.59	29278	29278	11.59	4.10	24175	28048	16.02	21514	21514	-11.01	4.14	22624	26183	15.73	20613	20613	-8.89	4.17
INST-30-14	25462	32904	29.23	32904	32904	29.23	4.17	24855	30520	22.79	21349	21349	-14.11	4.11	23304	29244	25.49	20766	20766	-10.89	4.09
INST-30-15	30904	30558	-1.12	30558	30558	-1.12	4.09	29040	28800	-0.83	23599	23599	-18.74	4.09	26447	26893	1.69	23118	23118	-12.59	4.18
INST-30-16	25881	21875	-15.48	21875	21875	-15.48	4.16	24902	22457	-9.82	21642	21642	-13.09	4.18	24340	22333	-8.25	21115	21115	-13.25	4.10
INST-30-17	20468	22796	11.37	22796	22796	11.37	4.18	22460	22684	1.00	20309	20309	-9.58	4.13	20296	22684	11.77	19939	19939	-1.76	4.14
INST-30-18	25052	22931	-8.47	22931	22931	-8.47	4.17	23857	22585	-5.33	21025	21025	-11.87	4.14	23250	22585	-2.86	20729	20729	-10.84	4.08
INST-30-19	23923	23401	-2.18	23401	23401	-2.18	4.13	23453	23401	-0.22	21198	21198	-9.61	4.20	22151	23509	6.13	20853	20853	-5.86	4.19
INST-30-20	29393	23629	-19.61	23629	23629	-19.61	4.14	27542	23671	-14.05	22588	22588	-17.99	4.16	23609	23587	-0.09	22311	22311	-5.50	4.15
aver			-5.38			-5.38	4.14			-4.30			-12.58	4.14			-3.56			-11.79	4.14

LES, levels of earthquake severity; MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; UB, upper bound; VNS, Variable Neighborhood Search.

Table A3 The experimental results for TDTSP-PD with INST-30-x instances that are proposed by [25] (*LES* = 3).

Instances	Travel time function 1							Travel time function 2							Travel time function 3						
	UB	*TS+VNS*		MA				*UB*	*TS+VNS*		MA				*UB*	*TS+VNS*		MA			
		Best. Sol	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*
INST-30-1	33185	26892	-18.96	28141	28141	-15.20	4.25	31889	25513	-19.99	26711	26711	-16.24	4.28	31564	25872	-18.03	25872	25872	-18.03	4.27
INST-30-2	59288	34241	-42.25	35950	35950	-39.36	4.28	46307	32232	-30.39	33595	33595	-27.45	4.27	40541	31690	-21.83	31690	31690	-21.83	4.26
INST-30-3	54934	34737	-36.77	34137	34137	-37.86	4.25	49895	31919	-36.03	32484	32484	-34.90	4.26	40181	30257	-24.70	30257	30257	-24.70	4.26
INST-30-4	41389	33746	-18.47	33181	33181	-19.83	4.27	38859	31680	-18.47	29884	29884	-23.10	4.27	37426	28685	-23.36	28685	28685	-23.36	4.26
INST-30-5	16321	22925	40.46	15946	15946	-2.30	4.27	17244	22124	28.30	16102	16102	-6.62	4.26	17075	15887	-6.96	14776	14776	-13.46	4.25
INST-30-6	48387	45100	-6.79	42375	42375	-12.42	4.28	44945	39653	-11.77	38160	38160	-15.10	4.25	43446	34927	-19.61	16898	16898	-61.11	4.25
INST-30-7	25852	31283	21.01	23380	23380	-9.56	4.25	24547	28552	16.32	22010	22010	-10.34	4.26	28516	21322	-25.23	21322	21322	-25.23	4.28
INST-30-8	62116	39813	-35.91	37415	37415	-39.77	4.26	46043	36750	-20.18	32943	32943	-28.45	4.25	43809	30921	-29.42	30921	30921	-29.42	4.28
INST-30-9	27310	28731	5.20	23650	23650	-13.40	4.26	25301	26904	6.34	22563	22563	-10.82	4.26	25961	21392	-17.60	21392	21392	-17.60	4.27
INST-30-10	36706	41188	12.21	32220	32220	-12.22	4.27	33248	38609	16.12	30251	30251	-9.01	4.26	32098	28910	-9.93	28910	28910	-9.93	4.25
INST-30-11	35545	34313	-3.47	29816	29816	-16.12	4.26	32754	31401	-4.13	28124	28124	-14.14	4.26	30994	26606	-14.16	26606	26606	-14.16	4.26
INST-30-12	35640	32625	-8.46	27421	27421	-23.06	4.28	30406	29907	-1.64	26154	26154	-13.98	4.25	29162	24743	-15.15	16375	16375	-43.85	4.26
INST-30-13	35477	41382	16.64	26877	26877	-24.24	4.26	31819	38019	19.49	24922	24922	-21.68	4.28	28511	24541	-13.92	24541	24541	-13.92	4.27
INST-30-14	32519	41505	27.63	30040	30040	-7.62	4.26	32417	38218	17.89	27803	27803	-14.23	4.28	30741	27174	-11.60	17938	17938	-41.65	4.25
INST-30-15	37690	39910	5.89	30454	30454	-19.20	4.25	32025	36354	13.52	27597	27597	-13.83	4.26	29262	27268	-6.81	27268	27268	-6.81	4.25
INST-30-16	35303	27959	-20.80	28636	28636	-18.89	4.25	33176	26283	-20.78	28230	28230	-14.91	4.26	31700	26288	-17.07	25329	25329	-20.10	4.26
INST-30-17	34271	34955	2.00	32686	32686	-4.62	4.28	39106	32159	-17.76	30244	30244	-22.66	4.26	32161	28939	-10.02	30575	30575	-4.93	4.27
INST-30-18	42801	38335	-10.43	34554	34554	-19.27	4.27	40231	34827	-13.43	31952	31952	-20.58	4.28	37578	30356	-19.22	33680	33680	-10.37	4.27
INST-30-19	44979	42853	-4.73	37131	37131	-17.45	4.27	42031	38276	-8.93	33944	33944	-19.24	4.26	38646	32373	-16.23	36221	36221	-6.27	4.27
INST-30-20	57114	39436	-30.95	40028	40028	-29.92	4.25	52133	36309	-30.35	37490	37490	-28.09	4.25	39098	35099	-10.23	33869	33869	-13.37	4.26
aver			-5.35			-19.12	4.26			-5.80			-18.27	4.26			-16.55			-21.01	4.26

LES, levels of earthquake severity; MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; UB, upper bound; VNS, Variable Neighborhood Search.

Table A4 The experimental results for TDTSP-PD with INST-30-x instances that are proposed by [25] (*LES* = 4).

