A DECOMPOSITION-BASED METAHEURISTIC APPROACH FOR SOLVING THE RAPID NEEDS ASSESSMENT ROUTING PROBLEM

A Thesis

by

Yurtsev Mıhçıoğlu

Submitted to the
Graduate School of Sciences and Engineering
In Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Industrial Engineering

Özyeğin University
January 2021

A DECOMPOSITION-BASED METAHEURISTIC APPROACH FOR SOLVING THE RAPID NEEDS ASSESSMENT ROUTING PROBLEM

Approved by:

Asst. Professor Erinç Albey, Advisor
Department of Industrial Engineering
Özyeğin University

Asst. Professor İhsan Yanıkoğlu
Department of Industrial Engineering
Özyeğin University

Assoc. Professor Mehmet Güray Güler
Department of Industrial Engineering
Yıldız Technical University

Date Approved: 18 January 2021

To all humanity...

ABSTRACT

This study proposes a decomposition-based tabu search algorithm for a multi-cover routing problem (MCRP), which aims to classify and evaluate the impacts of the disaster in different sites and the needs of different community groups affected by a disaster when remote communication is not possible, and highlights the solution time and quality performances of the introduced algorithm. The algorithm focuses on decomposing the problem to three phases and to apply different methods while solving them. Performance of the proposed tabu search algorithm is evaluated with respect to different benchmark solutions, findings are put to statistical tests, and the results indicate that the proposed algorithm can achieve high-quality solutions expeditiously, providing better results on average compared with the best-known solutions existing in the literature.

ÖZETÇE

Bu çalışma, uzaktan iletişimin mümkün olmadığı bir senaryoda afetlerin farklı bölgelerdeki etkilerini sınıflandırma ve gözlemleme, ve bu afetlerden etkilenenlerin oluşturduğu komünlerin ihtiyaçlarını belirleme amacı güden birden çok karşılama için rotalama problemine ayrıştırma tabanlı bir yasaklanarak korunma araştırması algoritması geliştirerek sonuçların kalitesini ve sonuçlara erişilme zamanını incelemektedir. Geliştirilen algoritma, problemi üç evreye ayrıştırarak bu evrelere farklı çözüm metotları uygulamayı hedefler. Bu çalışmada önerilen algoritmanın performansı farklı denektaşlarıyla kıyaslanmış, bulgular istatistiksel testlere tabi tutulmuş, ve sonuçlar, literatürde bilinen en kaliteli sonuçları geliştirerek önerilen algoritmanın yüksek kaliteli sonuçlara hızlı bir şekilde erişebildiğini göstermiştir.

ACKNOWLEDGEMENTS

First of all, I would like to express my thanks and deepest gratitude to my family, especially to my parents, for all the support and the opportunities they provided me throughout my whole life.

Also, my sincerest thanks goes to all my friends, especially to I. Edhem Sakarya, the friendship they provided is greatly appreciated.

Finally, I would like to thank all the teachers I had through my years of study, who have provided me the smallest bit of knowledge, enlightening me, and to all the professors in the Industrial Engineering Department of Özyeğin University, especially Dr. Erinç Albey, for helping and guiding me through both my undergraduate and graduate studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Needs assessment could be defined as how humanitarian organizations identify and measure the needs of the communities that is affected by a disaster, in other words, it is the best way to answer the question “What assistance do disaster-affected communities need?” [1]. In general, there exists two stages of needs assessment: the rapid needs assessment stage and the detailed needs assessment stage [2]. The former one should start immediately after a disaster occurs and should lead to a design of a plan based on the needs of the affected people within 3 days at the latest [3]. Teams conducted by high experienced generalists of humanitarian organizations visit the affected sites to make observations (e.g., look, smell, touch) and talk to people (e. g., interviews) [4]. Since, the rapid needs assessment stage should be completed in a small-time period, it is not possible to visit all the sites that is affected by the disaster. Thus, humanitarian organizations first identify several target community groups based on the relevant factors such as geographical and density characteristics, and severity of impact on the sites, that may lead to differences in survival needs, in the affected area [5]. Once the target community groups are identified, selection of the sites that carry the specified community groups to be visited is conducted. To increase the reliability of the evaluations, visiting the same type of community group several times is desired [6]. A multi-cover routing problem (MCRP) is defined in [6], which assumes a pre-specified coverage target for each community group (i.e., which represents the minimum number of site visits for each community group to achieve a reliable observation) and studied in this study. Once the site selection phase is done, the information gathered by the teams should be analyzed together to inform the stakeholders and to enable all humanitarian actors from the outset [7]. This requires communication among the teams once each selected site is visited. However, in such tremendous events, telecom landlines and wireless systems could be lacking, destroyed or overloaded [8]. What is more, sadly, even if those technological difficulties do not occur, a rapid deployment of communication systems is still required [9]. Also, as stated in [6], it is difficult to foresee the availability of the information-sharing environment, and flexible solution methods should be developed. All the above-mentioned information points out that, to analyze the information of all visited sites, teams must communicate in person, thus, in this study, it is assumed that remote communication is not possible, and all teams must return to the coordination center after visiting the sites that they are assigned to. Which in conclusion, breaks the problem into 3 phases: selection of the sites in a way that pre-specified coverage targets for each community group is met, assignment of the selected sites to assessment teams, and providing a route to each team that starts and ends at the coordination center. It is stated in the literature that exact solution methods of the formulations provided for the problem may not produce high quality solutions in reasonable times [6]. Therefore, a decomposition-based tabu search algorithm, which uses a near-exact solution method in solving the third phase of the problem and referring to this solution inside the neighborhood moves is developed. The remainder of this thesis is structured as follows. In Chapter 2, the relevant literature is reviewed. The problem is described in detail in Chapter 3. The solution methods are discussed in Chapter 4, and the computational results and comparisons with the benchmark solutions are presented in Chapter 5. The numerical analysis and statistical tests upon the solutions presented is going to be discussed in Chapter 6, and finally, Chapter 7 presents the conclusions and future work directions.

CHAPTER II

LITERATURE REVIEW

Routing problems, which consists of numerous variants, are widely studied in the literature for years, especially in logistics, and the number of papers published is growing with an increasing rate in the field [10]. In addition, there also exist a rich literature on post-disaster humanitarian logistics problems [6]. However, since some organizations are focusing on fulfilling the pre-identified needs by partners, they do not operate a needs assessment procedure [11]. Unfortunately, this may not be efficient since predicting the needs of the victims affected by a disaster may be hard, and the needs can depend on numerous factors. Even though, with the frequent occurrence of disasters, some studies are conducted upon the real-world disasters in recent years (such as, [2], [6], [12] and [13], where the Ya'an earthquake (2013), Van earthquake (2011), tsunami Aceh (2004) and Van earthquake (2011) are studied respectively), post-disaster needs assessment still requires more attention in the literature, since the number of articles published are relatively small compared with the other routing problem variants, and since the lives of the affected people may depend on the improvements. In addition, there are some other studies proposing using new technological improvements as well, such as [14], where the authors consider using drones in order to inspect the damages in the affected region. However, new technologies can be expensive, can require additional expertise, and direct communication and discussion of the conditions are crucial to detect the needs in a post-disaster scenario [6]. These findings highlight the need on focusing and studying the rapid needs assessment, and building reliable, flexible, and robust techniques to solve rapid needs assessment problems in the literature. There also some examples of decomposition-based metaheuristic approaches, a similar technique to what is proposed in this study, such as [15], where the authors developed a genetic algorithm as the route generation mechanism for the main routing problem, for routing problems, exists. This study is closely related to [6], since the need on focusing the needs assessment problem is seen, and the variant introduced in [6], MCRP, is considered as a realistic demonstration on the real-world problem. The author in [6] introduced some variants ($MCRP^{rd}$ and $MCRP^{na}$) of the prize collecting routing problem (PCRP), where the PCRP aims to minimize the total distance traveled, minimize the vehicles used, and maximize the prize collected, with no need to visit each customer selected, but every customer is associated a prize [16], while the $MCRP^{rd}$ is aiming to minimize the maximum route duration, and $MCRP^{na}$ aiming to minimize the maximum node duration, while satisfying the target coverage for each community group in the problem. Although there exists different exact [17] and heuristic [18] methods for solving the PCRP, they cannot be used directly to MCRP due to definition of the objective functions of the problem [6]. Thus, a new solution method to MCRP is introduced. In summary this study contributes to the literature by focusing on the rapid needs assessment stage of a post-disaster scheme of humanitarian operations, proposing the humanitarian organizations a new practical and flexible solution method, which is out-performing the state-of-the-art algorithm existing in the literature.

CHAPTER III

PROBLEM DEFINITION

The multi-cover routing problem (MCRP), is a combinatorial optimization problem, addressing the selection of sites, assignment of the assessment teams, and the routing of the selected sites among the assigned teams. Given a set of sites, with each site carrying at least one characteristic of the specified target community groups, and a target coverage for each community group, MCRP determines the sites to be visited, the assignment of the selected sites to the assessment teams and constructs the routes of the assessment teams to guarantee that all target coverages are met as fast as possible. It is assumed that the assessment coordinators already have the secondary data, which includes definitions of different community groups of the field before the disaster occurs, hence, the community groups are already specified and the set of community groups carried by each site is already known, considering they usually obtain this data from secondary sources pre-disaster [19]. In addition, since the number of observations per community group needed may depend on several factors such as the size of the region, existing knowledge or existing vulnerabilities of the field and the logistical constraints [20], it is assumed that assessment coordinators already specified a coverage target for each community group, and similar to the study in [6], the travel times, and the estimated time spent in the nodes, which are included in travel times, are assumed to be known. Furthermore, since this study assumes that remote communication is not possible due to flexibility and robustness concerns, all teams must return to coordination center, indicating that the MCRP in this study aims to *minimize the maximum route duration* of teams. However, it should be pointed out that one other variant of MCRP, which assumes that remote communication is available and focuses on *minimizing the maximum node arrival time*, is also mentioned in the literature [6]. In addition, the formulation of both MCRP variants with different subtour elimination constraints (i.e., Miller–Tucker–Zemlin (MTZ) constraints [21] & Dantzig, Fulkerson and Johnson (DFJ) constraints [22]) are already available, the interested readers are referred to review [6]. The performance of the algorithm proposed in this study against the mentioned formulations is going to be discussed as well in Chapter 5. What is more, since the focus is on the case where assessment teams should start and end their routes on coordination center, and the route lengths of the teams are independent from each other once the selection and assignment phase of the selected sites are over, the resulting routing problem for each team becomes the well-known traveling salesman problem (TSP).

CHAPTER IV

SOLUTION METHODS

Tabu search algorithm is first introduced by Fred Glover in 1986 [23] and it is one of the most widely applied metaheuristics for solving TSP [24] and different vehicle routing variants [6]. In fact, tabu search is considered as the most successful method compared to other classical heuristic methods [25]. The idea of the algorithm is to prevent cycling back to the same solutions by prohibiting specific moves that has already been made during the previous iterations for a specific number of iterations (i.e., tabu tenure), by keeping them in a tabu list. This study proposes a tabu search algorithm to solve the mentioned MCRP. The algorithm uses a near-exact solution method, Concorde solver to be specific, for solving the routing phase of the initial solution and in the created traveling salesman problem (TSP) instances inside the neighborhood moves of the algorithm, which basically translates into applying a tabu search algorithm to the first two phases of the problem, to conduct the selection and assignment of nodes that will be sent to Concorde solver at each iteration, in every neighborhood move, to obtain a solution to the MCRP. In summary, the tabu search algorithm creates assignments for teams with different routes, transfers the node assignments to Concorde to solve the respective TSPs, and according to solutions obtained from the third phase, decides which assignment to continue with in the algorithm.

