APPROXIMATION ALGORITHMS AND HEURISTICS FOR A HETEROGENEOUS TRAVELING SALESMAN PROBLEM

A Thesis

by

RAHUL RANGARAJAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2011

Major Subject: Mechanical Engineering

APPROXIMATION ALGORITHMS AND HEURISTICS FOR A HETEROGENEOUS TRAVELING SALESMAN PROBLEM

A Thesis

by

RAHUL RANGARAJAN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Sivakumar Rathinam
Committee Members,	Reza Langari
	Shankar P. Bhattacharyya
Head of Department,	Dennis O'Neal

May 2011

Major Subject: Mechanical Engineering

ABSTRACT

Approximation Algorithms and Heuristics for a Heterogeneous Traveling Salesman Problem. (May 2011)

Rahul Rangarajan, B.Tech., National Institute of Technology, Jalandhar

Chair of Advisory Committee: Dr. Sivakumar Rathinam

Unmanned Vehicles (UVs) are developed for several civil and military applications. For these applications, there is a need for multiple vehicles with different capabilities to visit and monitor a set of given targets. In such scenarios, routing problems arise naturally where there is a need to plan paths in order to optimally use resources and time. The focus of this thesis is to address a basic optimization problem that arises in this setting.

We consider a routing problem where some targets have to be visited by specific vehicles. We approach this problem by dividing the routing into two sub problems: partitioning the targets while satisfying vehicle target constraints and sequencing. We solve the partitioning problem with the help of a minimum spanning tree algorithm. We use 3 different approaches to solve the sequencing problem; namely, the 2 approximation algorithm, Christofide's algorithm and the Lin - Kernighan Heuristic (LKH). The approximation algorithms were implemented in MATLAB®. We also developed an integer programming (IP) model and a relaxed linear programming (LP) model in C++ with the help of Concert Technology for CPLEX, to obtain lower bounds.

We compare the performance of the developed approximation algorithms with both the IP and the LP model and found that the heuristic performed very well and provided the better quality solutions as compared to the approximation algorithms. It was also found that the approximation algorithms gave better solutions than the apriori guarantees.

To my supportive father Ranga and my late mom Ramya

ACKNOWLEDGMENTS

I wish to thank my advisor, Dr. Sivakumar Rathinam, for his constant support, motivation and guidance. I am grateful for the amount of faith he had in me and the patience he displayed as my mentor. I also wish to express my gratitude to Dr. Reza Langari and Dr. Shankar P. Bhattacharyya for being an important part of my committee.

I thank my parents and my sister for the constant support they offered. I also wish to thank my friends Amrish and Riddhi for helping me to refine my ideas and concepts. Finally, I would like to thank everyone who supported me directly or indirectly during the course of this work.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

One of the most challenging and widely studied problems in the field of optimization is the Traveling Salesman Problem (TSP). The TSP is a very simple problem to describe but a difficult problem to solve, which is why it has received so much attention from the scientific community. The TSP arises in several real life situations like path planning for an unmanned aerial vehicle (UAV) [1], chip manufacturing [2], job sequencing [3] etc. In the following section we will describe the TSP.

1.1 TSP

Before we state what a TSP is we need to define terms like *tour* and *depot*. The initial position at which the vehicle is located is called the *depot*. The sequence in which the vehicle visits the targets is called a *tour*. The TSP is defined in the following paragraph.

Given a list of n targets and the distances between each of the targets, the aim of the TSP is to find a tour such that each of target is visited exactly once and the sum of the distances traveled by the vehicle is minimum. The TSP belongs to a set of problems which are called NP hard [4]. Specifically, if we can find a polynomial time algorithm to solve the TSP in polynomial time, then we can find an algorithm for all other problems that are NP hard.

The journal model is *IEEE/ACM Transactions on Networking*.

1.2 Multiple depot multiple TSP

A multiple depot, multiple TSP (MDMTSP) is a variation of the TSP where there is a set of distinct depots and a vehicle is located at each depot. The aim of the MDMTSP is to find the tour for each vehicle such that each target is visited once and the total distance traveled by all the vehicles are minimized. The focus of this research is to address the MDMTSP with additional vehicle target constraints, while keeping the cost of travel between two targets the same for all vehicles. We call this the multiple depot multiple heterogeneous TSP (MDMHTSP).

Specifically, we plan to consider targets that can be classified into distinct sets as follows: the first set which is the set of common targets that can be visited by any vehicle; for each vehicle there is a distinct set of targets that have to be visited by the vehicle. We aim to develop approximation algorithms and heuristics to obtain feasible solutions for the MDMHTSP in polynomial time and provide a guarantee for the quality of the solution in the case of approximation algorithms. We also present a detailed computational study comparing the performances of the two approximation algorithms and the heuristic.

An overview of the rest of this thesis is as follows. In chapter II, we will discuss and review the literature on methods of solving combinatorial problems like the MDMHTSP. Chapter III, deals with formulation of the MDMHTSP and the integer programming model. In chapter IV, we present the two approximation algorithms and the heuristic. We also present the proof for the two approximation algorithms. In chapter V, we explain the simulations and perform a computational study. We compare the performances of the algorithms and heuristic developed based on the results of the computational study. We conclude the thesis in chapter VI by presenting a summary of the work done and the possible directions for further research.

CHAPTER II

LITERATURE REVIEW

The importance of the TSP is that it arises in several practical applications and belongs to a class of hard combinatorial problems referred to as NP hard problems in the literature. The MDMHTSP being a variation of the TSP is also NP hard. In the following section we outline different methods of solving combinatorial problems.

2.1 Methods for solving combinatorial problems

The most common methods for solving a difficult combinatorial problem such as the MDMHTSP can be classified as follows: exact algorithms, heuristics, approximation algorithms and transformation methods [5, 6].