Instances	Travel time function 1							Travel time function 2							Travel time function 3						
	UB	*TS+VNS*		MA				*UB*	*TS+VNS*		MA				*UB*	*TS+VNS*		MA			
		Best. Sol	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*
INST-30-1	83854	51839	-38.18	57198	57198	-31.79	4.27	75845	51326	-32.33	58892	58892	-22.35	4.27	76506	48619	-36.45	56726	56726	-25.85	4.26
INST-30-2	63815	45384	-28.88	48190	48190	-24.48	4.26	68041	41222	-39.42	44144	44144	-35.12	4.26	64013	38419	-39.98	44603	44603	-30.32	4.28
INST-30-3	20789	23689	13.95	18577	18577	-10.64	4.27	20320	22652	11.48	18503	18503	-8.94	4.27	19459	22216	14.17	18308	18308	-5.92	4.26
INST-30-4	26174	24733	-5.51	22956	22956	-12.29	4.26	24246	23189	-4.36	21842	21842	-9.92	4.27	24549	22987	-6.36	21146	21146	-13.86	4.26
INST-30-5	33755	28398	-15.87	28530	28530	-15.48	4.27	37617	26908	-28.47	27731	27731	-26.28	4.26	35438	26221	-26.01	26522	26522	-25.16	4.28
INST-30-6	61298	47161	-23.06	46412	46412	-24.28	4.26	57090	42602	-25.38	41102	41102	-28.00	4.28	52956	42456	-19.83	36745	36745	-30.61	4.28
INST-30-7	48248	49093	1.75	44331	44331	-8.12	4.26	54843	46105	-15.93	41020	41020	-25.20	4.28	64375	44645	-30.65	38141	38141	-40.75	4.26
INST-30-8	65341	48539	-25.71	46893	46893	-28.23	4.27	62304	48003	-22.95	42335	42335	-32.05	4.27	58365	47674	-18.32	41071	41071	-29.63	4.25
INST-30-9	48060	47194	-1.80	43840	43840	-8.78	4.27	46746	43097	-7.81	40170	40170	-14.07	4.27	57141	40051	-29.91	37150	37150	-34.99	4.26
INST-30-10	39058	44001	12.66	32523	32523	-16.73	4.25	35051	41247	17.68	29398	29398	-16.13	4.27	34979	40142	14.76	28521	28521	-18.46	4.26
INST-30-11	56832	48493	-14.67	39638	39638	-30.25	4.28	57627	45246	-21.48	39249	39249	-31.89	4.26	56932	43405	-23.76	34788	34788	-38.90	4.27
INST-30-12	29270	45512	55.49	23053	23053	-21.24	4.27	31117	42734	37.33	21991	21991	-29.33	4.26	30734	37764	22.87	21450	21450	-30.21	4.26
INST-30-13	33281	48384	45.38	26426	26426	-20.60	4.26	30236	46347	53.28	25338	25338	-16.20	4.26	29132	41182	41.36	24845	24845	-14.72	4.27
INST-30-14	25544	49422	93.48	23267	23267	-8.91	4.26	25311	48254	90.64	22253	22253	-12.08	4.26	24688	47367	91.86	21470	21470	-13.03	4.27
INST-30-15	31720	45500	43.44	25930	25930	-18.25	4.26	28252	43052	52.39	24901	24901	-11.86	4.28	29278	39544	35.06	23644	23644	-19.24	4.26
INST-30-16	32106	25059	-21.95	25755	25755	-19.78	4.26	31334	23960	-23.53	24479	24479	-21.88	4.26	30697	22833	-25.62	23262	23262	-24.22	4.25
INST-30-17	61796	17017	-72.46	52369	52369	-15.26	4.27	66994	47280	-29.43	55100	55100	-17.75	4.26	49436	45320	-8.33	47391	47391	-4.14	4.26
INST-30-18	61184	17120	-72.02	48495	48495	-20.74	4.27	50560	46954	-7.13	44386	44386	-12.21	4.26	52476	44925	-14.39	39267	39267	-25.17	4.26
INST-30-19	67994	18397	-72.94	52995	52995	-22.06	4.27	66309	53260	-19.68	51783	51783	-21.91	4.26	70097	48996	-30.10	47889	47889	-31.68	4.26
INST-30-20	47320	16960	-64.16	38438	38438	-18.77	4.27	45206	35787	-20.84	35476	35476	-21.52	4.26	39512	33932	-14.12	34184	34184	-13.48	4.27
aver			-9.55			-18.84	4.27			-1.80			-20.74	4.26			-5.19			-23.52	4.26

LES, levels of earthquake severity; MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; UB, upper bound; VNS, Variable Neighborhood Search.

Table A5 The experimental results for TDTSP-PD with INST-30-x instances that are proposed by [25] (*LES* = 5).

Instances	Travel time function 1							Travel time function 2							Travel time function 3						
	UB	*TS+VNS*		MA				*UB*	*TS+VNS*		MA				*UB*	*TS+VNS*		MA			
		Best. Sol	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*
INST-30-1	107317	66992	-37.58	74442	74442	-30.63	4.28	103333	64492	-37.59	71413	71413	-30.89	4.27	104827	67325	-35.78	63684	63684	-39.25	4.27
INST-30-2	65777	42675	-35.12	40267	40267	-38.78	4.30	64588	40471	-37.34	35461	35461	-45.10	4.36	66588	34378	-48.37	37150	37150	-44.21	4.34
INST-30-3	95476	60840	-36.28	65251	65251	-31.66	4.35	85773	58057	-32.31	61377	61377	-28.44	4.36	72614	60113	-17.22	59004	59004	-18.74	4.32
INST-30-4	21941	24209	10.34	18867	18867	-14.01	4.27	19665	23083	17.38	18347	18347	-6.70	4.35	19845	17955	-9.52	22562	22562	13.69	4.32
INST-30-5	31391	36593	16.57	27574	27574	-12.16	4.36	32930	34022	3.32	25958	25958	-21.17	4.28	31731	25359	-20.08	32287	32287	1.75	4.36
INST-30-6	83472	62040	-25.68	67676	67676	-18.92	4.34	76704	62104	-19.03	64423	64423	-16.01	4.30	73162	62765	-14.21	61901	61901	-15.39	4.33
INST-30-7	52247	51909	-0.65	45190	45190	-13.51	4.32	48283	49860	3.27	39921	39921	-17.32	4.30	67849	38560	-43.17	49916	49916	-26.43	4.33
INST-30-8	86827	62656	-27.84	68065	68065	-21.61	4.33	88741	58778	-33.76	62986	62986	-29.02	4.34	89191	61960	-30.53	59450	59450	-33.35	4.36
INST-30-9	71985	59531	-17.30	59545	59545	-17.28	4.29	67116	56185	-16.29	55532	55532	-17.26	4.28	68130	56755	-16.70	54269	54269	-20.34	4.35
INST-30-10	85009	61594	-27.54	66442	66442	-21.84	4.32	78913	61360	-22.24	65710	65710	-16.73	4.34	79309	61594	-22.34	61764	61764	-22.12	4.33
INST-30-11	79417	63816	-19.64	63597	63597	-19.92	4.37	76627	59418	-22.46	60469	60469	-21.09	4.28	68611	61145	-10.88	59028	59028	-13.97	4.29
INST-30-12	92764	62967	-32.12	69158	69158	-25.45	4.32	92878	61161	-34.15	67442	67442	-27.39	4.34	87700	67208	-23.37	60429	60429	-31.10	4.29
INST-30-13	98230	69898	-28.84	65739	65739	-33.08	4.32	91748	65071	-29.08	68377	68377	-25.47	4.32	82766	64957	-21.52	64213	64213	-22.42	4.36
INST-30-14	95962	69949	-27.11	62137	62137	-35.25	4.29	82241	67698	-17.68	61491	61491	-25.23	4.35	87850	61064	-30.49	67392	67392	-23.29	4.27
INST-30-15	48214	59664	23.75	39450	39450	-18.18	4.32	43296	59682	37.85	34727	34727	-19.79	4.34	50920	32381	-36.41	59638	59638	17.12	4.32
INST-30-16	51502	39822	-22.68	43980	43980	-14.61	4.33	45614	34609	-24.13	36772	36772	-19.38	4.36	43104	35108	-18.55	32880	32880	-23.72	4.29
INST-30-17	85508	64119	-25.01	63382	63382	-25.88	4.34	80281	62116	-22.63	65077	65077	-18.94	4.36	79535	63282	-20.44	61698	61698	-22.43	4.37
INST-30-18	31955	42257	32.24	26321	26321	-17.63	4.31	31128	37613	20.83	24397	24397	-21.62	4.30	30182	23852	-20.97	35252	35252	16.80	4.34
INST-30-19	104494	64539	-38.24	64925	64925	-37.87	4.31	105399	61818	-41.35	63663	63663	-39.60	4.34	87513	63056	-27.95	60744	60744	-30.59	4.32
INST-30-20	82712	65307	-21.04	64914	64914	-21.52	4.37	75155	61124	-18.67	62619	62619	-16.68	4.29	69422	61475	-11.45	59414	59414	-14.42	4.32
aver			-16.99			-23.49	4.32			-16.30			-23.19	4.32			-24.00			-17.62	4.32

LES, levels of earthquake severity; MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; UB, upper bound; VNS, Variable Neighborhood

Table A6 The experimental results for TDTSP-PD with INST-50-x instances that are proposed by [25] (*LES* = 1).