4.1 General Outline Of The Algorithm

First, an initial solution, which will be explained in Section 4.2, is constructed. Then, in each iteration, four different neighborhood moves, which are *shift*, *swap*, *remove-insert* and *remove*, are applied. All moves have their own tabu lists, and they will be explained in detail in Section 4.3. Algorithm continues until the specified iteration limit is reached. The general flow of the algorithm can be seen in Fig. 1.

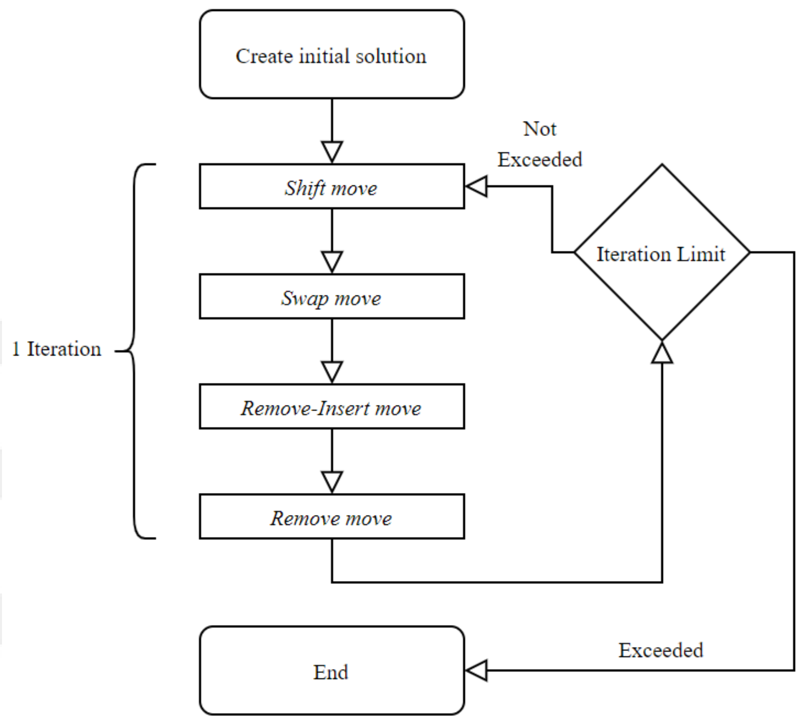

Figure 1: Flow chart of the algorithm

4.2 Construction of the Initial Solution

Constructing the initial solution has three phases: node (i.e., site) selection, vehicle assignment and routing. A very similar method to Iterative Clustering Heuristic proposed in [6] with small updates is used in the first two phases.

4.2.1 Node Selection

During the *node selection*, critical community group(s) are identified by calculating a critical ratio for each group, which is found by dividing the current target coverage (i.e., the minimum number of sites to visit to achieve the target number of observations) for that group by the number of unselected nodes carrying the characteristic of the respective group, and the community group(s) with the largest critical ratio are defined as critical groups(s). Then, candidate nodes are defined as the nodes that carry the characteristics of the critical group(s). Amongst the candidate nodes, the one with the highest number of total community group characteristic carried is selected, if this criterion is satisfied by multiple nodes, the one with the highest number of total community groups carried by the candidate node and its two *unselected* closest neighbors is selected, if there is a further tie, the one closest to the coordination center(depot) is selected. This procedure continues until target coverages for each group are satisfied. Once the nodes are selected, a removal procedure takes place, which checks if removing a node is possible without violating the target coverage constraints. If there are multiple nodes that can be removed, the one farthest from the depot is removed. Removal procedure continues until no more nodes can be removed from the selected nodes. The visualization of the node selection phase can be seen in Fig. 2.

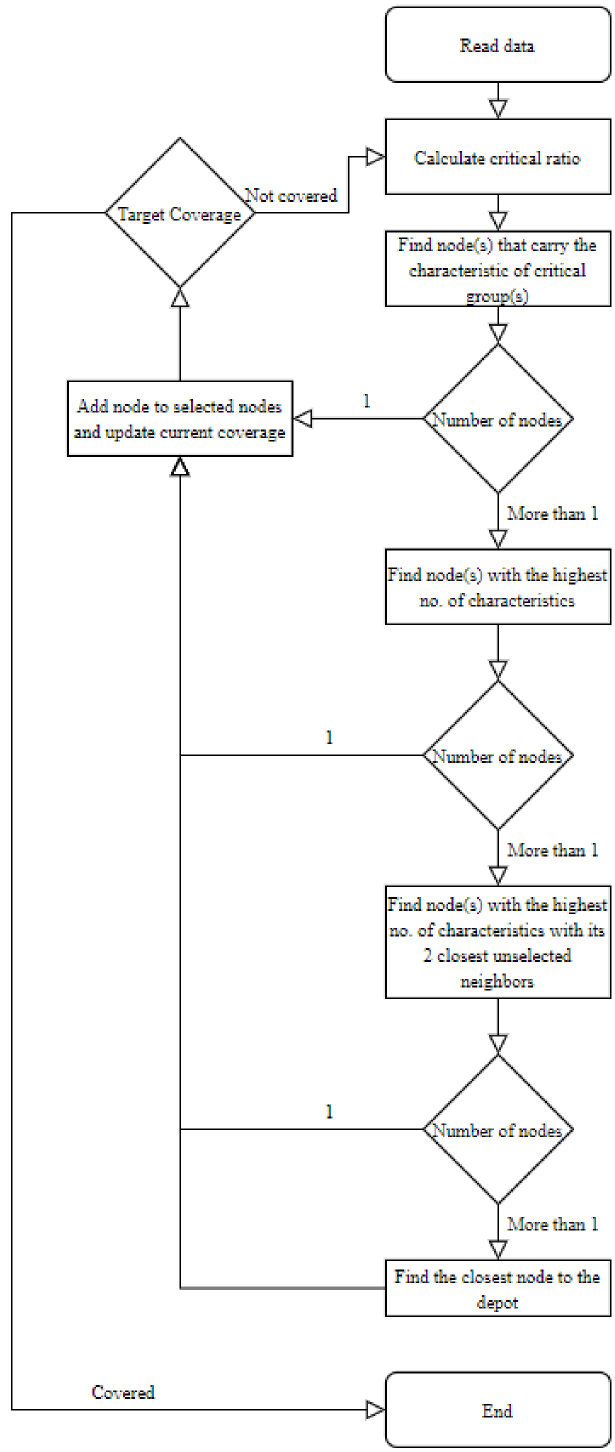

Figure 2: Node selection visualized

4.2.2 Assignment

During the *assignment* phase, initially one node is assigned to each team; the node that is farthest from the depot is assigned to the first team, then, for each remaining team, the node whose total distance to the currently assigned nodes is largest, is assigned to the next team. Once each team is assigned a node, the remaining nodes are assigned to the team that is currently visiting closest assigned node iteratively. When all selected nodes are assigned to a team, assignment phase ends.

4.2.3 Routing

As mentioned before, the *routing* phase becomes the well-known traveling salesman problem, thus, in the routing phase, the number of TSPs needs to be solved is equal to the number of teams. In the algorithm, TSP of each team is solved with Concorde solver and the maximum route duration is reported, since minimizing the duration of each route leads to the minimization of the maximum route duration. A very straightforward proof can be demonstrated as:

Proof. Assume that a problem has i routes, and the minimum route duration of the n^{th} route is represented as a_n, $n \in \{1, 2, ..., i\}$, thus, all other solutions to the n^{th} route can be represented as $a_n + \epsilon_n$, where $\epsilon_n \geq 0$, since the minimum value is already assumed as a_n. The objective function value of the problem when all routes are solved to optimality, z, becomes the duration of the route with maximum duration, and let us say that the j^{th} route is the one with the maximum duration.

$$z = a_j = max\{a_1, a_2, ..., a_j, ..., a_i\} \tag{1}$$

If we are to show that z is in fact the best solution that can be found, then the proof is done. Now let us assume that there is such solution, z_0, where $z_0 < z$ exists.

$$z_0 = max\{a_1 + \epsilon_1, a_2 + \epsilon_2, ..., a_j + \epsilon_j, ..., a_i + \epsilon_i\} \tag{2}$$

There are two possibilities, either

$$z_0 = a_j + \epsilon_j \tag{3}$$

or

$$z_0 = a_k + \epsilon_k, k \neq j \tag{4}$$

holds. If (3) holds, there cannot be such z_0, since we already know that $a_j \leq a_j + \epsilon_j$, since $\epsilon_j \geq 0$. If (4) holds, it will be known that $a_j + \epsilon_j \leq a_k + \epsilon_k$, because of (2). Since it is known that $a_j \leq a_j + \epsilon_j$ and $a_j + \epsilon_j \leq a_k + \epsilon_k$, implying $a_j \leq a_k + \epsilon_k$, there cannot be such z_0 as well, meaning z is in fact the best solution that can be found and that concludes the proof. □

4.3 Neighborhood Moves

4.3.1 Shift move

Shift move, first finds the route with the longest duration, checks each node in that route that is not in the tabu list and removes the one that the removal will lead to the highest decrease in the route duration. Afterwards, the removed node is inserted to each remaining route, and the insertion that will lead to the smallest route duration amongst the remaining routes is selected. An example of the shift move could be explained as: Let us assume that the current solution in hand has 4 node/team assignments (which means the problem has 4 teams), where nodes 1, 2, 3, 4 are assigned to team 1, nodes 5, 6, 7 are assigned to team 2, nodes 8, 9, 10 are assigned to team 3 and nodes 11, 12, 13 are assigned to team 4. And let us assume that the solutions of the TSPs of node/team assignments obtained with Concorde solver are 150, 120, 110 and 100 for teams, respectively. Shift move selects the first route since it is the one with the longest duration, removes one node, and solves the remaining assignment again, for each node that is not in tabu list (meaning that it will first remove node 1, calculate the route duration of remaining nodes (2, 3, 4), then it will remove node 2 from the original route, calculate the route duration of remaining nodes (1, 3, 4), and so on). After finding the node that its removal will lead to the highest decrease in the route duration, let us assume that it is node 1, making the route duration of team 1, 110, the removed node will be inserted to each remaining route, and the route durations are calculated again (meaning that it will first insert node 1 to assignment of team 2, leading the node/team assignment to become 1, 5, 6, 7, then it will insert node 1 to assignment of team 3, leading the node/team assignment to become 1, 8, 9, 10, and so on). After the insertions, let us say the new route durations for teams 2, 3 and 4 become 130, 170 and 120, which are 120, 110 and 100 before the insertion in that case, the move will select the insertion that leads to the smallest route duration, which in that case is 120, inserting node 1 to assignment of team 4. After the move, nodes 2, 3, 4 are assigned to team 1, nodes 5, 6, 7 are assigned to team 2, nodes 8, 9, 10 are assigned to team 3 and nodes 1, 11, 12, 13 are assigned to team 4, with the new durations of 110 for team 1, 120 for team 2, 110 for team 3 and 120 for team 4. During the move, all the duration changes in the routes that the insertion and removal made are calculated using Concorde solver. It should also be mentioned that, if all the nodes in the route with the longest duration are in the tabu list, this move only changes the tabu status of the nodes.