Exact algorithms are those that provide optimal solutions to the given problem, however, there is no guarantee on the running time of the algorithm [7]. Heuristics are algorithms that find feasible solutions, obtained in polynomial time but have no guarantee on the quality of the solution [8]. Approximation algorithms are those algorithms that have a polynomial running time and return a feasible solution that is a certain factor away from the optimal solution for any instance of the problem [9]. Transformation methods are those in which the given problem is transformed to another standard problem such as the TSP which has efficient methods of finding a good solution [5].

2.1.1 Combinatorial problems in MDMHTSP

There are three combinatorial problems while dealing with the MDMHTSP: partitioning the common set of targets to be assigned to each vehicle, the second deals with determining the order in which the assigned targets have to be visited and the last being able to satisfy the vehicle target constraints. The difficult part in dealing with MDMHTSP is that all these three problems are coupled with each other. Since this is a coupled problem we plan to solve the problem in stages which will be explained more in detail in chapter III and IV. In the next section, we will review the existing literature on approximation algorithms related to the MDMHTSP.

2.2 Approximation algorithms

As mentioned in the previous section an α - approximation algorithm provides a feasible solution in polynomial time that is α times away from the optimal solution for any instance of the problem [10, 11]. Presently, it is known that constant factor approximation algorithms for the TSP exists only if the costs satisfy the triangle inequality unless $P = NP$ [12]. So we will assume that all the costs to satisfy the triangle inequality unless explicitly mentioned.

2.2.1 Single vehicle problems

The $\frac{3}{2}$ - approximation algorithm by Christofides is the best known approximation algorithm for a single TSP [13]. The Christofides algorithm provides a feasible solution by combining the minimum spanning tree (MST) with a weighted non bipartite minimum cost perfect matching of the od degree nodes of the MST. There is also the 2 - approximation algorithms for the single TSP, in which the minimum spanning tree is doubled to obtain a feasible solution for the single TSP.

Another variation of the single TSP is the single depot hamiltonian path problem (HPP) in which the vehicle starts from the depot and visits a set of targets before reching the terminal point [14]. A $\frac{5}{3}$ - approximation algorithm has been developed for the single depot single terminal HPP as mentioned in [15] and [16]. For an asymmetric HPP, Chekuri et al., in [17] present an approximation algorithm that runs in $O(log$ $n)$ steps.

2.2.2 Multiple vehicle problems

In [18], Malik et al., develop a 2 - approximation algorithm for a symmetric generalized MDMTSP where they obtain the feasible solution using a degree constrained MST. Rathinam et al., [19] also have developed an approximation algorithm for multiple vehicle systems that runs in $O(log\ (m+n)^2)$ steps (where m is the number of targets and n is the number of vehicles). There is also a 3 - approximation algorithm for a two depot TSP in which the authors use multi commodity flow formulation to partition the common targets and then use the christofides algorithm to obtain a tour among the allocated nodes [20]. Xu and Rodrigues in [21] have partially addressed the MDMTSP by developing a $\frac{3}{2}$ - approximation algorithm subject to the fact that the number of depots do not vary and is a constant number.

As seen in 2.2.1, the single HPP and the single TSP are closely related. Similarly, we will examine the various approximation algorithms for multiple depot HPPs. There exists a 2 - approximation algorithm for a multiple depot multiple terminal HPP as illustrated by Rathinam et al., in [22]. They compare the results with a lower bound that they obtain using a mataroid intersection algorithm. They have also developed a $\frac{3}{2}$ - approximation algorithm for two variants of a 2 depot HPP [23] provided the costs are symmetric. An 8 - approximation algorithm was developed by Yadlapalli et al., for a 2 depot heterogeneous HPP [24].

2.3 Heuristics

Integer linear programming models are available that help in solving the TSP as mentioned in [25]. However, there is no guarantee on the time in which they will solve the problem. Hence, there is a need for algorithms that run fast and give a guarantee on the running time. There are heuristics such as the LKH [8], some nature inspired genetic algorithms mentioned in [26] and some ant colony optimization methods such as [27] and [28] that run relatively fast and produce good quality solutions. It is important to stress that these heuristics still have no guarantee for solution quality.

CHAPTER III

PROBLEM FORMULATION

3.1 Formulation of the MDMHTSP

All the depots and the targets are called the vertices of the graph and all the paths joining these targets or depots are edges joining the corresponding vertices. Let $D = \{d_1, d_2, d_3, \ldots, d_m\}$ be the set of vertices corresponding to the m depots. Let $T = \{t_1, t_2, t_3, \ldots, t_n\}$ be the set of vertices corresponding to n targets. Further, for all $i \in \{1, 2, 3, \ldots, m\}$, the vehicle at depot d_i has to visit all the targets in $S_i \subseteq T$ owing to vehicle target constraints. Let $\bigcap_{i=1}^{i=m} S_i = \emptyset$. Let R be the common set of targets defined as $R = T \setminus \bigcup_{i=1}^{i=m} S_i$. Let $V_i = R \cup S_i \cup \{d_i\}$ be the set of vertices corresponding to vehicle i. Each vehicle starts from the corresponding depot visits a set of targets and returns to the same depot. The depots themselves are not connected by any edges. For all $i \in \{1, 2, 3, \ldots, m\}$, let E_i denote the set of edges that join any two vertices in V_i.