Instances	Travel time function 1							Travel time function 2							Travel time function 3						
	UB	*TS+VNS*		MA				*UB*	*TS+VNS*		MA				*UB*	*TS+VNS*		MA			
		Best. Sol	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*
INST-50-1	25779	23320	-9.54	24551	24551	-4.76	68.36	25125	22751	-9.45	23945	23945	-4.70	69.04	25230	22414	-11.16	23263	23263	-7.80	73.71
INST-50-2	28809	25221	-12.45	27090	27090	-5.97	72.09	27097	24280	-10.40	26301	26301	-2.94	70.35	26523	23469	-11.51	25610	25610	-3.44	73.52
INST-50-3	34099	26327	-22.79	28186	28186	-17.34	68.25	29412	24974	-15.09	27044	27044	-8.05	72.99	30642	24457	-20.18	25318	25318	-17.37	68.32
INST-50-4	32195	27924	-13.27	29102	29102	-9.61	68.43	30136	26674	-11.49	28232	28232	-6.32	72.82	28867	25818	-10.56	27089	27089	-6.16	72.43
INST-50-5	26708	24793	-7.17	24451	24451	-8.45	71.13	24474	23876	-2.44	23190	23190	-5.25	68.36	24347	23001	-5.53	22729	22729	-6.65	69.61
INST-50-6	24252	24372	0.49	23986	23986	-1.10	68.58	23254	23758	2.17	23094	23094	-0.69	70.40	22725	23113	1.71	22562	22562	-0.72	70.54
INST-50-7	31853	28027	-12.01	26495	26495	-16.82	72.91	29361	26994	-8.06	26058	26058	-11.25	71.16	27690	26051	-5.92	25613	25613	-7.50	71.29
INST-50-8	30485	27602	-9.46	28203	28203	-7.49	72.91	30286	26149	-13.66	26933	26933	-11.07	70.50	29138	25335	-13.05	26173	26173	-10.18	73.66
INST-50-9	28425	26559	-6.56	24972	24972	-12.15	72.33	25375	25457	0.32	24298	24298	-4.24	71.94	24733	24756	0.09	23420	23420	-5.31	70.51
INST-50-10	28494	28984	1.72	27926	27926	-1.99	68.90	27205	27476	1.00	26117	26117	-4.00	71.77	27771	26786	-3.55	26082	26082	-6.08	73.90
INST-50-11	28407	30068	5.85	26010	26010	-8.44	71.96	27424	28490	3.89	25136	25136	-8.34	69.75	27193	27040	-0.56	24381	24381	-10.34	69.81
INST-50-12	30917	30324	-1.92	27982	27982	-9.49	71.11	30002	29396	-2.02	26782	26782	-10.73	70.59	27801	27878	0.28	26186	26186	-5.81	72.21
INST-50-13	27118	26920	-0.73	26784	26784	-1.23	73.84	25589	26052	1.81	25107	25107	-1.88	68.09	25262	24892	-1.46	24869	24869	-1.56	72.00
INST-50-14	28202	27788	-1.47	25240	25240	-10.50	71.89	26933	26706	-0.84	24380	24380	-9.48	73.90	25200	25971	3.06	24264	24264	-3.71	71.23
INST-50-15	27584	28667	3.93	26529	26529	-3.82	72.80	26881	27023	0.53	26073	26073	-3.01	69.00	26708	26158	-2.06	25468	25468	-4.64	72.19
INST-50-16	31522	27089	-14.06	29051	29051	-7.84	70.72	30465	27395	-10.08	27235	27235	-10.60	68.64	29683	27519	-7.29	27262	27262	-8.16	72.00
INST-50-17	27527	26373	-4.19	26285	26285	-4.51	70.59	25679	26679	3.89	25113	25113	-2.20	70.23	25093	26592	5.97	24550	24550	-2.16	69.07
INST-50-18	29002	24961	-13.93	25747	25747	-11.22	72.95	26979	25531	-5.37	24557	24557	-8.98	69.19	28002	25713	-8.17	25104	25104	-10.35	68.77
INST-50-19	32966	30349	-7.94	29709	29709	-9.88	68.50	29004	30714	5.90	27951	27951	-3.63	70.94	30129	30715	1.94	27457	27457	-8.87	73.99
INST-50-20	25944	25495	-1.73	25841	25841	-0.40	68.80	26645	25189	-5.46	24321	24321	-8.72	70.04	26627	25564	-3.99	25024	25024	-6.02	69.03
aver			-6.36			-7.65	70.85			-3.74			-6.30	70.49			-4.60			-6.64	71.39

LES, levels of earthquake severity; MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; UB, upper bound; VNS, Variable Neighborhood

Table A7 The experimental results for TDTSP-PD with INST-50-x instances that are proposed by [25] (*LES* = 2).

Instances	Travel time function 1							Travel time function 2							Travel time function 3						
	UB	*TS+VNS*		MA				*UB*	*TS+VNS*		MA				*UB*	*TS+VNS*		MA			
		Best. Sol	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*
INST-50-1	27081	24668	-8.91	25485	25485	-5.89	68.20	26218	23979	-8.54	25370	25370	-3.23	69.51	25628	23388	-8.74	24948	24948	-2.65	70.54
INST-50-2	29435	26736	-9.17	27898	27898	-5.22	71.37	27743	25857	-6.80	26967	26967	-2.80	69.74	27029	25103	-7.13	26177	26177	-3.15	68.57
INST-50-3	38328	31972	-16.58	32493	32493	-15.22	73.29	34521	30157	-12.64	30645	30645	-11.23	71.70	34372	29376	-14.54	29164	29164	-15.15	71.59
INST-50-4	35739	31613	-11.54	34384	34384	-3.79	72.02	33051	29876	-9.61	31653	31653	-4.23	69.59	33234	29152	-12.28	30652	30652	-7.77	70.83
INST-50-5	28103	28670	2.02	26851	26851	-4.46	69.14	26414	27464	3.98	25568	25568	-3.20	72.95	26339	26659	1.21	25307	25307	-3.92	72.18
INST-50-6	43371	37458	-13.63	38827	38827	-10.48	70.21	39997	34695	-13.26	34903	34903	-12.74	73.90	37061	32719	-11.72	33392	33392	-9.90	72.20
INST-50-7	34353	33031	-3.85	28994	28994	-15.60	70.76	30494	31319	2.71	27301	27301	-10.47	72.38	28654	30168	5.28	26767	26767	-6.59	71.83
INST-50-8	37225	30803	-17.25	30058	30058	-19.25	73.89	34802	29042	-16.55	28625	28625	-17.75	70.06	31144	27836	-10.62	27802	27802	-10.73	68.20
INST-50-9	42751	33303	-22.10	33455	33455	-21.74	68.94	33348	31300	-6.14	29875	29875	-10.41	71.50	33334	30375	-8.88	30200	30200	-9.40	68.41
INST-50-10	25497	31104	21.99	25227	25227	-1.06	73.13	24188	29228	20.84	23843	23843	-1.43	68.65	24674	28482	15.43	24109	24109	-2.29	69.92
INST-50-11	29270	32157	9.86	26644	26644	-8.97	71.87	27439	30570	11.41	26274	26274	-4.25	73.44	27254	29138	6.91	25005	25005	-8.25	71.19
INST-50-12	41039	34804	-15.19	33418	33418	-18.57	70.26	36029	32236	-10.53	31780	31780	-11.79	73.28	34259	30972	-9.59	31094	31094	-9.24	71.93
INST-50-13	28380	33225	17.07	27358	27358	-3.60	69.15	27084	31576	16.59	25138	25138	-7.19	72.91	27553	30277	9.89	25698	25698	-6.73	70.45
INST-50-14	41770	38953	-6.74	33825	33825	-19.02	70.57	39214	35662	-9.06	32105	32105	-18.13	69.56	33900	34056	0.46	30866	30866	-8.95	72.92
INST-50-15	28599	35664	24.70	27159	27159	-5.04	70.89	29352	33016	12.48	26494	26494	-9.74	71.57	27699	31487	13.68	26053	26053	-5.94	72.31
INST-50-16	38620	32954	-14.67	35015	35015	-9.33	68.72	37823	35829	-5.27	32519	32519	-14.02	68.14	34968	35942	2.79	31162	31162	-10.88	73.81
INST-50-17	44963	37625	-16.32	36843	36843	-18.06	71.54	41853	44310	5.87	34599	34599	-17.33	70.55	37576	44394	18.14	33966	33966	-9.61	71.19
INST-50-18	32965	30030	-8.90	27904	27904	-15.35	69.36	31501	33283	5.66	26851	26851	-14.76	69.88	30788	33147	7.66	25623	25623	-16.78	69.95
INST-50-19	54707	44461	-18.73	48319	48319	-11.68	70.31	53126	44831	-15.61	43283	43283	-18.53	68.97	51656	44624	-13.61	40848	40848	-20.92	68.63
INST-50-20	32357	35878	10.88	32083	32083	-0.85	71.50	32349	35919	11.04	30485	30485	-5.76	69.07	33080	35421	7.08	30260	30260	-8.52	71.67
aver			-4.85			-10.66	70.76			-1.17			-9.95	70.87			-0.43			-8.87	70.91