4.3.2 Swap move

Swap move basically tries to remove one node from the solution and insert another one among the nodes that currently are not in the solution, while ensuring feasibility. This move focuses on the route with the longest duration, it first selects all the non-tabu nodes in the longest route and starts checking if removing any of these nodes from the solution, while inserting another node which currently is not in the solution, is feasible in terms of target coverage. If there exist such nodes that it can be removed, the node with its removal leading to the highest decrease in the duration of the longest route is removed. Afterwards, all nodes that can re-ensure feasibility are inserted to each route one by one, and the insertion with the *smallest route duration* amongst all possible insertions is made. Just like the *shift* move, the algorithm calculates the route duration changes with Concorde solver. Once again, if all the nodes in the longest route are in the tabu list, this move only changes the tabu status of the nodes.

4.3.3 Remove-Insert move

Remove-Insert move first finds the non-tabu node that its removal will lead to the highest decrease in the duration of the longest route and removes the node without concerning feasibility. Afterwards, the community group(s) whose target coverages cannot be met after this removal are identified as infeasible groups. All unselected nodes carrying the characteristics of the infeasible group(s) are selected as candidate insertion nodes. Amongst the insertion of candidate insertion nodes to each route, the insertion of the node to the route with the *smallest route duration* is made. This procedure continues until the target coverage for each community group is met again. Same as the above-mentioned moves, the algorithm calculates the route duration changes using Concorde solver. Similar to the other moves, if all the nodes in the longest route are in the tabu list, this move only changes the tabu status of the nodes.

4.3.4 Remove move

Remove move checks whether removing a node from the solution is possible while ensuring feasibility. If there are multiple nodes that can be removed, the node that will lead to the highest decrease in the longest route is removed. Once again, the algorithm calculates the route duration changes based on Concorde solver.

Please note that, all the moves except *remove* move, since removing a node from the solution cannot increase any of the route durations, may lead to an increase in the objective function value, which is basically the diversification method of the algorithm, and all the moves are deterministic.

4.4 Concorde Solver

Concorde is a near-exact method to solve the traveling salesman problem. It has been used to solve all 110 TSPLIB instances, which are widely known and studied in the literature, to optimality, with the largest having 85,900 cities [26]. And for many other instances having millions of cities, the proposed solution by Concorde is reported to be within 2-3% of the optimal tour [27]. This implies that, in the context of this study, since this study focuses on the instances with a maximum of 100 nodes, Concorde is extremely reliable.

4.5 Decomposition

In general, decomposition approach means to solve a problem by breaking it up into smaller problems, and solving them either in parallel or sequentially, which has their origins in the context of large-scale linear programming in the 1960s [28]. As mentioned above, the MCRP has three phases, which are: the selection of the nodes to visit, assignment of the selected nodes to teams, and to obtain the routes of the assessment teams. If one is to decompose and divide the MCRP to two smaller problems, one containing the first two phases, and the other one containing the third phase, the first sub-problem becomes a smaller one, which is aiming to find a subset of the nodes in the original MCRP that satisfy the target coverage, and to assign them to teams (a variant of the set covering problem mixed with the team orienteering problem without concerning about the routing part of the team orienteering problem). And it is trivial to see that the second sub-problem becomes a variant of the vehicle routing problem, where the aim is to minimize the maximum route duration. In the algorithm, tabu search focuses on the first sub-problem by conducting neighborhood moves that changes the selected nodes and the assignment of the selected nodes to teams, with the information from the Concorde solver, which is basically focusing on the second sub-problem by solving the TSP of each route to minimize the maximum duration of the routes in the solution.

4.6 Tabu Search Parameters

Parameter-tuning, which can be simply put as deciding on the predefined set of parameters of an approach according to the problem at hand, is known to be crucial while using a metaheuristic approach since those parameters may have a large influence on the effectiveness and efficiency of the search [29]. Even if it is widely studied in the literature, universally optimal parameter values set for a specified metaheuristic does not exist [30]. The algorithm requires setting up the tabu tenure (TT) value, so, 3 different values of static tabu tenures, which are 5, 7 and 10 are analyzed. 5, 7 and 10 are chosen according to the number of nodes in the problem. The 48 instances introduced in [6], which are going to be used in solution quality and time comparison as well, have 25, 50, 75 and 100 nodes, and those instances will be further discussed in Chapter 5. So, a set of tabu tenure values is decided accordingly. To be more specific, the set is decided upon this criterion: find the square root of the minimum instance size, which is 5 in that case, find the square root of the maximum instance size, which is 10 in that case, and the average of them, which is 7.5 in that case, which added to the set as 7. The short-run solutions of TS_I algorithm mentioned in [6] is chosen as the reference point of the analysis. The average percentage of objective value difference of 48 instances with respect to the reference point comparison between different tabu tenure values can be seen in Fig. 3.

Even though the performances are in fact close, the algorithm performed best when the tabu tenure is selected as 7, with an average of 1.58% improvement compared to the short run of the proposed tabu search algorithm in [6]. In addition, the difference between the objective value difference percentages with different tabu tenure values are indeed close, being -1.48%, -1.58%, -1.52% with tabu tenures 5, 7 and 10 respectively, indicating that the performance of the algorithm is robust against

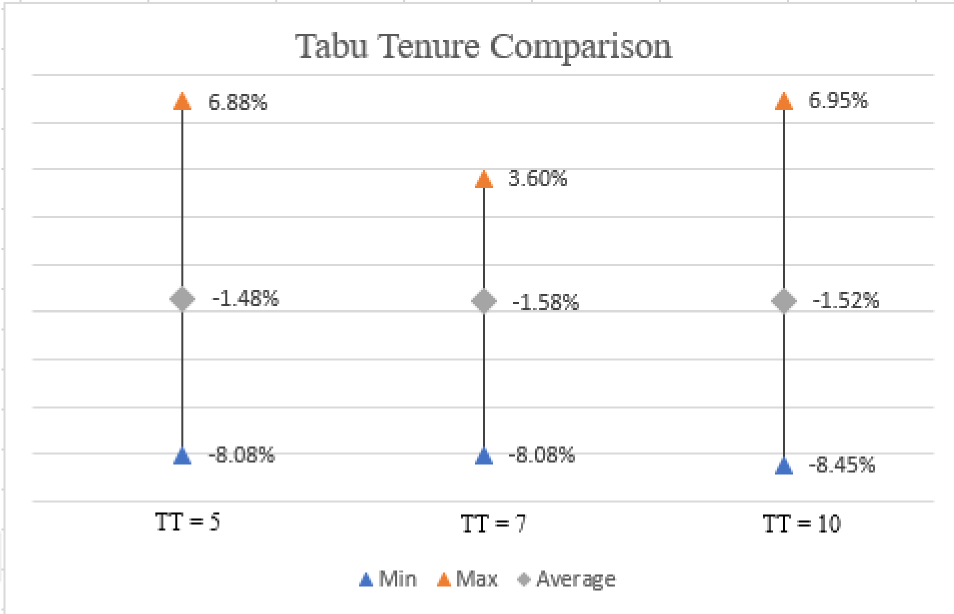

Figure 3: Tabu tenure comparison

the differences in tabu tenure values if it is in a specified range (in this case, between 5 and 10). During further analysis, the average objective value difference with respect to the reference point according to the instance size for different tabu tenure values are calculated and can be seen in Table 1.

	TT=5	TT=7	TT=10
25 Nodes	0.69%	0.16%	1.14%
50 Nodes	-1.27%	-1.43%	-1.27%
75 Nodes	-3.14%	-2.79%	-3.30%
100 Nodes	-2.19%	-2.25%	-2.63%

Table 1: Average objective value difference percentages according to problem size with different Tabu Tenure (TT) values.

It is concluded that, the tabu tenure value which is closest to the square root of the instance size among the selected set, performs better, except for the instances with 25 nodes. This is because, selecting tabu tenure as small as 5, may lead to cycling, when the instance size is small. Thus, it is decided to set the tabu tenure value as the closest of 7 and 10 compared to the square root of instance size (i.e., the tabu tenure is set to 7 for the first 24 instances, which have 25 and 50 nodes, and the tabu tenure is set to 10 for the remaining 24 instances, which have 75 and 100 nodes). Please note that, this is not a benchmark comparison, this is the result of the analysis that is conducted to decide on the metaheuristic parameter. To generalize, selecting the square root of the number of sites in a problem as the tabu tenure, if it is greater than 5, is suggested. In addition, it is believed that the performance of the algorithm could be further improved if a different methodology of choosing the tabu tenure value is applied, this will be discussed in Chapter 7. The details of the instances, benchmark comparisons and the result of the algorithm proposed in this study is going to be discussed in Chapter 5.

CHAPTER V

COMPUTATIONAL RESULTS

In this chapter, the results of the algorithm are going to be presented to evaluate the performance. The benchmark is chosen as the tabu search algorithm introduced in [6], since it provides the best-known solutions in the literature for the defined 48 instances of MCRP in [6]. The comparison of results with the solutions found by CPLEX in 1 hour with different subtour elimination constraints is presented in Section 5.3. Afterwards, the comparison between the algorithm and the proposed tabu search algorithm in [6], TS_I, is going to be done in Section 5.4. The algorithm proposed in this study is going to be referred as TS_C from this point on. I also want to mention the differences between the two tabu search algorithms proposed in [6] briefly. First one, TS_I, builds the initial solution based on an iterative clustering heuristic, while the second one, TS_L, constructs the initial solution of the algorithm by a location-based clustering model using an optimization model. Also, the difference between short-runs and long-runs is just the number of iterations, with short-run running for 3×10^5 iterations, and long-run running for 2.4×10^6 iterations. Interested readers are referred to [6] for more details. Since it is mentioned in [6] that using TS_I could be more reasonable because it is slightly faster and providing better solutions overall, the comparisons is going to be build with respect to TS_I. However, the maximum, average, and minimum objective value difference in percentages with respect to other algorithm, TS_L, without performing any further statistical analysis is going to be presented as well. All computations were performed on a macOS with 8 GB DDR3 RAM and 1.8 GHz Intel Core i5 CPU, and the algorithm is coded using Python.

5.1 Problem Instances

The performance of the proposed algorithm is tested on 48 hypothetical instances, obtained from [6]. The authors create the instances by modifying well-known Solomon instances [31] studied in [13]. In detail, 24 of these 48 instances consist randomly distributed nodes, which are denoted by R, and the remaining 24 has randomly clustered nodes, which are denoted by RC, among the network, with sizes of 25, 50, 75 and 100. In each instance, there are twelve community groups, which are assigned to sites randomly. Number of teams in an instance is created randomly, ranging between 2 and 6 [6]. Additionally, three different target coverage levels for community groups are defined, being low (L), medium (M) and high (H), by considering the total number of occurrences of a community group in the network [6]. The authors use different fractional numbers, ranging between 0.1 and 0.8, to set the target coverage parameters. For more information and details about the instance creation procedure, readers can refer to [6]. Instance name definition is as follows: the number of nodes/distribution type/number of teams/target coverage level. For example, the first instance, which is 25/R/2/L, has 25 nodes inside the problem which are distributed randomly among the network, has 2 teams and a low level of target coverage ratio. A visualization of the node distribution including the coordination center of instances with randomly distributed and randomly-clustered nodes can be seen in Fig. 4. and Fig. 5.