Let c_{ij} be the cost of travel from vertex i to vertex j. It is important to note that the cost to travel between any two vertices does not change with the vehicle used. Another assumption that was made is that the costs satisfy the triangle inequality (that is $c_{ij} + c_{jk} \geq c_{ik}$ for every $i, j, k \in \{T \cup D\}$).Let $G_i = (V_i, E_i, c)$, be the corresponding graph for vehicle i. Then, the combined graph that all the vehicles form together in order to find the shortest individual tours is

$$G = \bigcup_{i=1}^{i=m} G_i$$

Let $Tour_i$ and z_i be the tour and the number of vertices (other than the depot) visited by the i^{th} vehicle respectively. Then, the tour of the i^{th} vehicle be given by $Tour_i = (d_i, x_1^i, x_2^i, \ldots, x_{z_i}^i, d_i)$. The cost of the tour of the ith vehicle is given by $C_i = c_{d_i x_1^i} + \sum_{j=2}^{j=z_i} c_{x_{j-1}^i x_j^i} + c_{x_{z_i}^i d_i}$. The problem is to find all the individual tours such that each vertex is visited exactly once, all the vertices in S_i is visited by the vehicle at d_i and the total cost of travel, $\sum_{i=1}^{i=n} C_i$ is minimized.

3.2 Integer program formulation

To compare the solutions of our algorithms and the heuristic we pose the MDMTSP as a multi commodity flow problem so as to obtain the optimal solution and the lower bound.

Let p_{ij}^k be the flow of the k^{th} commodity from vertex i to vertex j. Let x_{ij} denote the binary variable that decides if the edge between vertex i and vertex j is used. $x_{ij} = 1$ if the edge between vertex i and vertex j is present in the tour of any vehicle and is equal to 0 otherwise. Let $D^{'}$ be the copy of the set of depots D. The individual tours will start from $d_i \in D$ and end at $d_i^{'} \in D^{'}$. Let $V = T \cup D \cup D'$ The following is the integer programming of the MDMHTSP:

$$C_{opt} = \min \sum_{x,p,\psi} c_{ij} x_{ij} \tag{3.1}$$

$$x_{ii} = 0 \forall i \in V \tag{3.2}$$

$$\sum_{j \in V} x_{ij} = 1 \quad \forall \quad i \in \{T \cup D\} \tag{3.3a}$$

$$\sum_{i \in V} x_{ij} = 1 \quad \forall \quad j \in \{T \cup D'\} \tag{3.3b}$$

$$\sum_{j \in \{T \cup D\}} x_{dj} \geq \psi_{di} \quad \forall \quad \begin{cases} d \in D \\ i \in T \cup D' \end{cases} \tag{3.4}$$

$$\sum_{j \in V} x_{d'j} = 0 \quad \forall \quad d' \in D' \tag{3.5a}$$

$$\sum_{i \in V} x_{jd} = 0 \quad \forall \quad d \in D \tag{3.5b}$$

$$\psi_{d_f i} = 1 \quad \forall \quad \begin{cases} d_f \in D \\ i \in S_f \end{cases} \tag{3.6}$$

$$\psi_{d_f d'_f} = 1 \quad \forall \quad \Big\{ \, d_f \in D \tag{3.7}$$

$$\sum_{j \in V} p^k_{dj} \geq \psi_{dk} \quad \forall \quad \begin{cases} k \in T \cup D' \\ d \in D \end{cases} \tag{3.8}$$

$$\sum_{i \in T \cup D} p^k_{ij} = \sum_{i \in T \cup D'} p^k_{ji} \quad \forall \quad \begin{cases} j, k \in T \cup D' \\ j \neq k \end{cases} \tag{3.9}$$

$$\sum_{i \in T \cup D} p^k_{ik} - \sum_{i \in T \cup D'} p^k_{ki} \geq 1 \quad \forall \quad k \in T \cup D' \tag{3.10}$$

$$p^k_{ij} \leq x_{ij} \quad \forall \quad \begin{cases} i, j \in V \\ k \in T \end{cases} \tag{3.11}$$

$$p^k_{ij} \geq 0 \tag{3.12}$$

$$x_{ij} \in \{0, 1\} \tag{3.13a}$$

$$\psi_{ij} \in \{0, 1\} \tag{3.13b}$$

The constraints (3.3a) and (3.3b) make sure that every target is visited by one vehicle only. The degree constraint (3.4) specify the number of edges connected to a depot visiting at least one target. The out degree constraint on the copy of the depot which is modeled as a terminal is enforced by the constraint (3.5a). Similarly, the in degree constraint of the depot is enforced by constraint (3.5b). The vehicle target constraint is incorporated by the constraint (3.6). All the tours that start from a depot d_f should end in the copy of the depot d'_f as shown in constraint (3.7). To make sure that the demand of the k^{th} target from depot d is satisfied only by depot d and no other depot (3.8). Constraint (3.9) makes sure that the amount of commodity flowing into the intermediate target flows out of it. Constraint (3.10) is the flow constraint for a terminal node where all the commodity should reach the terminal node. The capacity constraint that arises is enforced by constraint (3.11). (3.13a) and (3.13b) are the integer constraints. The non negativity constraint is enforced on the flow variable since we do not want negative flows in the model.

The above described is the integer program where the variables x_{ij} and ψ_{di} can take values 0 or 1. When we remove this constraint and let these variables take values between $\{0, 1\}$ we get the model for a relaxed problem. The solution of this relaxed problem can be used as a lower bound for the integer program.

CHAPTER IV

ALGORITHMS AND HEURISTIC

This chapter explains all the approximation algorithms and the heuristic developed. We have also presented the proofs for the two approximation algorithms developed. The MDMHTSP is solved using a two stage approach. In the first stage, the aim is to partition the common set of targets that could be visited by any vehicle exactly once. Once all the common set of targets are allocated to a particular vehicle, we also include the vehicle - target constraints and include the other targets to be visited. Finally, we find a feasible tour for each of the vehicles over their corresponding targets. After these partitions are obtained we further use approximation algorithms and LKH heuristic to arrive at a feasible solution for the MDMHTSP. The first stage of this approach is the partitioning step which is common to all the approximation algorithms and the heuristic and is described in the next section. The Christofide's algorithm and the Euler walk algorithm explained in Figs. 4.1 and 4.2 respectively.