LES, levels of earthquake severity; MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; UB, upper bound; VNS, Variable Neighborhood Search.

Table A8 The experimental results for TDTSP-PD with INST-50-x instances that are proposed by [25] (*LES* = 3).

Instances	Travel time function 1							Travel time function 2							Travel time function 3						
	UB	*TS+VNS*		MA				*UB*	*TS+VNS*		MA				*UB*	*TS+VNS*		MA			
		Best. Sol	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*
INST-50-1	36960	31643	-14.39	29433	29433	-20.37	72.67	32216	29156	-9.50	27679	27679	-14.08	69.53	31653	26877	-15.09	28615	28615	-9.60	69.06
INST-50-2	44799	36853	-17.74	35806	35806	-20.07	70.54	36041	35285	-2.10	33266	33266	-7.70	69.34	35939	31743	-11.68	32831	32831	-8.65	72.33
INST-50-3	86836	56623	-34.79	56623	56623	-34.79	68.54	71395	51036	-28.52	48599	48599	-31.93	72.01	81496	46119	-43.41	53650	53650	-34.17	70.84
INST-50-4	33216	29183	-12.14	29183	29183	-12.14	69.60	31038	28462	-8.30	28462	28462	-8.30	73.07	30488	27514	-9.75	27514	27514	-9.75	68.92
INST-50-5	63739	46929	-26.37	43905	43905	-31.12	68.92	49645	40830	-17.76	40898	40898	-17.62	70.07	67433	39295	-41.73	39927	39927	-40.79	70.05
INST-50-6	40960	38696	-5.53	38696	38696	-5.53	69.69	41376	36899	-10.82	36899	36899	-10.82	72.68	38466	34771	-9.61	34771	34771	-9.61	71.64
INST-50-7	43468	34992	-19.50	34992	34992	-19.50	70.64	38120	33634	-11.77	33634	33634	-11.77	72.05	35064	32256	-8.01	32256	32256	-8.01	69.15
INST-50-8	74503	55013	-26.16	54112	54112	-27.37	71.16	66960	54181	-19.08	51094	51094	-23.69	68.04	60719	48383	-20.32	47246	47246	-22.19	72.43
INST-50-9	68916	47903	-30.49	53270	53270	-22.70	70.74	50785	41549	-18.19	51004	51004	0.43	71.61	52824	48994	-7.25	39340	39340	-25.53	69.46
INST-50-10	31497	27851	-11.58	27851	27851	-11.58	73.25	30516	26638	-12.71	26638	26638	-12.71	70.32	28878	26446	-8.42	26446	26446	-8.42	73.50
INST-50-11	46739	43923	-6.02	43923	43923	-6.02	71.11	46428	41265	-11.12	41265	41265	-11.12	73.50	43495	38552	-11.36	38552	38552	-11.36	69.61
INST-50-12	41242	32576	-21.01	32576	32576	-21.01	73.66	36294	31392	-13.51	31392	31392	-13.51	68.01	34620	30125	-12.98	30125	30125	-12.98	72.59
INST-50-13	37942	33945	-10.53	33945	33945	-10.53	71.83	35034	32093	-8.39	32093	32093	-8.39	70.77	34217	30822	-9.92	30822	30822	-9.92	69.13
INST-50-14	41402	34996	-15.47	34996	34996	-15.47	73.75	39235	31904	-18.68	31904	31904	-18.68	70.55	36709	32148	-12.42	32148	32148	-12.42	69.72
INST-50-15	28022	27835	-0.67	27835	27835	-0.67	69.44	28296	26703	-5.63	26703	26703	-5.63	70.77	28635	25947	-9.39	25947	25947	-9.39	68.55
INST-50-16	72915	62848	-13.81	54582	54582	-25.14	72.06	72863	61662	-15.37	59288	59288	-18.63	72.62	79089	58188	-26.43	60254	60254	-23.81	71.46
INST-50-17	74497	57897	-22.28	55255	55255	-25.83	69.73	69266	57268	-17.32	57693	57693	-16.71	69.93	69845	57627	-17.49	53220	53220	-23.80	72.10
INST-50-18	29162	26652	-8.61	26652	26652	-8.61	72.03	27071	25093	-7.31	25093	25093	-7.31	72.71	28437	25321	-10.96	25321	25321	-10.96	71.28
INST-50-19	66066	52259	-20.90	49722	49722	-24.74	72.17	65210	48449	-25.70	49517	49517	-24.07	70.83	56196	50586	-9.98	49765	49765	-11.44	70.55
INST-50-20	38985	33368	-14.41	43638	43638	11.94	68.41	35817	31335	-12.51	31335	31335	-12.51	68.21	36287	30602	-15.67	30602	30602	-15.67	71.87
aver			-16.62			-16.56	71.00			-13.71			-13.74	70.83			-15.59			-15.92	70.71

LES, levels of earthquake severity; MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; UB, upper bound; VNS, Variable Neighborhood Search.

Table A9 The experimental results for TDTSP-PD with INST-50-x instances that are proposed by [25] (*LES* = 4).