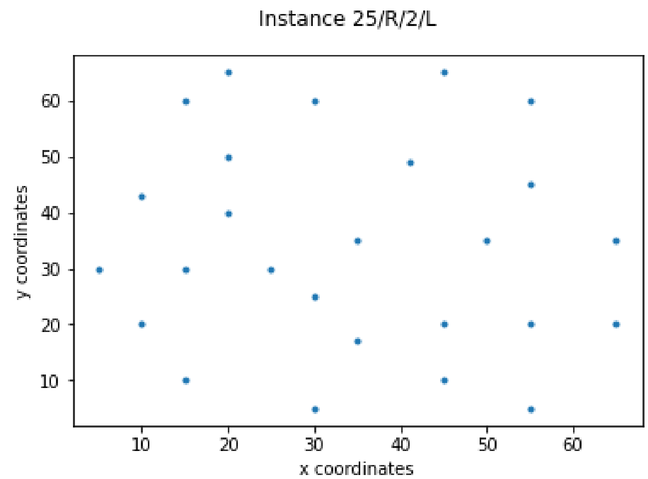

Figure 4: An instance with randomly distributed nodes

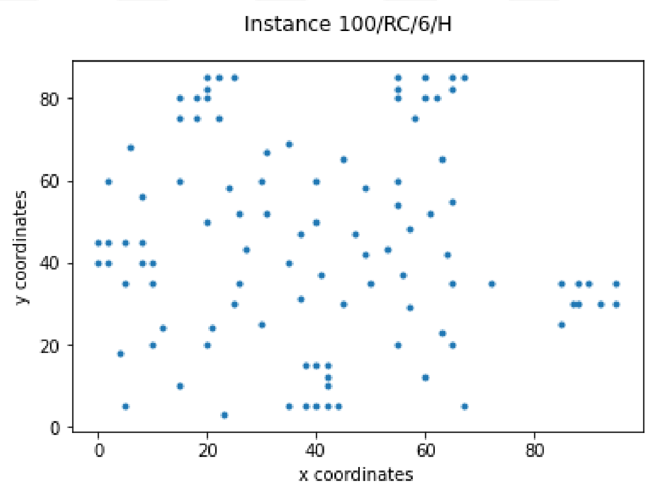

Figure 5: An instance with randomly-clustered nodes

5.2 General Comparison

The boxplots, where one can see the mean, median and the quartile information, of the objective value difference in percentages with respect to different benchmark algorithms, and the runtime graphics of all the mentioned methods is going to be presented in this chapter.

5.3 *Performance of the algorithm versus the formulation of MCRP*

In this section, the performance of the algorithm compared with the formulation solved in CPLEX in 1 hour is presented. The solutions of the formulation solved in CPLEX are obtained from [6]. The authors provided two alternatives on formulating the problem with different subtour elimination constraints in [6], however, since the solutions of the MTZ formulation are better for 30 out 48 instances, the same for 11 out of 48, and better overall, I will only be comparing the results of the algorithm with the MTZ formulation. The results can be seen in Fig. 6 and Table 2, where the boxplot of objective value differences in percentages of 48 instances and the maximum, average and minimum objective value difference in percentages is presented.

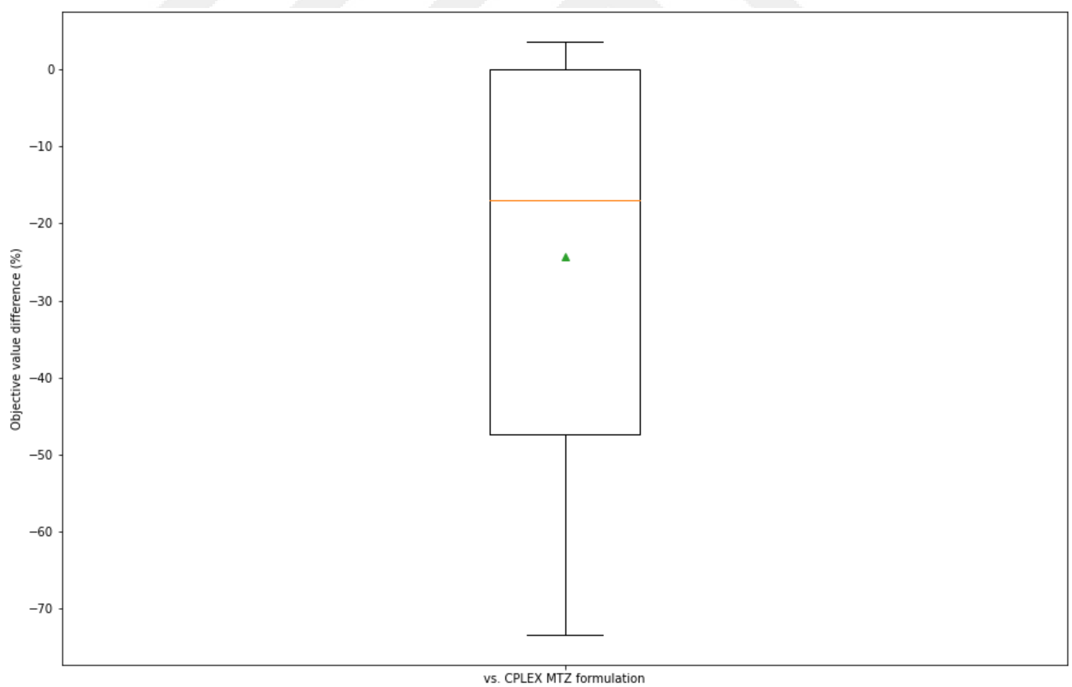

Figure 6: Objective value difference (%) comparison with CPLEX MTZ formulation

	vs. CPLEX MTZ formulation
Maximum	3.6%
Average	-24.4%
Minimum	-73.4%

Table 2: Maximum, average, minimum of objective value diff. (%) vs. CPLEX MTZ formulation

In addition, the run-time comparison in seconds can be seen in Fig. 7 and Table 3.

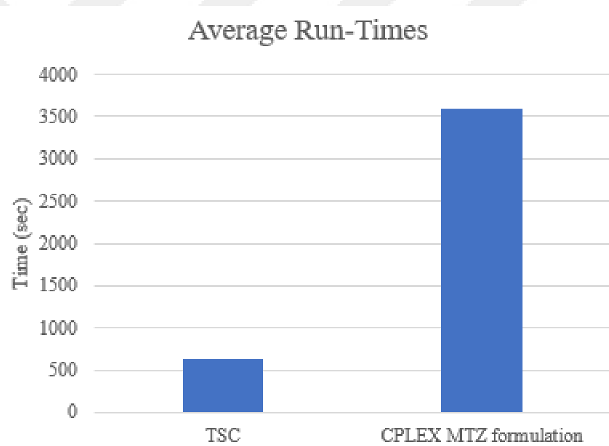

Figure 7: Run-time comparison with CPLEX MTZ formulation

	TS_C	CPLEX MTZ formulation
Maximum	1491.61	3600.00
Average	630.73	3600.00
Minimum	44.41	3600.00

Table 3: Run-time comparison vs. CPLEX MTZ formulation

I would also like to add that the objective value difference in percentages of TS_C

compared with the DFJ formulation is -34.5% on average. The results clearly illustrate that using a metaheuristic approach could result in higher-quality solutions, by outperforming the exact formulation by 24.4% and running nearly 6 times faster on average.

5.4 Performance of the algorithm versus the benchmark algorithm

This section presents the solution comparison between the algorithm proposed in this study versus the short and long runs of the best-known algorithm, TS_I, for the MCRP in terms of both solution quality and runtime. A comparison of the results with a procedure, which is referred as warmed up CPLEX method, where the authors in [6] feed the optimization software with the solution gathered from the short run of TS_I algorithm as the initial feasible solution, and solve for both MTZ and DFJ constraints, each for 1 hour, returning the best solution found (either from the model with MTZ subtour elimination constraints or from the model with DFJ subtour elimination constraints) for each instance is also done. The authors mentioned that this procedure failed to improve the results by an impactful amount, thus, applying this method is not reasonable, since the runtime of this procedure is the summation of the time it takes to complete the short run of TS_I algorithm plus 2 hours for each instance, in terms of solution quality, this is the state-of-the-art algorithm, providing the best results on average. The objective value difference comparison of TS_C with respect to both short run of TS_I, long run of TS_I and the warmed up CPLEX method can be seen in Fig. 8 and Table 4.

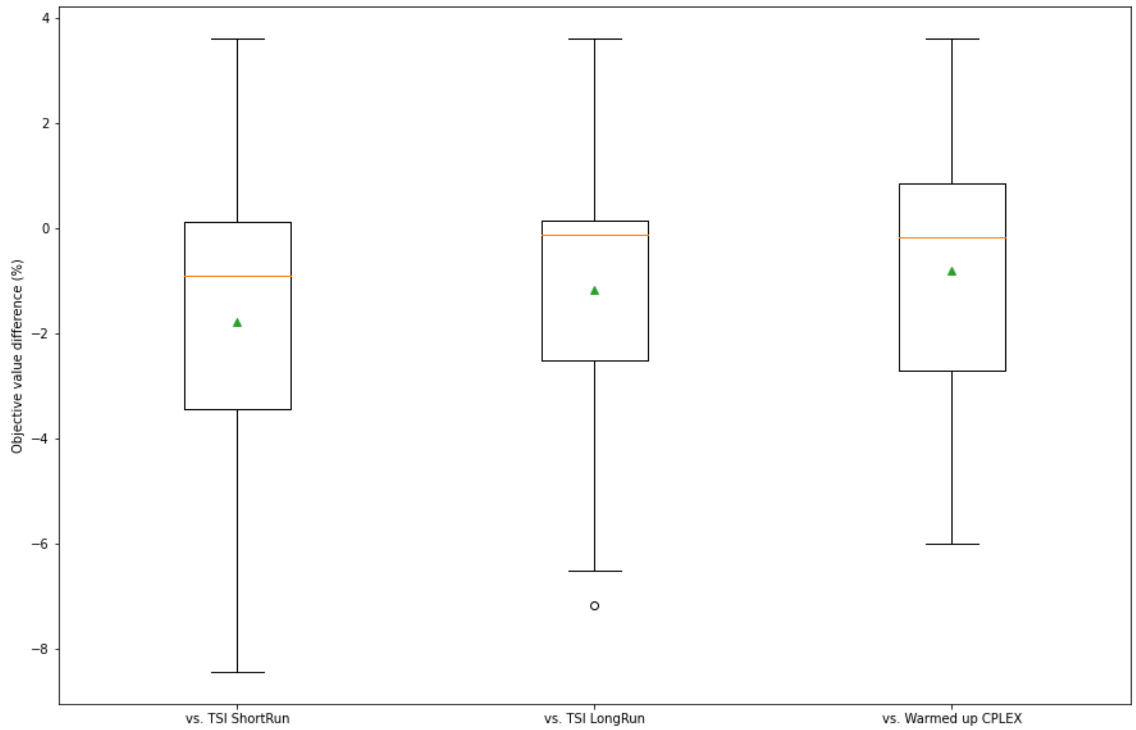

Figure 8: TS_C vs. Short Run and Long Run of TS_I, and TS_C vs. Warmed up CPLEX method

	vs. TS_I ShortRun	vs. TS_I LongRun	vs. Warmed up CPLEX
Maximum	3.6%	3.6%	3.6%
Average	-1.8%	-1.2%	-0.8%
Minimum	-8.5%	-7.2%	-6.0%

Table 4: Max., avg., min. obj. diff. (%) vs. TS_I ShortRun, TS_I LongRun, Warmed up CPLEX

It can be seen that the algorithm proposed in this study outperforms both the short run of TS_I, long run of TS_I and warmed up CPLEX procedure, with an average objective value difference percentage of -1.8%, -1.2% and -0.8% respectively. The runtime comparison of TS_C and three different methods mentioned above can be seen in Fig. 9 and Table 5.