4.1 Partitioning

1. Create a root node and connect all the depots to the root node with zero cost edges.

2. Find a MST over all the depots, the common targets and the root node. After this step we obtain a MST rooted at the root node.

3. Remove the zero cost edges and the root node. As a result, we get individual MSTs rooted at depots of each vehicle. This collection of MSTs is called a

minimum spanning forest (MSF). Thus, the partition P_i is the set of all common targets in the MST connected to the depot d_i. Therefore, the targets that are assigned to the vehicle at d_i is $P_i \cup s_i$.

Algorithm 1 Christofide's Algorithm

---

for Partition $R_i : i = 1 \leftarrow n$ **do**
  procedure Christofide(R_i)
  $MST \leftarrow$ Minimum spanning tree $(R_i \cup d_i)$ {MST over R_i and corresponding depot}
  for each vertex $v \in R_i \cup d_i$ **do**
    if degree is odd **then**
      $Match \leftarrow \{v\}$
    end if
  end for
  $Match \leftarrow$ **blossom**$(Match)$ {Minimum weight matching is done}
  $euleriangraph \leftarrow$ Add$(MST, Match)$
  $walk \leftarrow$ eulerwalk$(d_i, euleriangraph)$ {Refer to **Algorithm 2**}
  convert *walk* to *tour* by short-cutting
  end procedure
end for

Fig. 4.1. Christofide's algorithm

Algorithm 2 Euler Walk Algorithm

procedure eulerwalk(d_i, $euleriangraph$)

if $euleriangraph$ has no edges **then**

  return $euleriangraph$

else

  starting from d_i create a $walk$ of $euleriangraph$, never

  visiting the same edge twice until d_i is reached again

  $walk \leftarrow \{d_i, x_1^i, x_2^i, \ldots, x_{z_i}^i, d_i\}$

  DeleteEdges $[d_i, x_1^i], \ldots, [x_{z_i}^i, d_i]$ from $euleriangraph$

  return [eulerwalk(d_i),eulerwalk(x_1^i), $\ldots$,eulerwalk($x_{z_i}^i$), d_i]

end if

end procedure

Fig. 4.2. Euler walk algorithm

4.2 4 - Approximation algorithm for MDMHTSP

Algorithm 3 4 - Approximation Algorithm for MDMHTSP

for Partition $i = 1 \leftarrow n$ **do**

  procedure 4approx

  Obtain partition P_i from steps $1, 2$ and 3 of section 4.1

  if vehicle i should visit the vertices in the set S_i **then**

    $MST \leftarrow$ Minimum spanning tree $(P_i \cup S_i \cup d_i)$ {MST over P_i, S_i and corresponding depot}

  else

    Obtain partition R_i from steps $4, 5, 6$ and 7 of section 4.1

    $MST \leftarrow$ Minimum spanning tree $(P_i \cup R_i \cup d_i)$ {MST over P_i, R_i and corresponding depot}

  end if

  $euleriangraph \leftarrow$ DoubleSpanningTree(MST)

  $walk \leftarrow$ eulerwalk($d_i, euleriangraph$) {Refer to **Algorithm 2**}

  convert $walk$ to $tour$ by short-cutting

  end procedure

end for

Fig. 4.3. 4 - Approximation algorithm for MDMHTSP

In this section we give the proof of algorithm 3 in Fig. 4.3. Consider a set of optimal tours for the problem given. Short cut all the targets that belong to the set S_i $(or R_i)$ such that the vehicle visits only targets that belong to the set T. Let this cost be C_{common}. We assume that the costs satisfy the triangle equality. Therefore, the cost of the tour obtained by short cutting the targets that belong to S_i $(or R_i)$ namely

C_{common} is not going to increase and is at most equal to the optimal cost. In the first step of algorithm 3 in Fig. 4.3 we use an approximation factor of 2 to obtain a feasible solution and the cost of this solution is C_{feas_common}. Hence we can state that

$$C_{common} \leq C_{optimal} \tag{4.1}$$

$$C_{feas\_common} \leq 2C_{common} \tag{4.2}$$

From 4.1 and 4.2 it follows that

$$C_{feas\_common} \leq 2C_{optimal} \tag{4.3}$$

A similar argument can be made for the specific targets, in which we shortcut all the common targets and visit only the targets that belong to $S_i(or R_i)$. Let the cost of the feasible tour obtained from this be $C_{feas\_specific}$. Therefore we get,

$$C_{feas\_specific} \leq 2C_{optimal} \tag{4.4}$$

Now, the cost of the solution $C_{4-Approx}$ is given by

$$C_{4-Approx} = C_{feas\_specific} + C_{feas\_common} \tag{4.5}$$

From equation 4.3, 4.4 and 4.5 it follows that

$$C_{4-Approx} \leq 4C_{optimal} \tag{4.6}$$

4.3 3.5 - Approximation algorithm for MDMHTSP

Algorithm 4 3.5 - Approximation Algorithm for MDMHTSP

for Partition $i = 1 \leftarrow n$ **do**

  procedure 3.5approx

  Obtain partition P_i from steps $1, 2$ and 3 of section 4.1

  $MST \leftarrow$ Minimum spanning tree $(P_i \cup d_i)$ {MST over P_i and corresponding depot}

  if vehicle i should visit the vertices in the set S_i **then**

    $MatchedGraph \leftarrow$ Christofide(S_i) {Refer to **Algorithm 1**}

  else

    Obtain partition R_i from steps $4, 5, 6$ and 7 of section 4.1

    $MatchedGraph \leftarrow$ Christofide(R_i) {Refer to **Algorithm 1**}

  end if

  $euleriangraph \leftarrow$ Add($MST, MatchedGraph$)