Instances	Travel time function 1							Travel time function 2							Travel time function 3						
	UB	*TS+VNS*		MA				*UB*	*TS+VNS*		MA				*UB*	*TS+VNS*		MA			
		Best. Sol	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*
INST-50-1	90721	59708	-34.19	64111	64111	-29.33	71.89	80178	58215	-27.39	63723	63723	-20.52	71.49	78833	57996	-26.43	63604	63604	-19.32	70.43
INST-50-2	105933	71984	-32.05	74348	74348	-29.82	72.07	91001	68994	-24.18	70795	70795	-22.20	71.24	82412	66852	-18.88	72609	72609	-11.90	70.69
INST-50-3	130186	73047	-43.89	89086	89086	-31.57	71.81	105073	67025	-36.21	73625	73625	-29.93	73.22	124142	66753	-46.23	77012	77012	-37.96	70.19
INST-50-4	50183	45174	-9.98	45174	45174	-9.98	73.67	48066	43148	-10.23	43148	43148	-10.23	69.59	50855	40724	-19.92	40724	40724	-19.92	72.58
INST-50-5	47427	38994	-17.78	38994	38994	-17.78	69.25	44495	35812	-19.51	35812	35812	-19.51	69.91	41528	33300	-19.81	33300	33300	-19.81	71.77
INST-50-6	33024	31982	-3.16	31982	31982	-3.16	72.26	30865	29901	-3.12	29901	29901	-3.12	68.72	29483	28460	-3.47	28460	28460	-3.47	72.63
INST-50-7	111388	69232	-37.85	75012	75012	-32.66	69.42	102101	68522	-32.89	70430	70430	-31.02	73.64	91054	66033	-27.48	69306	69306	-23.88	73.60
INST-50-8	84805	67932	-19.90	62219	62219	-26.63	68.72	80537	60657	-24.68	60657	60657	-24.68	71.87	75575	63546	-15.92	62184	62184	-17.72	73.84
INST-50-9	116119	77012	-33.68	75944	75944	-34.60	71.64	95961	68227	-28.90	68227	68227	-28.90	70.88	108639	67877	-37.52	66228	66228	-39.04	69.15
INST-50-10	92878	66461	-28.44	62199	62199	-33.03	70.70	73390	65626	-10.58	57155	57155	-22.12	71.84	79420	64955	-18.21	64035	64035	-19.37	68.83
INST-50-11	117808	37880	-67.85	72165	72165	-38.74	70.75	108470	74004	-31.77	73308	73308	-32.42	71.27	95130	71275	-25.08	65569	65569	-31.07	72.18
INST-50-12	54588	47793	-12.45	37880	37880	-30.61	71.97	45681	35797	-21.64	35797	35797	-21.64	71.88	46092	34724	-24.66	34724	34724	-24.66	68.56
INST-50-13	56652	67088	18.42	47793	47793	-15.64	72.62	53802	44750	-16.82	44750	44750	-16.82	71.26	48891	45034	-7.89	45034	45034	-7.89	71.15
INST-50-14	110621	74108	-33.01	69707	69707	-36.99	70.10	95455	71750	-24.83	72416	72416	-24.14	72.33	93613	69797	-25.44	70821	70821	-24.35	71.18
INST-50-15	108141	68924	-36.26	65270	65270	-39.64	71.97	97884	64938	-33.66	63211	63211	-35.42	71.13	77631	64678	-16.69	62981	62981	-18.87	73.17
INST-50-16	104432	64834	-37.92	74305	74305	-28.85	70.50	93251	71426	-23.40	70139	70139	-24.78	73.96	88572	72489	-18.16	69460	69460	-21.58	70.91
INST-50-17	47118	45770	-2.86	43328	43328	-8.04	73.05	52865	40110	-24.13	40110	40110	-24.13	69.31	44022	38712	-12.06	38712	38712	-12.06	70.36
INST-50-18	107997	61346	-43.20	71762	71762	-33.55	73.00	93741	62021	-33.84	65644	65644	-29.97	68.63	89645	61642	-31.24	65188	65188	-27.28	72.03
INST-50-19	51795	50976	-1.58	38870	38870	-24.95	69.54	41371	35374	-14.50	35374	35374	-14.50	68.66	41837	33656	-19.55	33656	33656	-19.55	72.45
INST-50-20	63304	60841	-3.89	55002	55002	-13.11	71.68	66898	59145	-11.59	48602	48602	-27.35	68.38	64854	61339	-5.42	47738	47738	-26.39	71.12
aver			-24.07			-25.93	71.33			-22.69			-23.17	70.96			-21.00			-21.31	71.34

LES, levels of earthquake severity; MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; UB, upper bound; VNS, Variable Neighborhood Search.

Table A10 The experimental results for TDTSP-PD with INST-50-x instances that are proposed by [25] (*LES* = 5).

Instances	Travel time function 1							Travel time function 2							Travel time function 3						
	UB	*TS+VNS*		MA				*UB*	*TS+VNS*		MA				*UB*	*TS+VNS*		MA			
		Best. Sol	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*
INST-50-1	124595	103625	-16.83	81046	81046	-34.95	70.09	117677	94790	-19.45	74963	74963	-36.30	72.93	121967	81693	-33.02	74171	74171	-39.19	71.09
INST-50-2	44342	36006	-18.80	36006	36006	-18.80	68.90	36093	34050	-5.66	34050	34050	-5.66	70.58	43044	32760	-23.89	32760	32760	-23.89	73.31
INST-50-3	96477	60652	-37.13	60948	60948	-36.83	71.52	88449	60595	-31.49	60346	60346	-31.77	73.33	85224	59743	-29.90	59012	59012	-30.76	71.53
INST-50-4	32207	29728	-7.70	29728	29728	-7.70	69.57	29984	27618	-7.89	27618	27618	-7.89	70.35	31406	27015	-13.98	27015	27015	-13.98	68.93
INST-50-5	112312	70111	-37.57	68421	68421	-39.08	68.27	97745	68555	-29.86	65631	65631	-32.85	72.61	103663	68015	-34.39	63580	63580	-38.67	69.20
INST-50-6	115818	99136	-14.40	74293	74293	-35.85	72.53	117412	78535	-33.11	71821	71821	-38.83	70.38	114142	78901	-30.87	69780	69780	-38.87	70.44
INST-50-7	122233	73607	-39.78	75115	75115	-38.55	69.46	106082	74065	-30.18	70976	70976	-33.09	72.85	100155	71693	-28.42	71648	71648	-28.46	72.49
INST-50-8	71251	53897	-24.36	67231	67231	-5.64	70.65	61386	46500	-24.25	46500	46500	-24.25	72.53	66551	49033	-26.32	49033	49033	-26.32	72.95
INST-50-9	130596	71411	-45.32	78225	78225	-40.10	72.13	115319	68315	-40.76	68315	68315	-40.76	70.26	110042	69122	-37.19	70928	70928	-35.54	72.74
INST-50-10	130260	94730	-27.28	76544	76544	-41.24	70.16	123613	75283	-39.10	72220	72220	-41.58	69.30	125969	81350	-35.42	70401	70401	-44.11	69.91
INST-50-11	43996	34020	-22.67	34020	34020	-22.67	72.42	39751	32437	-18.40	32437	64419	-18.40	72.74	38053	31282	-17.79	31282	63943	-17.79	71.20
INST-50-12	75711	49628	-34.45	73111	73111	-3.43	70.37	76570	49320	-35.59	49320	49320	-35.59	73.70	58067	48494	-16.49	48494	48494	-16.49	68.54
INST-50-13	69083	53668	-22.31	53668	53668	-22.31	72.10	60718	53212	-12.36	53212	53212	-12.36	69.97	68383	51129	-25.23	51129	51129	-25.23	68.67
INST-50-14	45619	38028	-16.64	38028	38028	-16.64	72.22	39627	36114	-8.87	36114	36114	-8.87	72.03	39751	34197	-13.97	34197	34197	-13.97	68.82
INST-50-15	168344	81494	-51.59	79251	79251	-52.92	70.65	132271	73986	-44.06	73651	73651	-44.32	70.63	110706	72189	-34.79	72004	72004	-34.96	72.07
INST-50-16	115686	75305	-34.91	64870	64870	-43.93	68.12	114345	66469	-41.87	77362	77362	-32.34	73.00	102725	67547	-34.24	67547	67547	-34.24	70.97
INST-50-17	156696	94849	-39.47	86284	86284	-44.94	69.99	105750	84094	-20.48	95818	95818	-9.39	72.61	115038	83174	-27.70	83174	83174	-27.70	69.14
INST-50-18	122323	78179	-36.09	72517	72517	-40.72	70.55	125431	72702	-42.04	73089	73089	-41.73	69.00	110135	77149	-29.95	70992	70992	-35.54	70.97
INST-50-19	72316	56972	-21.22	64163	64163	-11.27	69.62	71341	59969	-15.94	59969	59969	-15.94	73.17	73790	58100	-21.26	58100	58100	-21.26	68.89
INST-50-20	134246	114506	-14.70	87896	87896	-34.53	69.18	126710	105254	-16.93	88605	88605	-30.07	73.94	124364	97646	-21.48	92813	92813	-25.37	68.33
aver			-28.16			-29.61	70.42			-25.91			-27.10	71.80			-26.82			-28.62	70.51

LES, levels of earthquake severity; MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; UB, upper bound; VNS, Variable Neighborhood Search.

Table A11 The experimental results for TDTSP-PD with INST-100-x instances that are proposed by [25] (*LES* = 1).