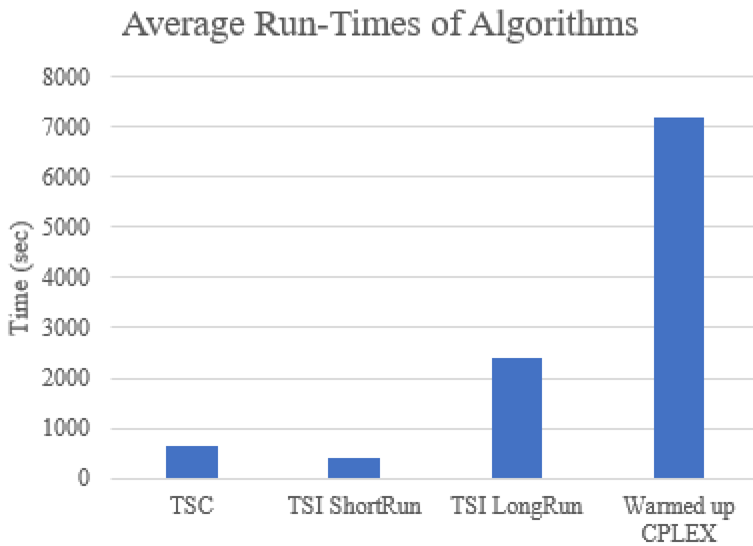

Figure 9: Run-time comparison of TS_C vs. Short and Long Run of TS_I and Warmed up CPLEX method

	TS_C	TS_I ShortRun	TS_I LongRun	Warmed up CPLEX
Maximum	1491.61	851.89	3600.00	7200.00+
Average	630.73	390.25	2411.56	7200.00+
Minimum	44.41	66.28	491.26	7200.00+

Table 5: Run-time comparison of TS_C vs. Short and Long Run of TS_I and Warmed up CPLEX method

The TS_C algorithm provides better solutions compared to the short run in 30 out of 48 instances (62.5%) and provides the same results in 4 out of 48 instances ($\sim$ 8%), while running nearly 1.5 times longer. However, this runtime difference in minutes is only translating into $\sim$ 4 on average. In addition, TS_C algorithm is also providing better results compared with the long run of the suggested algorithm, TS_I, with an average of 1.2% improvement, while running 4 times faster (the time difference translating into $\sim$ 30 minutes) and outperforming TS_I in 24 out of 48 instances

(50%), providing same results in 6 out of 48 instances (12.5%). The algorithm is also finding better results with an average of 1.9% and 1.2% improvement in the objective value compared to short run and long run of TS_L, respectively. Also, the algorithm also outperforms the warmed up CPLEX procedure as well, with an average of 0.8% improvement, while running for $\sim$ 10 minutes on average, compared to 120+ minutes. Furthermore, the solutions obtained for each instance from different algorithms and procedures can be seen in Appendix A.

CHAPTER VI

NUMERICAL ANALYSIS

In this chapter, whether if the findings of this study are significant, by applying statistical tests, and whether if there are some factors amongst instances that have an impact on the solution quality of the algorithm is going to be tested. Paired t-tests are conducted in order to comment on if the average objective value of 48 instances found by the algorithm is significantly different than the averages of objective values found with the short run of TS_I, long run of TS_I and the warmed up CPLEX procedure. Even though it is known that for $N > 30$, the results of the t-test are approximately correct even the compared two samples does not follow a normal distribution, Kolmogorov-Smirnov tests are done for each of the objective value set found by above-mentioned 4 methods, and see that the objective values may indeed follow a normal distribution, with the test failing to reject the null hypothesis, which assumes that the objective values follow a normal distribution, for each method, with p-values 0.42, 0.55, 0.44 and 0.52 for TS_C, short run TS_I, long run TS_I and warmed up CPLEX. After that, paired t-tests for TS_C and short run TS_I, TS_C and long run TS_I, and TS_C and warmed up CPLEX are conducted. Where the null hypothesis of the tests assuming the means of the objective values found by different algorithms are equal. The p-values of the conducted paired t-tests can be seen in Table 6.

	TS_C and TS_I S.Run	TS_C and TS_I L.Run	TS_C and W. up CPLEX
p-value	$6.33e-05$	0.00233	0.03352

Table 6: p-values of paired t-tests for different algorithms

The p-values show that the null hypothesis is rejected with a confidence level of

95%. Indicating that the solutions found by the algorithms are significantly different. In addition, whether if the number of nodes in the solution affects the solution quality of the algorithm is investigated, by first breaking the solutions achieved for instances in hand to 4 different categories according to node size (which are 25, 50, 75 and 100), the average objective function values for TS_C, short run of TS_I, long run of TS_I and warmed up CPLEX procedure can be seen in Table 7. Other comparisons made by differentiating amongst the characteristics of the instances can be seen in Appendix B.

	TS_C	TS_I S.Run	TS_I L.Run	Warmed up CPLEX
25 Nodes	84.8	84.7	84.5	84.4
50 Nodes	99.6	101.0	100.1	99.8
75 Nodes	105.8	109.0	108.4	107.8
100 Nodes	102.4	105.0	104.2	103.6

Table 7: Average objective values according to problem size

This basic comparison showed that, with the problem size growing in terms of the nodes it has, the algorithm is providing better results overall. To further investigate how the node size in an instance affects the solution quality, a basic linear regression model is used. The objective value difference in percentages for each instance versus the short run of TS_I can be seen in Fig. 10.

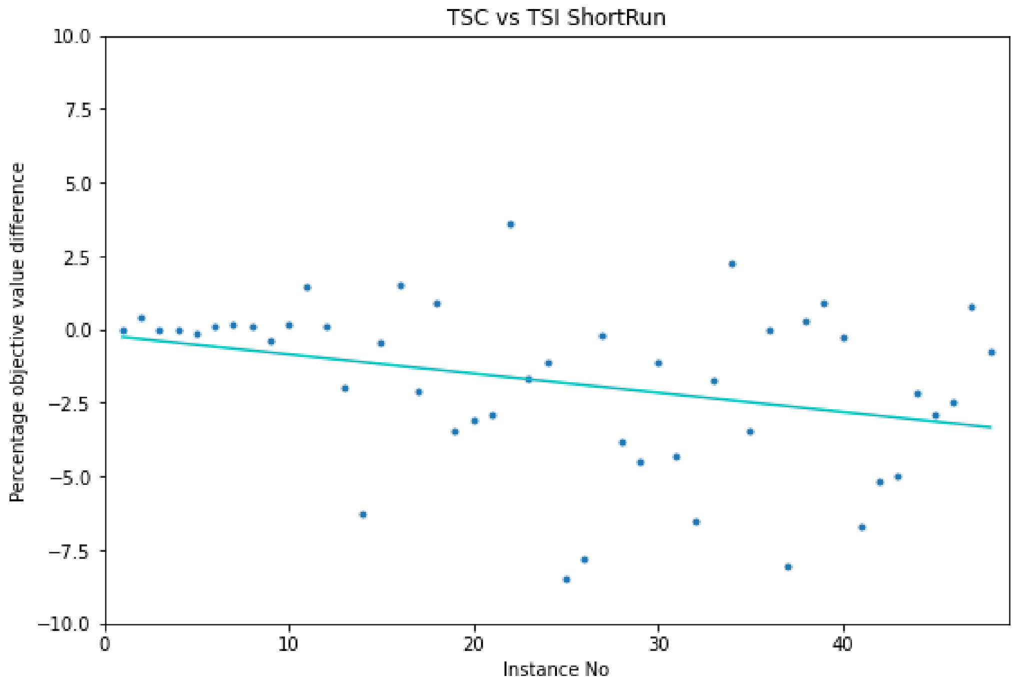

Figure 10: Objective value difference (%) for each instance vs. TS_I ShortRun

As mentioned above, a simple linear regression model is fitted, and it can be seen that when the instance size grows up (please note that the number of nodes in an instance increases with the instance no, for example, instances 1-12 has 25 nodes, instances 13-24 has 50 nodes, instances 25-36 has 75 nodes and instances 37-48 has 100 nodes), the improvement in the objective value difference tend to grow up as well, with the instance no having a p-value for the regression of 0.025, indicating that the instance no and the objective value difference is indeed correlated with a confidence level of 95%. This trend is also seen when the objective value differences between TS_C and the long run of TS_I, and TS_C and warmed up CPLEX method are investigated as well, the results can be seen in Appendix C.

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

In conclusion, this study provides a solution approach, which is outperforming the existing approaches in the literature, that can be used by humanitarian organizations in a post-disaster environment, during the rapid needs assessment planning. Even though the assumptions made in this study are realistic, this study is focusing on a deterministic problem, which, on some occasions might not be the case. Studying on a stochastic setting, which may include randomness in problem parameters such as the expected time spent in sites, or travel times, could be interesting as also mentioned in [6]. In addition, implementing a similar approach proposed in this study to the other variant of the MCRP introduced in [6], which aims to minimize the maximum node duration, by again applying a decomposition methodology, dividing the problem into two sub-problems, with one including the first two phase, and one including the last phase (which is going to be a variant of the well-known all pairs shortest path problem (APSP) where the start point, the coordination center, is fixed) may lead to improved results as well since there are also algorithms providing high-quality solutions to the problem [32] . Also, applying an adaptive method to decide on the tabu tenure, i.e., changing the tabu tenure value in between the moves of the algorithm depending on some factors, could lead to improved results. What is more, future research can also focus on introducing a new method, prediction for example, while obtaining the results for the sub-problems to decrease the runtime or to increase the solution quality of the algorithm.

APPENDIX A

OBJECTIVE VALUES OF EACH INSTANCE AMONGST DIFFERENT BENCHMARK ALGORITHMS

In Appendix A, the TS_C column will represent the objective values found by the algorithm proposed in this study, MTZ column will show the solutions obtained by CPLEX with MTZ formulation in 1 hour, TS_I column will represent the solutions found by the proposed benchmark algorithm, TS_L column will represent the solutions found by the variant introduced in [6], TS_B column shows the best objective value of the existing models before this paper (i. e., the minimum of the objective values of TS_I and TS_L) and the Warmed up CPLEX column will show the solutions obtained from warmed up CPLEX method. The Δ_{obj}^{MTZ} column will represent the objective value difference in percentages between TS_C and MTZ formulation, Δ_{obj}^{I} column will represent the objective value difference in percentages between TS_C and TS_I, Δ_{obj}^{L} column will represent the objective value difference in percentages between TS_C and TS_L, and similarly, Δ_{obj}^{B} column will represent the objective value difference in percentages between TS_C and TS_B, as well as $\Delta_{obj}^{wuCPLEX}$ column will represent the objective value difference in percentages between TS_C and warmed up CPLEX procedure.

The comparison of TS_C with respect to MTZ formulation in CPLEX solved for 1 hour (obtained from [6]) can be seen in Table 8. The comparison of TS_C with respect to short runs of the algorithms can be seen in Table 9 and Table 10. The comparison of TS_C with respect to long runs of the algorithms can be seen in Table 11 and Table 12. And the comparison of TS_C with respect to warmed up CPLEX method can be seen in Table 13.