  $walk \leftarrow$ eulerwalk($d_i, euleriangraph$) {Refer to **Algorithm 2**}

  convert *walk* to *tour* by short-cutting

  end procedure

end for

Fig. 4.4. 3.5 - Approximation algorithm for MDMHTSP

The proof of the 3.5 approximation algorithm is very similar to the proof in section 4.2. We can state that equation 4.3 is true for algorithm 4 in Fig. 4.4. Short cut the common targets from the optimal tour and we end up with a tour whose cost is at most equal to the optimal tour since short cutting will not increase the cost. However, we produce a feasible solution to the problem where common targets are short cut by using the Christofides algorithm. This gives an approximation factor of $\frac{3}{2}$. Hence we have,

$$C_{feas\_specific} \leq \frac{3}{2} C_{optimal} \tag{4.7}$$

Now, the cost of the solution $C_{4-Approx}$ is given by

$$C_{3.5-Approx} = C_{feas\_specific} + C_{feas\_common} \tag{4.8}$$

From equation 4.3, 4.7 and 4.8 it follows that

$$C_{4-Approx} \leq \frac{7}{2} C_{optimal} \tag{4.9}$$

4.4 Heuristic for MDMHTSP

In the final section of this chapter we will present a heuristic to solve the MDMHTSP. We first partition the given targets and allocate them to their respective vehicles by following the steps listed in section 4.1 in this chapter. After we have found all the partitions we use the Lin - Kernighan Heuristic (LKH) [29, 30] to solve individual TSPs for every vehicle.

The LKH uses a popular $k-opt$ heuristic method to solve the TSP. The heuristic starts with a feasible solution to the TSP. Then it performs k changes between the given graph and the feasible tour. Further it checks if the resulting solution is a cheaper tour. If the tour is cheaper then this will be used as the feasible solution for the next iteration. The process goes on until no further improvement can be made. The initial feasible tour that is picked and the termination criteria determine the speed and effectiveness of LKH.

CHAPTER V

COMPUTATIONAL STUDY AND RESULTS

5.1 Implementation

The two approximation algorithms for the MDMHTSP was implemented in MATLAB®. To generate the Minimum Spanning trees in MATLAB® the bioinformatics toolbox was used.

The minimum cost matching used in the second part of the of algorithm 4.4 was implemented using *BlossomV* [31]. *BlossomV* is an implementation of Edmond's algorithm [32]. This was run in a unix environment through MATLAB®.

For the heuristic, we partitioned all the targets and then used the LKH to find the solution. LKH gives very good high quality solutions for the TSP in a short time. This was developed by Helsgaun and the executable for this is available on line at [33]. The input to this could either be a cost matrix or the co-ordinates of the targets and vehicles The co-ordinates of targets and vehicles were given as the input. It is important to note that the LKH does not solve multiple vehicle problems. It solves only individual TSPs.

The algorithms were applied to test cases generated in an area of 25000 sq. units. The number of targets on the test cases varied from 10 to 50 with increments of 5 and the number of vehicles were $2, 3$ and 4. The vehicle target constraints are enforced based on the number that is equal to $\lceil \frac{m}{4n} \rceil$. The number $\lceil \frac{m}{4n} \rceil$ denotes how many specific targets should be (or should not be) visited by one vehicle.

The solutions for the integer program and a relaxed linear program were found by implementing the model of the MDMHTSP in CPLEX from IBM's ILOG Concert technology. We compare the solutions of our algorithm to the solution of the relaxed integer program where we relax the integral constraints as mentioned in section (3.2). The time taken by each of these algorithms and models were also recorded and compared with both the relaxed linear program and the integer program.

5.2 Results

The tests were implemented on an Intel® Xeon® X5450 3.00Ghz/16GB machine. The quality of the algorithms were compared to a lower bound and was averaged over all the instances. The average computation times of each of these algorithms and the lower bounds were plotted as a function of the number of targets. The quality of the algorithms is given by equation 5.1

$$Quality_i = \frac{Cost_i(alg) - Cost_i(LB)}{Cost_i(LB)} X100 \tag{5.1}$$

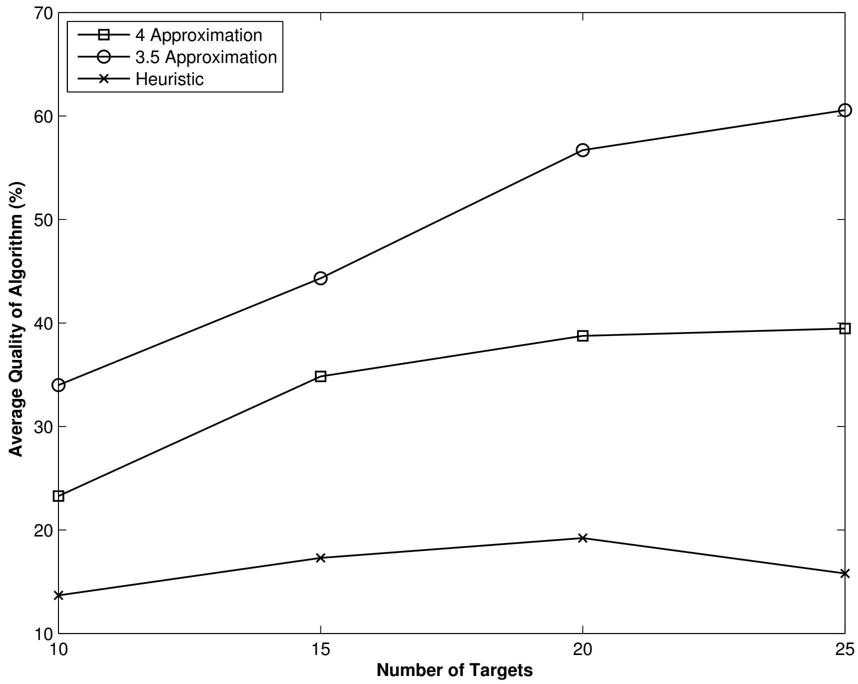
where,

$Cost_i(alg)$ = Cost of the solution obtained by algorithm alg for the instance i