Instances	Travel time function 1							Travel time function 2							Travel time function 3						
	UB	*TS+VNS*		MA				*UB*	*TS+VNS*		MA				*UB*	*TS+VNS*		MA			
		Best. Sol	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*
INST-100-1	52928	46293	-12.54	46945	46945	-11.30	220.6	53252	44777	-15.91	45276	45276	-14.98	223.7	48806	43870	-10.11	46340	46340	-5.05	222.8
INST-100-2	46079	43170	-6.31	44430	44430	-3.58	222.5	46236	41180	-10.94	44873	44873	-2.95	220.3	46160	40357	-12.57	43804	43804	-5.10	224.3
INST-100-3	46461	43800	-5.73	43543	43543	-6.28	224.3	48170	41477	-13.89	45881	45881	-4.75	224.3	43020	40973	-4.76	40576	40576	-5.68	221.7
INST-100-4	51478	49010	-4.79	50026	50026	-2.82	224.4	52645	46994	-10.73	47357	47357	-10.04	224.7	51536	47430	-7.97	47296	47296	-8.23	222.2
INST-100-5	48546	48083	-0.95	47447	47447	-2.26	221.4	48630	47485	-2.35	47872	47872	-1.56	224.9	54469	46987	-13.74	47933	47933	-12.00	220.3
INST-100-6	53367	51074	-4.30	48128	48128	-9.82	221.0	53007	50560	-4.62	47293	47293	-10.78	224.3	50594	48127	-4.88	47905	47905	-5.31	220.9
INST-100-7	68395	52336	-23.48	52191	52191	-23.69	222.8	60400	51331	-15.01	50569	50569	-16.28	223.9	61891	50269	-18.78	47812	47812	-22.75	223.3
INST-100-8	48616	46676	-3.99	46676	46676	-3.99	223.2	46419	43743	-5.76	43743	43743	-5.76	222.6	48390	46703	-3.49	46703	46703	-3.49	221.7
INST-100-9	48315	43172	-10.64	43172	43172	-10.64	222.1	44883	41623	-7.26	41623	41623	-7.26	220.9	42793	38713	-9.53	38713	38713	-9.53	224.5
INST-100-10	60203	53685	-10.83	50645	50645	-15.88	221.0	59537	54082	-9.16	48300	48300	-18.87	222.0	58823	53129	-9.68	47437	47437	-19.36	220.6
INST-100-11	61886	59954	-3.12	51665	51665	-16.52	224.7	60806	59147	-2.73	51076	51076	-16.00	220.7	57803	51718	-10.53	51718	51718	-10.53	224.9
INST-100-12	48187	47391	-1.65	47391	47391	-1.65	220.4	47657	46198	-3.06	46198	46198	-3.06	220.2	46113	43239	-6.23	43239	43239	-6.23	222.7
INST-100-13	56126	55215	-1.62	48284	48284	-13.97	220.5	51205	48737	-4.82	48737	48737	-4.82	224.7	54341	52854	-2.74	49584	49584	-8.75	223.5
INST-100-14	45913	42448	-7.55	42448	42448	-7.55	220.7	46889	41997	-10.43	41997	41997	-10.43	221.5	45628	42916	-5.94	42916	42916	-5.94	225.0
INST-100-15	50074	48616	-2.91	48616	48616	-2.91	220.8	52218	50762	-2.79	50762	50762	-2.79	221.5	53113	47720	-10.15	47720	47720	-10.15	221.4
INST-100-16	55285	48060	-13.07	51367	51367	-7.09	223.1	49378	46916	-4.99	47692	47692	-3.41	221.7	48958	48218	-1.51	47798	47798	-2.37	222.1
INST-100-17	49796	49289	-1.02	48472	48472	-2.66	222.9	50403	47791	-5.18	47492	47492	-5.78	222.3	57602	51514	-10.57	48858	48858	-15.18	222.3
INST-100-18	52086	51141	-1.81	49115	49115	-5.70	220.3	49146	46820	-4.73	46820	46820	-4.73	223.2	49508	47899	-3.25	47899	47899	-3.25	223.8
INST-100-19	59546	49325	-17.16	50258	50258	-15.60	224.7	53680	50475	-5.97	48552	48552	-9.55	220.1	54706	52218	-4.55	48817	48817	-10.76	224.1
INST-100-20	59070	52055	-11.88	48173	48173	-18.45	223.6	51763	50997	-1.48	46184	46184	-10.78	224.2	53877	52025	-3.44	46973	46973	-12.81	220.5
aver			-7.27			-9.12	222.2			-7.09			-8.23	222.6			-7.72			-9.12	222.6

LES, levels of earthquake severity; MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; UB, upper bound; VNS, Variable Neighborhood Search.

Table A12 The experimental results for TDTSP-PD with INST-100-x instances that are proposed by [25] (*LES* = 2).

Instances	Travel time function 1							Travel time function 2							Travel time function 3						
	UB	*TS+VNS*		MA				*UB*	*TS+VNS*		MA				*UB*	*TS+VNS*		MA			
		Best. Sol	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*
INST-100-1	63321	55439	-12.45	51483	51483	-18.70	220.9	65865	54668	-17.00	50963	5.0963	-22.63	224.5	74184	55971	-24.55	49359	49359	-33.46	223.2
INST-100-2	82591	62333	-24.53	59104	59104	-28.44	221.8	73301	58850	-19.71	57367	5.7367	-21.74	220.4	73328	61406	-16.26	57843	57843	-21.12	220.4
INST-100-3	74466	59660	-19.88	61801	61801	-17.01	220.3	78800	62284	-20.96	60728	6.0728	-22.93	221.2	83825	63796	-23.89	60241	60241	-28.13	220.4
INST-100-4	82634	65759	-20.42	71059	71059	-14.01	222.6	79150	64475	-18.54	66842	6.6842	-15.55	220.3	82181	65330	-20.50	68263	68263	-16.94	223.9
INST-100-5	42996	41578	-3.30	59836	59836	39.17	221.7	42166	41227	-2.23	58491	5.8491	38.72	222.2	46306	41951	-9.40	58553	58553	26.45	224.5
INST-100-6	47694	44376	-6.96	57175	57175	19.88	220.9	47249	44082	-6.70	57150	5.715	20.95	220.1	47715	45880	-3.85	57004	57004	19.47	222.7
INST-100-7	79382	59495	-25.05	67478	67478	-15.00	221.0	70375	59472	-15.49	66612	6.6612	-5.35	224.5	68504	57372	-16.25	64433	64433	-5.94	220.5
INST-100-8	96292	69723	-27.59	73162	73162	-24.02	224.5	102237	71249	-30.31	71428	7.1428	-30.13	221.0	89153	71504	-19.80	71487	71487	-19.82	224.1
INST-100-9	73958	58565	-20.81	69945	69945	-5.43	223.4	71334	55325	-22.44	69529	6.9529	-2.53	220.5	69347	55776	-19.57	66733	66733	-3.77	221.7
INST-100-10	68293	55634	-18.54	71405	71405	4.56	222.3	66652	54716	-17.91	69649	6.9649	4.50	221.5	64151	53755	-16.21	68514	68514	6.80	221.5
INST-100-11	62121	52017	-16.27	81119	81119	30.58	224.6	60068	50810	-15.41	76363	7.6363	27.13	222.3	59434	51598	-13.18	74789	74789	25.84	223.7
INST-100-12	57183	50314	-12.01	74191	74191	29.74	220.5	57988	51704	-10.84	71522	7.15563	23.34	220.5	56632	47935	-15.36	70390	70390	24.29	220.1
INST-100-13	58512	53062	-9.31	71958	71958	22.98	223.7	53218	50378	-5.34	70336	7.0336	32.17	225.0	55614	49078	-11.75	69268	69268	24.55	220.2
INST-100-14	46347	42567	-8.16	70647	70647	52.43	223.7	46726	42823	-8.35	70884	7.0884	51.70	221.7	46588	42507	-8.76	70173	70173	50.62	223.3
INST-100-15	93549	70231	-24.93	80096	80096	-14.38	222.8	91957	71460	-22.29	76515	7.6515	-16.79	221.5	85832	64065	-25.36	71847	71847	-16.29	223.0
INST-100-16	61754	52811	-14.48	49042	49042	-20.58	220.9	55279	48803	-11.72	57967	5.7967	4.86	220.3	49537	48503	-2.09	58440	58440	17.97	222.6
INST-100-17	82999	61843	-25.49	63623	63623	-23.34	223.0	78417	64004	-18.38	70498	7.0498	-10.10	221.5	98754	64249	-34.94	69380	69380	-29.74	223.6
INST-100-18	79515	62222	-21.75	64711	64711	-18.62	221.5	72121	62097	-13.90	66017	6.6017	-8.46	220.2	70848	62559	-11.70	65510	65510	-7.53	223.5
INST-100-19	48893	46211	-5.49	61024	61024	24.81	220.7	47705	46800	-1.90	61730	6.173	29.40	222.5	47347	46103	-2.63	60643	60643	28.08	223.9
INST-100-20	46768	46102	-1.42	60320	60320	28.98	221.1	45669	45403	-0.58	59260	5.926	29.76	223.8	45862	44909	-2.08	59839	59839	30.48	221.4
aver			-15.94			2.68	222.1			-14.00			5.32	221.8			-14.91			3.59	222.4