It also needs to be mentioned that a rounding error is noticed in [6], since the optimal objective value found by CPLEX of Instance No. 4, 25/RC/2/L, is 49.18 rounded to two decimal points, is reported as 49.1, and the optimal objective value found by CPLEX of Instance No. 3, 25/R/2/H, is 123.41 rounded to two decimal points, is reported as 123.5 in [6]. With the best intentions, it is believed that this is because the route durations inside the problem is also rounded, leading up to an information loss. Thus, the values for the 3^{rd} and 4^{th} instance is going to be reported as 123.4 and 49.2 for the solutions obtained from [6].

Table 8: TS_C versus MTZ formulation(1h)

Ins. No.	Ins. Name	TS_C	MTZ	Δ_{obj}^{MTZ}
1	25/R/2/L	74.8	74.8	0.0%
2	25/R/2/M	103.6	101.6	2.0%
3	25/R/2/H	123.4	123.4	0.0%
4	25/RC/2/L	49.2	49.2	0.0%
5	25/RC/2/M	82.6	82.5	0.1%
6	25/RC/2/H	104.0	103.9	0.1%
7	25/R/3/L	63.4	63.3	0.2%
8	25/R/3/M	77.5	77.4	0.1%
9	25/R/3/H	96.2	95.2	1.1%
10	25/RC/3/L	70.1	70	0.1%
11	25/RC/3/M	82.6	82.5	0.1%
12	25/RC/3/H	90.3	90.2	0.1%
13	50/R/3/L	79.2	87.6	-9.6%
14	50/R/3/M	97.3	107.4	-9.4%
15	50/R/3/H	130.7	140.3	-6.8%
16	50/RC/3/L	74.1	74.2	-0.1%
17	50/RC/3/M	94.3	101.3	-6.9%
18	50/RC/3/H	125.2	138.4	-9.5%
19	50/R/4/L	84.3	91.3	-7.7%
20	50/R/4/M	94.3	115.6	-18.4%
21	50/R/4/H	104.5	163.2	-36.0%
22	50/RC/4/L	97.9	94.5	3.6%
23	50/RC/4/M	100.2	101.9	-1.7%
24	50/RC/4/H	112.9	127	-11.1%
25	75/R/3/L	70.4	85.9	-18.0%
26	75/R/3/M	99.5	191.1	-47.9%
27	75/R/3/H	147.4	232.9	-36.7%
28	75/RC/3/L	98.4	117.1	-16.0%
29	75/RC/3/M	127.1	180.5	-29.6%
30	75/RC/3/H	152.4	443.2	-65.6%
31	75/R/4/L	75.9	128.4	-40.9%
32	75/R/4/M	83.1	167.9	-50.5%
33	75/R/4/H	101.1	191.2	-47.1%
34	75/RC/4/L	94.2	104	-9.4%
35	75/RC/4/M	100.9	136.7	-26.2%
36	75/RC/4/H	118.6	195.8	-39.4%
37	100/R/3/L	72.8	100.9	-27.8%
38	100/R/3/M	110.7	245.1	-54.8%
39	100/R/3/H	155.8	311.3	-50.0%
40	100/RC/3/L	79.5	98.4	-19.2%
41	100/RC/3/M	112.9	249.2	-54.7%
42	100/RC/3/H	149.1	332.9	-55.2%
43	100/R/6/L	70.3	195.5	-64.0%
44	100/R/6/M	81.3	258.1	-68.5%
45	100/R/6/H	91.5	238.8	-61.7%
46	100/RC/6/L	82.5	156.1	-47.1%
47	100/RC/6/M	104.0	235.6	-55.9%
48	100/RC/6/H	118.6	445.7	-73.4%
	Min. Δ_{obj}			-73.4%
	Avg. Δ_{obj}			**-24.4%**
	Max. Δ_{obj}			3.6%
	Avg. CPU(s)	630.73	3600.00	

Table 9: TS_C versus Short Runs of TS_I and TS_L

Ins. No.	Ins. Name	TS_C	TS_I	Δ^I_{obj}	TS_L	Δ^L_{obj}
1	25/R/2/L	74.8	74.8	0.0%	74.8	0.0%
2	25/R/2/M	103.6	103.2	0.4%	103.2	0.4%
3	25/R/2/H	123.4	123.4	0.0%	123.4	0.0%
4	25/RC/2/L	49.2	49.2	0.0%	49.2	0.0%
5	25/RC/2/M	82.6	82.7	-0.1%	82.5	0.1%
6	25/RC/2/H	104.0	103.9	0.1%	105.5	-1.4%
7	25/R/3/L	63.4	63.3	0.2%	63.3	0.2%
8	25/R/3/M	77.5	77.4	0.1%	80.3	-3.5%
9	25/R/3/H	96.2	96.6	-0.4%	97.3	-1.1%
10	25/RC/3/L	70.1	70	0.1%	70	0.1%
11	25/RC/3/M	82.6	81.4	1.5%	82.5	0.1%
12	25/RC/3/H	90.3	90.2	0.1%	90.2	0.1%
13	50/R/3/L	79.2	80.8	-2.0%	80.7	-1.9%
14	50/R/3/M	97.3	103.8	-6.3%	103.9	-6.4%
15	50/R/3/H	130.7	131.3	-0.5%	131.2	-0.4%
16	50/RC/3/L	74.1	73	1.5%	73	1.5%
17	50/RC/3/M	94.3	96.3	-2.1%	97.5	-3.3%
18	50/RC/3/H	125.2	124.1	0.9%	123.4	1.5%
19	50/R/4/L	84.3	87.3	-3.4%	86.8	-2.9%
20	50/R/4/M	94.3	97.3	-3.1%	99.1	-4.8%
21	50/R/4/H	104.5	107.6	-2.9%	108	-3.2%
22	50/RC/4/L	97.9	94.5	3.6%	94.5	3.6%
23	50/RC/4/M	100.2	101.9	-1.7%	101.9	-1.7%
24	50/RC/4/H	112.9	114.2	-1.1%	114.4	-1.3%
25	75/R/3/L	70.4	76.9	-8.5%	75.4	-6.6%
26	75/R/3/M	99.5	107.9	-7.8%	105.8	-6.0%
27	75/R/3/H	147.4	147.7	-0.2%	146.3	0.8%
28	75/RC/3/L	98.4	102.3	-3.8%	101.8	-3.3%
29	75/RC/3/M	127.1	133.1	-4.5%	135.8	-6.4%
30	75/RC/3/H	152.4	154.1	-1.1%	154.3	-1.2%
31	75/R/4/L	75.9	79.3	-4.3%	78.7	-3.6%
32	75/R/4/M	83.1	88.9	-6.5%	88.8	-6.4%
33	75/R/4/H	101.1	102.9	-1.7%	101.8	-0.7%
34	75/RC/4/L	94.2	92.1	2.3%	92.2	2.2%
35	75/RC/4/M	100.9	104.5	-3.4%	104.5	-3.4%
36	75/RC/4/H	118.6	118.6	0.0%	118.6	0.0%
37	100/R/3/L	72.8	79.2	-8.1%	80.4	-9.5%
38	100/R/3/M	110.7	110.4	0.3%	110.8	-0.1%
39	100/R/3/H	155.8	154.4	0.9%	153.8	1.3%
40	100/RC/3/L	79.5	79.7	-0.3%	82.8	-4.0%
41	100/RC/3/M	112.9	121	-6.7%	120.2	-6.1%
42	100/RC/3/H	149.1	157.2	-5.2%	155.5	-4.1%
43	100/R/6/L	70.3	74	-5.0%	73.3	-4.1%
44	100/R/6/M	81.3	83.1	-2.2%	82.3	-1.2%
45	100/R/6/H	91.5	94.2	-2.9%	92.8	-1.4%
46	100/RC/6/L	82.5	84.6	-2.5%	84.8	-2.7%
47	100/RC/6/M	104.0	103.2	0.8%	103.6	0.4%
48	100/RC/6/H	118.6	119.5	-0.8%	119.5	-0.8%
	Min. Δ_{obj}			-8.5%		-9.5%
	Avg. Δ_{obj}			**-1.8%**		**-1.9%**
	Max. Δ_{obj}			3.6%		3.6%
	Min. CPU(s)	44.41	66.28		66.6	
	Avg. CPU(s)	630.73	390.25		405.32	
	Max. CPU(s)	1491.61	851.89		904.96	

Table 10: TS_C versus Short Run TS_B

Ins. No.	Ins. Name	TS_C	TS_B	Δ^B_{obj}
1	25/R/2/L	74.8	74.8	0.0%
2	25/R/2/M	103.6	103.2	0.4%
3	25/R/2/H	123.4	123.4	0.0%
4	25/RC/2/L	49.2	49.2	0.0%
5	25/RC/2/M	82.6	82.5	0.1%
6	25/RC/2/H	104.0	103.9	0.1%
7	25/R/3/L	63.4	63.3	0.2%
8	25/R/3/M	77.5	77.4	0.1%
9	25/R/3/H	96.2	96.6	-0.4%
10	25/RC/3/L	70.1	70	0.1%
11	25/RC/3/M	82.6	81.4	1.5%
12	25/RC/3/H	90.3	90.2	0.1%
13	50/R/3/L	79.2	80.7	-1.9%
14	50/R/3/M	97.3	103.8	-6.3%
15	50/R/3/H	130.7	131.2	-0.4%
16	50/RC/3/L	74.1	73	1.5%
17	50/RC/3/M	94.3	96.3	-2.1%
18	50/RC/3/H	125.2	123.4	1.5%
19	50/R/4/L	84.3	86.8	-2.9%
20	50/R/4/M	94.3	97.3	-3.1%
21	50/R/4/H	104.5	107.6	-2.9%
22	50/RC/4/L	97.9	94.5	3.6%
23	50/RC/4/M	100.2	101.9	-1.7%
24	50/RC/4/H	112.9	114.2	-1.1%
25	75/R/3/L	70.4	75.4	-6.6%
26	75/R/3/M	99.5	105.8	-6.0%
27	75/R/3/H	147.4	146.3	0.8%
28	75/RC/3/L	98.4	101.8	-3.3%
29	75/RC/3/M	127.1	133.1	-4.5%
30	75/RC/3/H	152.4	154.1	-1.1%
31	75/R/4/L	75.9	78.7	-3.6%
32	75/R/4/M	83.1	88.8	-6.4%
33	75/R/4/H	101.1	101.8	-0.7%
34	75/RC/4/L	94.2	92.1	2.3%
35	75/RC/4/M	100.9	104.5	-3.4%
36	75/RC/4/H	118.6	118.6	0.0%
37	100/R/3/L	72.8	79.2	-8.1%
38	100/R/3/M	110.7	110.4	0.3%
39	100/R/3/H	155.8	153.8	1.3%
40	100/RC/3/L	79.5	79.7	-0.3%
41	100/RC/3/M	112.9	120.2	-6.1%
42	100/RC/3/H	149.1	155.5	-4.1%
43	100/R/6/L	70.3	73.3	-4.1%
44	100/R/6/M	81.3	82.3	-1.2%
45	100/R/6/H	91.5	92.8	-1.4%
46	100/RC/6/L	82.5	84.6	-2.5%
47	100/RC/6/M	104.0	103.2	0.8%
48	100/RC/6/H	118.6	119.5	-0.8%
	Min. Δ_{obj}			-8.1%
	Avg. Δ_{obj}			**-1.5%**
	Max. Δ_{obj}			3.6%