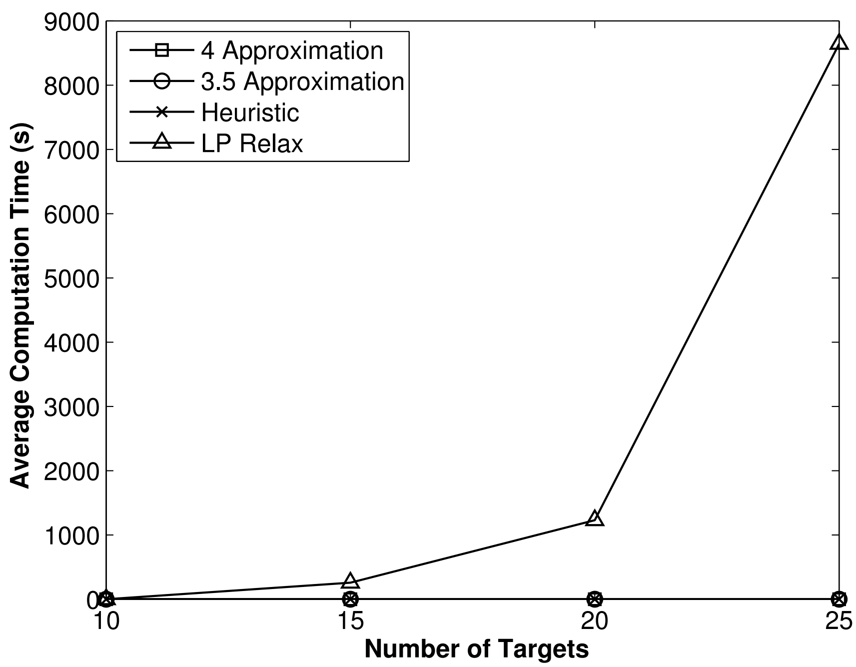
$Cost_i(LB)$ = Cost of the lower bound for the instance i

As we see in Fig. 5.1 the performance for the 2 vehicle case has been shown. From Figs. 5.1(a) we can see that the heuristic has the best performance in terms of the quality and varies from 14 - 20%. It is also observed from Fig 5.1(b) that in terms of time, the approximation algorithms and the heuristics perform very well as compared to the integer program.

We can also notice that the time taken to find the optimal solution increases significantly with increasing number of targets from Fig. 5.1(b).

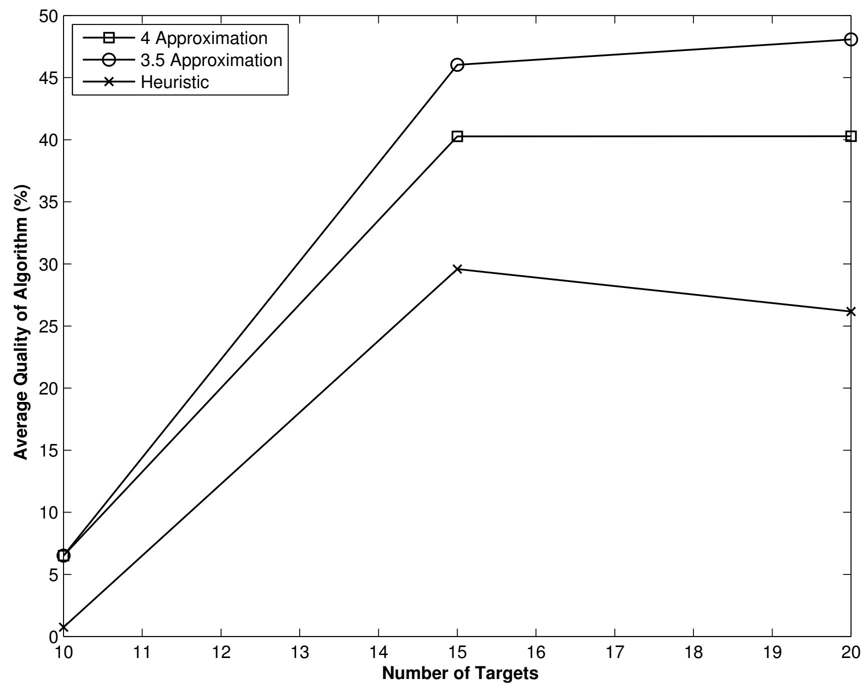


(a) Average Quality

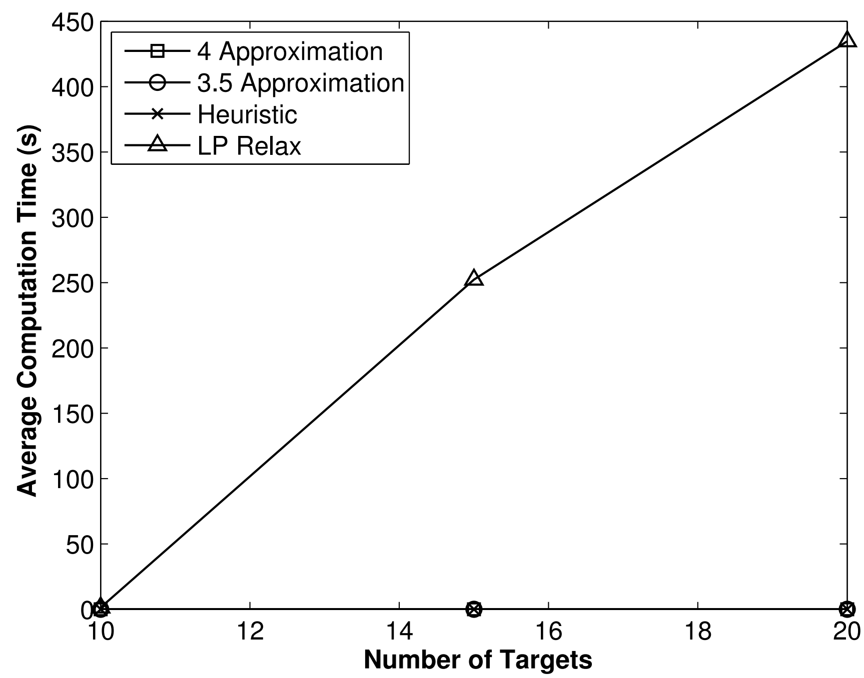


(b) Average Time

Fig. 5.1. Performance for 2 Vehicles (with no LP relaxation)