LES, levels of earthquake severity; MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; UB, upper bound; VNS, Variable Neighborhood Search.

Table A13 The experimental results for TDTSP-PD with INST-100-x instances that are proposed by [25] (*LES* = 3).

Instances	Travel time function 1							Travel time function 2							Travel time function 3						
	UB	*TS+VNS*		MA				*UB*	*TS+VNS*		MA				*UB*	*TS+VNS*		MA			
		Best. Sol	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*
INST-100-1	63321	55439	-12.45	51483	51483	-18.70	220.9	65865	54668	-17.00	50963	5.0963	-22.63	224.5	74184	55971	-24.55	49359	49359	-33.46	223.2
INST-100-2	82591	62333	-24.53	59104	59104	-28.44	221.8	73301	58850	-19.71	57367	5.7367	-21.74	220.4	73328	61406	-16.26	57843	57843	-21.12	220.4
INST-100-3	74466	59660	-19.88	61801	61801	-17.01	220.3	78800	62284	-20.96	60728	6.0728	-22.93	221.2	83825	63796	-23.89	60241	60241	-28.13	220.4
INST-100-4	82634	65759	-20.42	71059	71059	-14.01	222.6	79150	64475	-18.54	66842	6.6842	-15.55	220.3	82181	65330	-20.50	68263	68263	-16.94	223.9
INST-100-5	42996	41578	-3.30	59836	59836	39.17	221.7	42166	41227	-2.23	58491	5.8491	38.72	222.2	46306	41951	-9.40	58553	58553	26.45	224.5
INST-100-6	47694	44376	-6.96	57175	57175	19.88	220.9	47249	44082	-6.70	57150	5.715	20.95	220.1	47715	45880	-3.85	57004	57004	19.47	222.7
INST-100-7	79382	59495	-25.05	67478	67478	-15.00	221.0	70375	59472	-15.49	66612	6.6612	-5.35	224.5	68504	57372	-16.25	64433	64433	-5.94	220.5
INST-100-8	96292	69723	-27.59	73162	73162	-24.02	224.5	102237	71249	-30.31	71428	7.1428	-30.13	221.0	89153	71504	-19.80	71487	71487	-19.82	224.1
INST-100-9	73958	58565	-20.81	69945	69945	-5.43	223.4	71334	55325	-22.44	69529	6.9529	-2.53	220.5	69347	55776	-19.57	66733	66733	-3.77	221.7
INST-100-10	68293	55634	-18.54	71405	71405	4.56	222.3	66652	54716	-17.91	69649	6.9649	4.50	221.5	64151	53755	-16.21	68514	68514	6.80	221.5
INST-100-11	62121	52017	-16.27	81119	81119	30.58	224.6	60068	50810	-15.41	76363	7.6363	27.13	222.3	59434	51598	-13.18	74789	74789	25.84	223.7
INST-100-12	57183	50314	-12.01	74191	74191	29.74	220.5	57988	51704	-10.84	71522	7.15563	23.34	220.5	56632	47935	-15.36	70390	70390	24.29	220.1
INST-100-13	58512	53062	-9.31	71958	71958	22.98	223.7	53218	50378	-5.34	70336	7.0336	32.17	225.0	55614	49078	-11.75	69268	69268	24.55	220.2
INST-100-14	46347	42567	-8.16	70647	70647	52.43	223.7	46726	42823	-8.35	70884	7.0884	51.70	221.7	46588	42507	-8.76	70173	70173	50.62	223.3
INST-100-15	93549	70231	-24.93	80096	80096	-14.38	222.8	91957	71460	-22.29	76515	7.6515	-16.79	221.5	85832	64065	-25.36	71847	71847	-16.29	223.0
INST-100-16	61754	52811	-14.48	49042	49042	-20.58	220.9	55279	48803	-11.72	57967	5.7967	4.86	220.3	49537	48503	-2.09	58440	58440	17.97	222.6
INST-100-17	82999	61843	-25.49	63623	63623	-23.34	223.0	78417	64004	-18.38	70498	7.0498	-10.10	221.5	98754	64249	-34.94	69380	69380	-29.74	223.6
INST-100-18	79515	62222	-21.75	64711	64711	-18.62	221.5	72121	62097	-13.90	66017	6.6017	-8.46	220.2	70848	62559	-11.70	65510	65510	-7.53	223.5
INST-100-19	48893	46211	-5.49	61024	61024	24.81	220.7	47705	46800	-1.90	61730	6.173	29.40	222.5	47347	46103	-2.63	60643	60643	28.08	223.9
INST-100-20	46768	46102	-1.42	60320	60320	28.98	221.1	45669	45403	-0.58	59260	5.926	29.76	223.8	45862	44909	-2.08	59839	59839	30.48	221.4
aver			-15.94			2.68	222.1			-14.00			5.32	221.8			-14.91			3.59	222.4

LES, levels of earthquake severity; MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; UB, upper bound; VNS, Variable Neighborhood Search.

Table A14 The experimental results for TDTSP-PD with INST-100-x instances that are proposed by [25] (*LES* = 4).