Table 11: TS_C versus Long Runs of TS_I and TS_L

Ins. No.	Ins. Name	TS_C	TS_I	Δ^I_{obj}	TS_L	Δ^L_{obj}
1	25/R/2/L	74.8	74.8	0.0%	74.8	0.0%
2	25/R/2/M	103.6	101.6	2.0%	101.6	2.0%
3	25/R/2/H	123.4	123.4	0.0%	123.4	0.0%
4	25/RC/2/L	49.2	49.2	0.0%	49.2	0.0%
5	25/RC/2/M	82.6	82.5	0.1%	82.5	0.1%
6	25/RC/2/H	104.0	103.9	0.1%	103.9	0.1%
7	25/R/3/L	63.4	63.3	0.2%	63.3	0.2%
8	25/R/3/M	77.5	77.4	0.1%	77.4	0.1%
9	25/R/3/H	96.2	96.2	0.0%	96.6	-0.4%
10	25/RC/3/L	70.1	70	0.1%	70	0.1%
11	25/RC/3/M	82.6	81.4	1.5%	81.8	1.0%
12	25/RC/3/H	90.3	90.2	0.1%	90.2	0.1%
13	50/R/3/L	79.2	79.1	0.1%	78.9	0.4%
14	50/R/3/M	97.3	101.5	-4.1%	99.6	-2.3%
15	50/R/3/H	130.7	129.4	1.0%	131	-0.2%
16	50/RC/3/L	74.1	73	1.5%	73	1.5%
17	50/RC/3/M	94.3	96	-1.8%	94.6	-0.3%
18	50/RC/3/H	125.2	123.4	1.5%	123.4	1.5%
19	50/R/4/L	84.3	86.3	-2.3%	86.6	-2.7%
20	50/R/4/M	94.3	97.3	-3.1%	97.8	-3.6%
21	50/R/4/H	104.5	104.5	0.0%	106	-1.4%
22	50/RC/4/L	97.9	94.5	3.6%	94.5	3.6%
23	50/RC/4/M	100.2	101.9	-1.7%	101.9	-1.7%
24	50/RC/4/H	112.9	114	-1.0%	114.1	-1.1%
25	75/R/3/L	70.4	75.3	-6.5%	75.3	-6.5%
26	75/R/3/M	99.5	107.2	-7.2%	105.8	-6.0%
27	75/R/3/H	147.4	147	0.3%	146.3	0.8%
28	75/RC/3/L	98.4	100.6	-2.2%	100.9	-2.5%
29	75/RC/3/M	127.1	133.1	-4.5%	134.3	-5.4%
30	75/RC/3/H	152.4	153.9	-1.0%	154	-1.0%
31	75/R/4/L	75.9	78.7	-3.6%	77.8	-2.4%
32	75/R/4/M	83.1	88.4	-6.0%	88.3	-5.9%
33	75/R/4/H	101.1	101.8	-0.7%	100.7	0.4%
34	75/RC/4/L	94.2	92.1	2.3%	92.1	2.3%
35	75/RC/4/M	100.9	104.5	-3.4%	104.5	-3.4%
36	75/RC/4/H	118.6	118.6	0.0%	118.6	0.0%
37	100/R/3/L	72.8	77.4	-5.9%	79.2	-8.1%
38	100/R/3/M	110.7	110.3	0.4%	109.4	1.2%
39	100/R/3/H	155.8	153.3	1.6%	152.1	2.4%
40	100/RC/3/L	79.5	79.7	-0.3%	80.6	-1.4%
41	100/RC/3/M	112.9	119.3	-5.4%	119.7	-5.7%
42	100/RC/3/H	149.1	155.1	-3.9%	155.5	-4.1%
43	100/R/6/L	70.3	73.7	-4.6%	73.3	-4.1%
44	100/R/6/M	81.3	82.3	-1.2%	82.3	-1.2%
45	100/R/6/H	91.5	93.7	-2.3%	93.7	-2.3%
46	100/RC/6/L	82.5	83.5	-1.2%	84.5	-2.4%
47	100/RC/6/M	104.0	103.2	0.8%	103.6	0.4%
48	100/RC/6/H	118.6	119.4	-0.7%	119.5	-0.8%
	Min. Δ_{obj}			-7.2%		-8.1%
	Avg. Δ_{obj}			**-1.2%**		**-1.2%**
	Max. Δ_{obj}			3.6%		3.6%
	Min. CPU(s)	44.41	491.26		465.34	
	Avg. CPU(s)	630.73	2411.6		2317.04	
	Max. CPU(s)	1491.61	3600.00		3600.00	

Table 12: TS_C versus Long Run TS_B

Ins. No.	Ins. Name	TS_C	TS_B	Δ_{obj}^{B}
1	25/R/2/L	74.8	74.8	0.0%
2	25/R/2/M	103.6	101.6	2.0%
3	25/R/2/H	123.4	123.4	0.0%
4	25/RC/2/L	49.2	49.2	0.0%
5	25/RC/2/M	82.6	82.5	0.1%
6	25/RC/2/H	104.0	103.9	0.1%
7	25/R/3/L	63.4	63.3	0.2%
8	25/R/3/M	77.5	77.4	0.1%
9	25/R/3/H	96.2	96.2	0.0%
10	25/RC/3/L	70.1	70	0.1%
11	25/RC/3/M	82.6	81.4	1.5%
12	25/RC/3/H	90.3	90.2	0.1%
13	50/R/3/L	79.2	78.9	0.4%
14	50/R/3/M	97.3	99.6	-2.3%
15	50/R/3/H	130.7	129.4	1.0%
16	50/RC/3/L	74.1	73	1.5%
17	50/RC/3/M	94.3	94.6	-0.3%
18	50/RC/3/H	125.2	123.4	1.5%
19	50/R/4/L	84.3	86.3	-2.3%
20	50/R/4/M	94.3	97.3	-3.1%
21	50/R/4/H	104.5	104.5	0.0%
22	50/RC/4/L	97.9	94.5	3.6%
23	50/RC/4/M	100.2	101.9	-1.7%
24	50/RC/4/H	112.9	114	-1.0%
25	75/R/3/L	70.4	75.3	-6.5%
26	75/R/3/M	99.5	105.8	-6.0%
27	75/R/3/H	147.4	146.3	0.8%
28	75/RC/3/L	98.4	100.6	-2.2%
29	75/RC/3/M	127.1	133.1	-4.5%
30	75/RC/3/H	152.4	153.9	-1.0%
31	75/R/4/L	75.9	77.8	-2.4%
32	75/R/4/M	83.1	88.3	-5.9%
33	75/R/4/H	101.1	100.7	0.4%
34	75/RC/4/L	94.2	92.1	2.3%
35	75/RC/4/M	100.9	104.5	-3.4%
36	75/RC/4/H	118.6	118.6	0.0%
37	100/R/3/L	72.8	77.4	-5.9%
38	100/R/3/M	110.7	109.4	1.2%
39	100/R/3/H	155.8	152.1	2.4%
40	100/RC/3/L	79.5	79.7	-0.3%
41	100/RC/3/M	112.9	119.3	-5.4%
42	100/RC/3/H	149.1	155.1	-3.9%
43	100/R/6/L	70.3	73.3	-4.1%
44	100/R/6/M	81.3	82.3	-1.2%
45	100/R/6/H	91.5	93.7	-2.3%
46	100/RC/6/L	82.5	83.5	-1.2%
47	100/RC/6/M	104.0	103.2	0.8%
48	100/RC/6/H	118.6	119.4	-0.7%
	Min. Δ_{obj}			-6.5%
	Avg. Δ_{obj}			**-1.0%**
	Max. Δ_{obj}			3.6%

Table 13: TS_C versus Warmed up CPLEX

Ins. No.	Ins. Name	TS_C	Warmed up CPLEX	$\Delta_{obj}^{wuCPLEX}$
1	25/R/2/L	74.8	74.8	0.0%
2	25/R/2/M	103.6	101.6	2.0%
3	25/R/2/H	123.4	123.4	0.0%
4	25/RC/2/L	49.2	49.2	0.0%
5	25/RC/2/M	82.6	82.5	0.1%
6	25/RC/2/H	104.0	103.9	0.1%
7	25/R/3/L	63.4	63.3	0.2%
8	25/R/3/M	77.5	77.4	0.1%
9	25/R/3/H	96.2	95.2	1.1%
10	25/RC/3/L	70.1	70	0.1%
11	25/RC/3/M	82.6	80.9	2.1%
12	25/RC/3/H	90.3	90.2	0.1%
13	50/R/3/L	79.2	78.4	1.0%
14	50/R/3/M	97.3	100	-2.7%
15	50/R/3/H	130.7	128.6	1.6%
16	50/RC/3/L	74.1	72.2	2.6%
17	50/RC/3/M	94.3	95.1	-0.8%
18	50/RC/3/H	125.2	123.4	1.5%
19	50/R/4/L	84.3	85.1	-0.9%
20	50/R/4/M	94.3	97.3	-3.1%
21	50/R/4/H	104.5	107.5	-2.8%
22	50/RC/4/L	97.9	94.5	3.6%
23	50/RC/4/M	100.2	101.9	-1.7%
24	50/RC/4/H	112.9	113.7	-0.7%
25	75/R/3/L	70.4	74.9	-6.0%
26	75/R/3/M	99.5	103.6	-4.0%
27	75/R/3/H	147.4	145	1.7%
28	75/RC/3/L	98.4	101.2	-2.8%
29	75/RC/3/M	127.1	133.1	-4.5%
30	75/RC/3/H	152.4	153.6	-0.8%
31	75/R/4/L	75.9	78.6	-3.4%
32	75/R/4/M	83.1	87	-4.5%
33	75/R/4/H	101.1	101.2	-0.1%
34	75/RC/4/L	94.2	92.1	2.3%
35	75/RC/4/M	100.9	104.5	-3.4%
36	75/RC/4/H	118.6	118.6	0.0%
37	100/R/3/L	72.8	77.4	-5.9%
38	100/R/3/M	110.7	108.6	1.9%
39	100/R/3/H	155.8	151.2	3.0%
40	100/RC/3/L	79.5	79.7	-0.3%
41	100/RC/3/M	112.9	118.7	-4.9%
42	100/RC/3/H	149.1	152.1	-2.0%
43	100/R/6/L	70.3	73.3	-4.1%
44	100/R/6/M	81.3	82.3	-1.2%
45	100/R/6/H	91.5	92.8	-1.4%
46	100/RC/6/L	82.5	84.6	-2.5%
47	100/RC/6/M	104.0	103.2	0.8%
48	100/RC/6/H	118.6	119.5	-0.8%
	Min Δ_{obj}			-6.0%
	Avg. Δ_{obj}			**-0.8%**
	Max. Δ_{obj}			3.6%
	Min. CPU(s)	44.41	7200.00+	
	Avg. CPU(s)	630.73	7200.00+	
	Max. CPU(s)	1491.61	7200.00+	

APPENDIX B

AVERAGE OBJECTIVE VALUES ACCORDING TO CHARACTERISTICS OF THE INSTANCES

Distribution Type	TS_C	TS_I S.Run	TS_I L.Run	Warmed up CPLEX
Random	104.8	107.1	106.3	105.5
Random-clustered	91.5	92.8	92.4	92.3

Table 14: Average objective values according to distribution amongst the network

Team Level	TS_C	TS_I S.Run	TS_I L.Run	Warmed up CPLEX
Low	95.4	97.7	96.8	96.2
High	100.9	102.1	101.8	101.6

Table 15: Average objective values according to team number level

Please note that, each instance with a fixed number of nodes, fixed type of distribution amongst the network, and fixed level of coverage ratio have 2 different levels of team number. Thus, the team number level can be divided into 2; low and high. Low level of teams can mean 2 for the instances with 25 nodes, and can mean 3 for the instances with 50 nodes, and so on.