(a) Average Quality



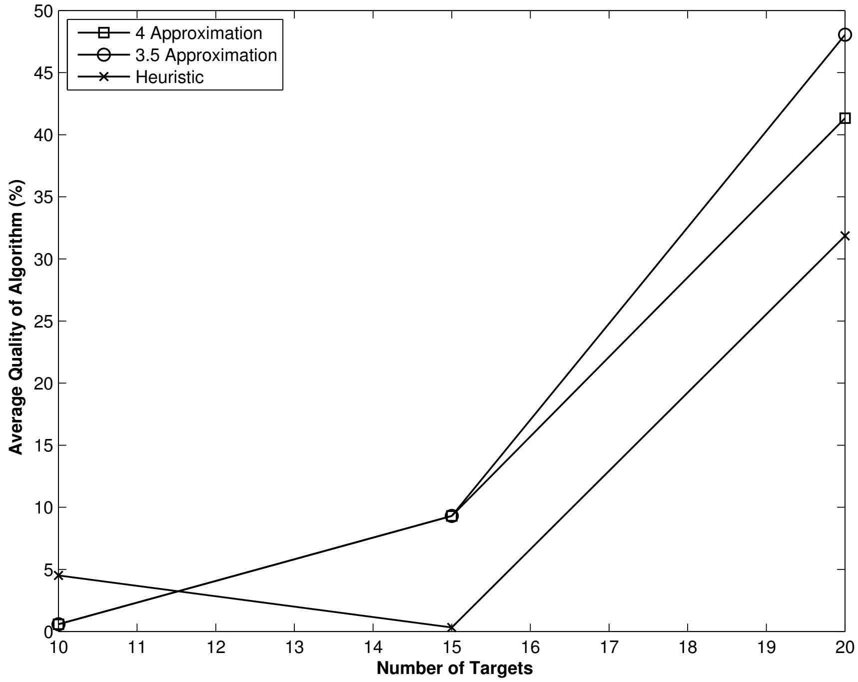
(b) Average Time

Fig. 5.2. Performance for 3 Vehicles (with no LP relaxation)

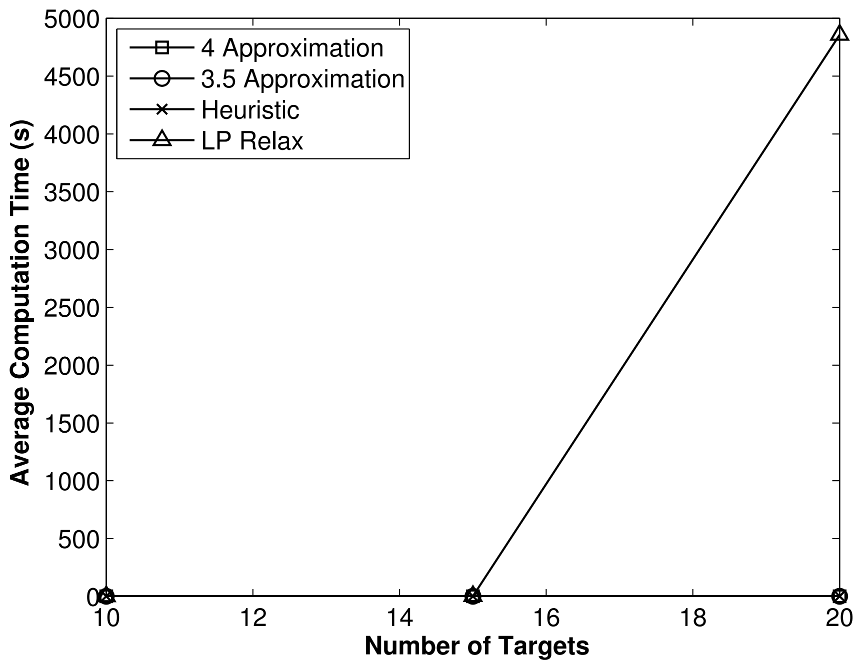
There is a similar trend that is observed in the 3 vehicle case. Figs. 5.2(a) and 5.2(b) clearly show that the two approximation algorithms produce similar solutions that are within 10% for 10 targets. The heuristic performs better than the two approximation algorithms in this case also. In terms of the time performance we have a similar trend as observed in the case of 2 vehicles.

In Fig. 5.3 it has to be noted that the 2 approximation algorithms behave in a very similar manner for 10 and 15 targets. They produce solutions that have a quality of around 10%. The performance of the heuristic increases from 10 to 15 targets and the quality reduces after that. This could be attributed to the vehicle target constraints which don't exist for both 10 and 15 targets for a 4 vehicle MDMHTSP. The plots for the time performance of the 4 vehicle case in Fig. 5.3(b) is very similar to the previous two cases where the integer program took the most time.

In the second part of the simulation we change the lower bound to the cost from the LP relaxation program. We obtain this cost by removing the integral constraints and solving the multi commodity flow problem to get a tighter lower bound as compared to the integer program.