Instances	Travel time function 1							Travel time function 2							Travel time function 3						
	UB	*TS+VNS*		MA				*UB*	*TS+VNS*		MA				*UB*	*TS+VNS*		MA			
		Best. Sol	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*
INST-100-1	65412	52462	-19.80	53042	53042	-18.91	220.6	65313	51196	-21.61	52680	52680	-19.34	223.4	66653	50917	-23.61	50801	50801	-23.78	220.5
INST-100-2	218523	109291	-49.99	128683	128683	-41.11	223.9	198152	100346	-49.36	127806	127806	-35.50	222.1	172026	114340	-33.53	107741	107741	-37.37	220.7
INST-100-3	188345	102656	-45.50	103161	103161	-45.23	221.5	179912	99077	-44.93	107242	107242	-40.39	222.3	154598	95008	-38.55	96780	96780	-37.40	220.8
INST-100-4	135381	122036	-9.86	102419	102419	-24.35	223.0	128096	117594	-8.20	100848	100848	-21.27	223.0	153184	126078	-17.70	110091	110091	-28.13	221.0
INST-100-5	163379	112232	-31.31	93392	93392	-42.84	224.8	150939	106277	-29.59	94847	94847	-37.16	220.3	154729	112777	-27.11	98899	98899	-36.08	221.6
INST-100-6	214933	117590	-45.29	112669	112669	-47.58	222.2	159040	114598	-27.94	109493	109493	-31.15	221.6	160059	113662	-28.99	106091	106091	-33.72	221.6
INST-100-7	236665	141534	-40.20	150495	150495	-36.41	223.5	220303	140319	-36.31	142785	142785	-35.19	223.9	201233	144293	-28.30	122754	122754	-39.00	221.1
INST-100-8	133270	94392	-29.17	94392	94392	-29.17	223.8	124163	85236	-31.35	85236	85236	-31.35	223.5	123445	78181	-36.67	78181	78181	-36.67	221.3
INST-100-9	164714	143738	-12.73	102917	102917	-37.52	222.2	157274	134973	-14.18	94745	94745	-39.76	220.6	142387	141545	-0.59	84024	84024	-40.99	224.5
INST-100-10	292495	143847	-50.82	188734	188734	-35.47	223.3	248116	141265	-43.06	128868	128868	-48.06	220.7	257640	148501	-42.36	154986	154986	-39.84	223.5
INST-100-11	108589	76568	-29.49	76568	76568	-29.49	220.5	111469	68769	-38.31	68769	68769	-38.31	220.5	103633	75335	-27.31	75335	75335	-27.31	222.8
INST-100-12	107405	70732	-34.14	70732	70732	-34.14	224.7	106396	71099	-33.18	71099	71099	-33.18	220.0	96225	67319	-30.04	67319	67319	-30.04	220.9
INST-100-13	57700	51705	-10.39	51705	51705	-10.39	220.9	54166	48384	-10.67	48384	48384	-10.67	222.1	54343	48136	-11.42	47612	47612	-12.39	221.1
INST-100-14	65672	53186	-19.01	53186	53186	-19.01	221.3	69884	57048	-18.37	57048	57048	-18.37	223.3	64539	50626	-21.56	55604	55604	-13.84	220.4
INST-100-15	165720	153432	-7.41	111953	111953	-32.44	224.0	151239	149998	-0.82	102910	102910	-31.96	223.6	171745	81386	-52.61	99776	99776	-41.90	224.6
INST-100-16	261298	127806	-51.09	160501	160501	-38.58	222.4	236687	140848	-40.49	159754	159754	-32.50	222.7	219396	145117	-33.86	154376	154376	-29.64	223.5
INST-100-17	62086	53591	-13.68	53591	53591	-13.68	223.8	60358	52010	-13.83	52010	52010	-13.83	220.5	69510	54287	-21.90	54287	54287	-21.90	222.8
INST-100-18	240627	111410	-53.70	114170	114170	-52.55	222.0	206273	125559	-39.13	115948	115948	-43.79	223.2	171602	130176	-24.14	120050	120050	-30.04	221.6
INST-100-19	131199	120526	-8.13	76457	76457	-41.72	221.4	119472	76169	-36.25	76169	76169	-36.25	220.6	134497	128121	-4.74	72737	72737	-45.92	220.8
INST-100-20	186279	120678	-35.22	109644	109644	-41.14	220.2	182859	128192	-29.90	110558	110558	-39.54	220.7	170685	143055	-16.19	105063	105063	-38.45	223.1
aver			-29.85			-33.59	222.5			-28.37			-31.88	221.9			-26.06			-32.22	221.9

LES, levels of earthquake severity; MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; UB, upper bound; VNS, Variable Neighborhood Search.

Table A15 The experimental results for TDTSP-PD with INST-100-x instances that are proposed by [25] (*LES* = 5).

Instances	Travel time function 1							Travel time function 2							Travel time function 3						
	UB	*TS+VNS*		MA				*UB*	*TS+VNS*		MA				*UB*	*TS+VNS*		MA			
		Best. Sol	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*		*Best. Sol*	gap [%]	*Best. Sol*	*Aver. Sol*	gap [%]	*T*
INST-100-1	320948	115312	-64.07	153335	153335	-52.22	224.9	261613	115682	-55.78	149112	149112	-43.00	223.0	255489	111366	-56.41	127362	127362	-50.15	223.8
INST-100-2	317627	151191	-52.40	203116	203116	-36.05	220.9	262047	155266	-40.75	167747	167747	-35.99	221.7	268073	149965	-44.06	195344	195344	-27.13	223.7
INST-100-3	351773	190780	-45.77	235556	235556	-33.04	221.3	349212	157754	-54.83	214004	214004	-38.72	221.5	344277	165241	-52.00	235298	235298	-31.65	223.7
INST-100-4	383419	188563	-50.82	238187	238187	-37.88	222.0	345549	181595	-47.45	194428	194428	-43.73	222.3	318229	164906	-48.18	209353	209353	-34.21	220.5
INST-100-5	91279	59984	-34.28	59984	59984	-34.28	220.4	87189	60636	-30.45	60636	60636	-30.45	222.1	85437	65239	-23.64	65239	65239	-23.64	223.4
INST-100-6	259352	166448	-35.82	148403	148403	-42.78	223.4	212302	154785	-27.09	117792	117792	-44.52	221.8	213675	150737	-29.46	123555	123555	-42.18	222.3
INST-100-7	154978	153641	-0.86	77868	77868	-49.76	222.0	128241	93704	-26.93	93704	93704	-26.93	222.8	140053	76217	-45.58	76217	76217	-45.58	221.1
INST-100-8	76206	55957	-26.57	55957	55957	-26.57	224.9	84883	54298	-36.03	54298	54298	-36.03	223.7	70865	60452	-14.69	60452	60452	-14.69	220.5
INST-100-9	49626	47911	-3.46	47911	47911	-3.46	222.0	47949	45501	-5.11	45501	45501	-5.11	222.1	46599	44799	-3.86	44799	44799	-3.86	224.1
INST-100-10	317837	184318	-42.01	191365	191365	-39.79	223.1	267244	164002	-38.63	193069	193069	-27.76	222.1	275642	163132	-40.82	189795	189795	-31.14	220.9
INST-100-11	67816	52689	-22.31	52689	52689	-22.31	220.8	62423	50696	-18.79	50696	50696	-18.79	220.6	69450	54792	-21.11	54792	54792	-21.11	220.8
INST-100-12	203535	189049	-7.12	111296	111296	-45.32	221.9	164967	111601	-32.35	111601	111601	-32.35	220.1	162531	114533	-29.53	114533	114533	-29.53	223.3
INST-100-13	88848	70651	-20.48	70651	70651	-20.48	220.8	91547	70050	-23.48	70050	70050	-23.48	221.5	82631	69132	-16.34	69132	69132	-16.34	224.5
INST-100-14	45345	41855	-7.70	41855	41855	-7.70	223.8	46028	40931	-11.07	40931	40931	-11.07	221.6	45147	42352	-6.19	42352	42352	-6.19	222.6
INST-100-15	300140	154103	-48.66	170937	170937	-43.05	224.4	255836	171907	-32.81	160716	160716	-37.18	223.3	254656	172765	-32.16	162426	162426	-36.22	223.5
INST-100-16	229541	104959	-54.27	145578	145578	-36.58	221.8	220501	185869	-15.71	101083	101083	-54.16	224.8	214003	172214	-19.53	100854	100854	-52.87	220.8
INST-100-17	350959	195350	-44.34	243616	243616	-30.59	223.4	322204	199489	-38.09	201411	201411	-37.49	224.7	358986	201302	-43.92	206935	206935	-42.36	224.8
INST-100-18	338441	174499	-48.44	211901	211901	-37.39	221.5	315218	175297	-44.39	240402	240402	-23.73	222.3	313635	178885	-42.96	213485	213485	-31.93	222.7
INST-100-19	264471	158971	-39.89	115428	115428	-56.36	222.7	245019	162952	-33.49	115659	115659	-52.80	221.2	218182	194353	-10.92	111754	111754	-48.78	223.4
INST-100-20	349101	193266	-44.64	205158	205158	-41.23	224.2	314820	172524	-45.20	201297	201297	-36.06	223.8	311962	188438	-39.60	175182	175182	-43.85	220.2
aver			-34.70			-34.84	222.5			-32.92			-32.97	222.3			-31.05			-31.67	222.5

LES, levels of earthquake severity; MA, Memetic Algorithm; TDTSP-PD, Time-Dependent Traveling Salesman Problem in Postdisaster; TS, Tabu Search; UB, upper bound; VNS, Variable Neighborhood Search.