Coverage Ratio Level	TS_C	TS_I S.Run	TS_I L.Run	Warmed up CPLEX
Low	77.3	78.8	78.2	78.1
Medium	97.0	99.8	99.2	98.6
High	120.1	121.2	120.5	120.0

Table 16: Average objective values according to coverage ratio level

APPENDIX C

LINEAR REGRESSION GRAPHICS OF OBJECTIVE VALUE DIFFERENCE AMONGST DIFFERENT BENCHMARK ALGORITHMS

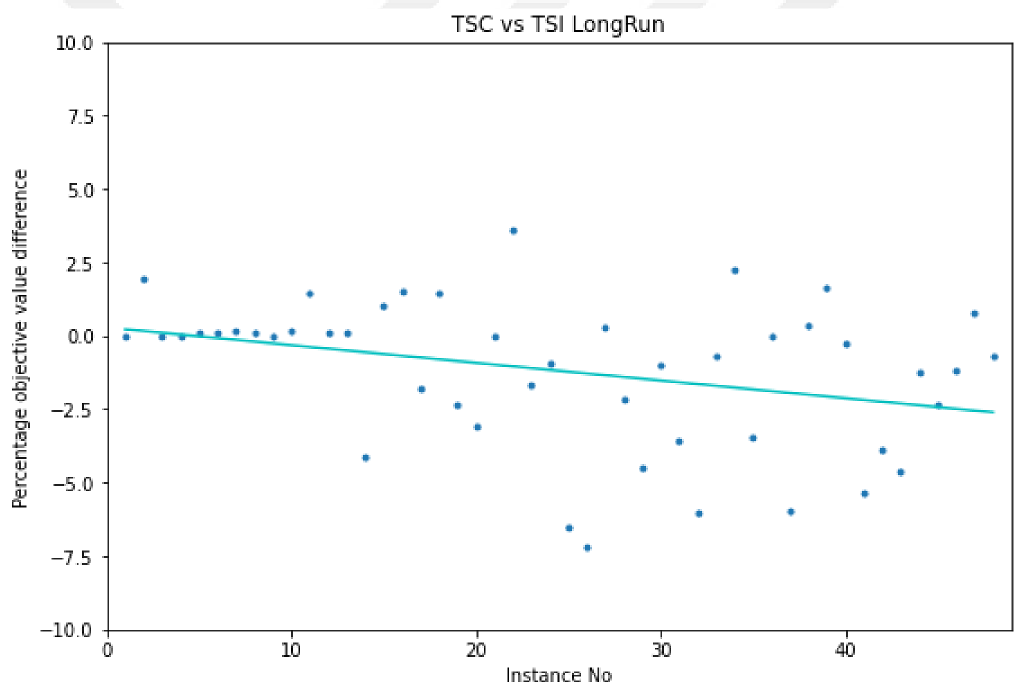

Figure 11: Objective value difference (%) for each instance vs. TS_I LongRun

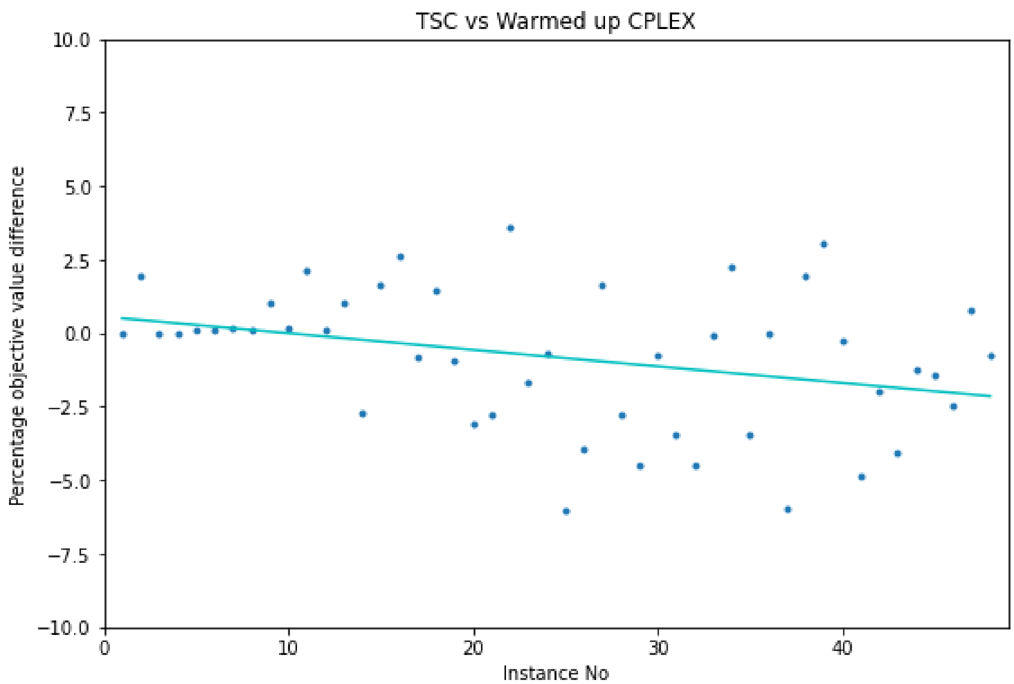

Figure 12: Objective value difference (%) for each instance vs. Warmed up CPLEX

It can be seen that the claim that when the instance size grows up, the improvement in the objective value difference tend to grow up, as mentioned earlier, still holds when TS_C is compared with different methods as well. Also, the p-value for the regressions in Fig. 11 and Fig. 12 are 0.019 and 0.022, respectively.

Bibliography

[1] T. assessment capacities project (ACAPS), "Humanitarian needs assessment: The good enough guide," 2014.

[2] X. Li, X. Liu, H. Ma, and S. Hu, "Integrated routing optimization for post-disaster rapid-detailed need assessment," *International Journal of General Systems*, vol. 49, no. 5, pp. 521–545, 2020.

[3] M. Arii, "Rapid assessment in disasters," *Japan Medical Association*, vol. 56, no. 1, pp. 19–24, 2013.

[4] I. F. of Red Cross and R. C. S. (IFRC), "Guidelines for assessment in emergencies," 2008.

[5] T. assessment capacities project (ACAPS), "Technical brief: purposive sampling and site selection in phase 2," 2011.

[6] D. Pamukcu and B. Balcik, "A multi-cover routing problem for planning rapid needs assessment under different information-sharing settings," *OR Spectrum*, vol. 42, no. 1, pp. 1–42, 2020.

[7] I.-A. S. Committee, "Multi-cluster/sector initial rapid assessment (MIRA)," 2012.

[8] R. Barney, "Five ideas... for efficient, humanitarian, satcom networks," 2018.

[9] B. S. Manoj and A. Hubenko, "Communication challenges in emergency response," *Communications of the ACM*, vol. 50, no. 3, pp. 51–53, 2007.

[10] M. Han and Y. Wang, "A survey for vehicle routing problems and its derivatives," *IOP Conference Series: Materials Science and Engineering*, vol. 452, no. 4, 2018.

[11] L. E. Torre, I. S. Dolinskaya, and K. R. Smilowitz, "Disaster relief routing: integrating research and practice," *Socio-Economic Planning Sciences*, vol. 46, no. 1, pp. 88–97, 2012.

[12] L. R. Garces, M. D. Pido, R. S. Pomeroy, S. Koeshendrajana, B. I. Prisantoso, N. A. Fatan, D. Adhuri, T. Raiful, S. Rizal, A. Tewfik, and M. Dey, "Rapid assessment of community needs and fisheries status in tsunami-affected communities in aceh province, indonesia," *Ocean Coastal Management*, vol. 53, no. 2, pp. 69–79, 2010.

[13] B. Balcik, "Site selection and vehicle routing for post-disaster rapid needs assessment," *Transportation Research Part E: Logistics and Transportation Review*, vol. 101, pp. 30–58, 2017.

[14] W. P. Coutinho, J. Fliege, and M. Batarra, “Glider routing and trajectory optimisation in disaster assessment,” *European Journal of Operations Research*, vol. 274, no. 3, pp. 1138–1154, 2019.

[15] G. B. Alvarenga, G. R. Mateus, and G. Tomi, “A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows,” *Computers Operations Research*, vol. 34, no. 6, pp. 1561–1584, 2007.

[16] L. Tang and X. Wang, “Iterated local search algorithm based on very large-scale neighborhood for prize-collecting vehicle routing problem,” *The International Journal of Advanced Manufacturing Technology*, vol. 29, pp. 1246–1258, 2006.

[17] J.-F. Berube, M. Gendreau, and J.-Y. Potvin, “A branch-and-cut algorithm for the undirected prize collecting traveling salesman problem,” *Networks*, vol. 54, no. 1, pp. 56–67, 2009.

[18] M. Dell’Amico, F. Maffioli, and A. Sciomachen, “A lagrangian heuristic for the prize collection travelling salesman problem,” *Annals of Operations Research*, vol. 81, pp. 289–306, 1998.

[19] M. sans frontiers, “Assessment toolkit part i: practical steps for the assessment of health and humanitarian crises,” 2012.

[20] “Technical sheet: how to do a purposive sample,” 2014.

[21] C. E. Miller, A. W. Tucker, and R. A. Zemlin, “Integer programming formulation of traveling salesman problems,” *Journal of the ACM*, vol. 7, no. 4, 1960.

[22] G. Dantzig, R. Fulkerson, and S. Johnson, “Solution of a large-scale traveling-salesman problem,” *Journal of the Operations Research Society of America*, vol. 2, no. 4, pp. 393–410, 1954.

[23] F. Glover, “Future paths for integer programming and links to artificial intelligence,” *Computers Operations Research*, vol. 13, no. 5, pp. 533–549, 1986.

[24] S. Basu, “Tabu search implementation on traveling salesman problem and its variations: a literature survey,” *American Journal of Operations Research*, vol. 2, no. 2, pp. 163–173, 2012.

[25] G. Laporte, M. Gendreau, J.-Y. Potvin, and F. Semet, “Classical and modern heuristics for the vehicle routing problem,” *International Transactions in Operational Research*, vol. 7, no. 4-5, pp. 285–300, 2000.

[26] “Concorde TSP solver,” 2015.

[27] C. Rego, D. Gamboa, F. Glover, and C. Osterman, “Traveling salesman problem heuristics: leading methods, implementations and latest advances,” *European Journal of Operations Research*, vol. 211, no. 3, pp. 427–441, 2011.

[28] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations Research*, vol. 8, no. 1, pp. 101–111, 1960.

[29] M. Fallahi, S. Amiri, and M. Yaghini, "A parameter tuning methodology for metaheuristics based on design of experiments," *International Journal of Engineering Technology Sciences*, vol. 2, no. 6, pp. 497–521, 2014.

[30] E.-G. Talbi, *Metaheuristics: From Design to Implementation*, ch. Common concepts for metaheuristics. John Wiley & Sons, 2009.

[31] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, no. 2, pp. 163–324, 1987.

[32] T. M. Chan, "All-pairs shortest path with real weights in O($n^3/logn$) time," *Algorithmica*, vol. 50, pp. 236–243, 2007.

VITA

Yurtsev Mıhçıoğlu graduated from Kartal Anatolian High School in 2013. He received his B.Sc. degree in Industrial Engineering from Özyeğin University in January 2018 and continued his education with joining the Master of Science program in Industrial Engineering at Özyeğin University, under the supervision of Assistant Professor Erinç Albey. His main research areas are combinatorial optimization and heuristic & metaheuristic methods.