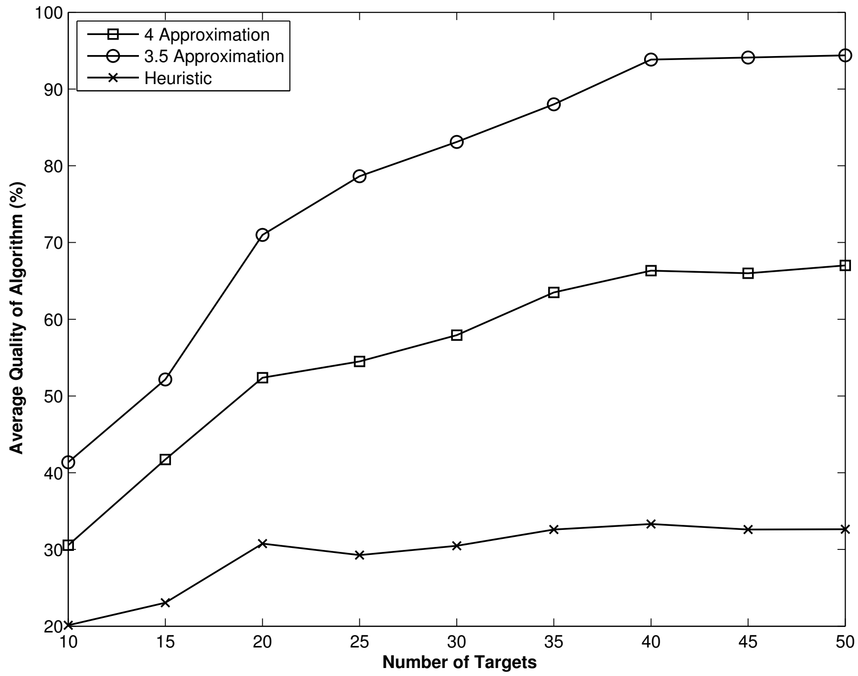


(a) Average Quality

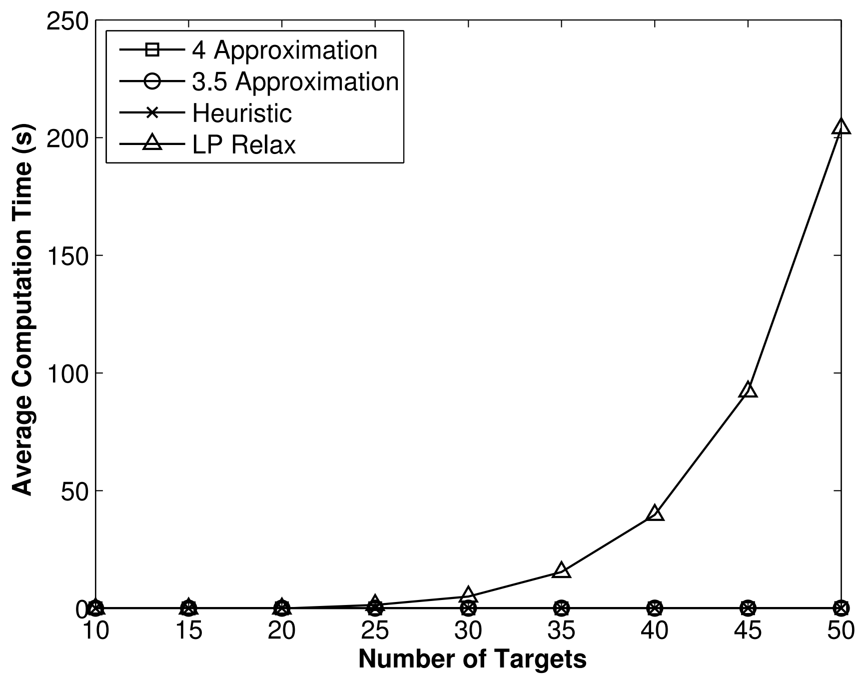


(b) Average Time

Fig. 5.3. Performance for 4 Vehicles (with no LP relaxation)



(a) Average Quality



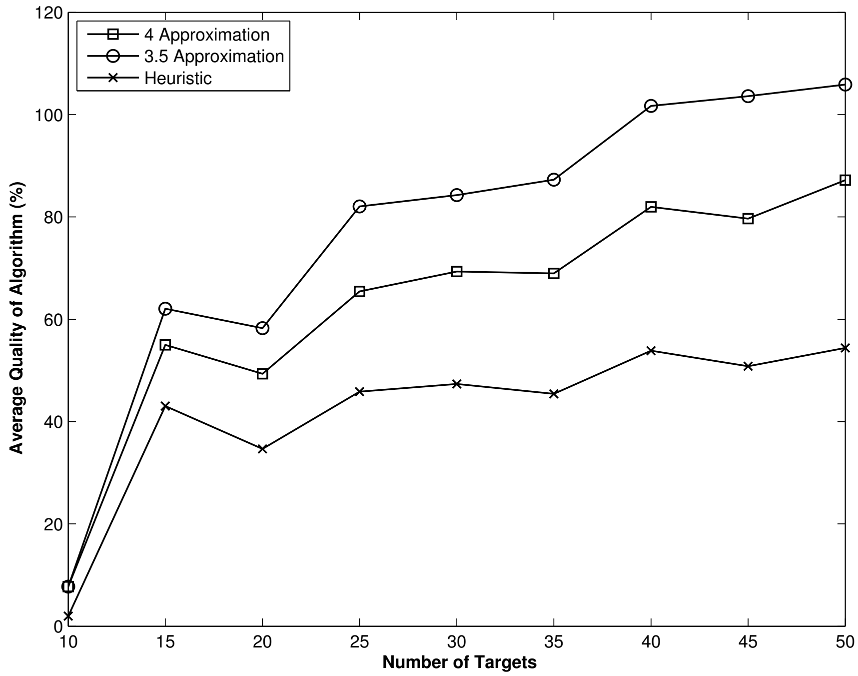
(b) Average Time

Fig. 5.4. Performance for 2 Vehicles (with LP relaxation)

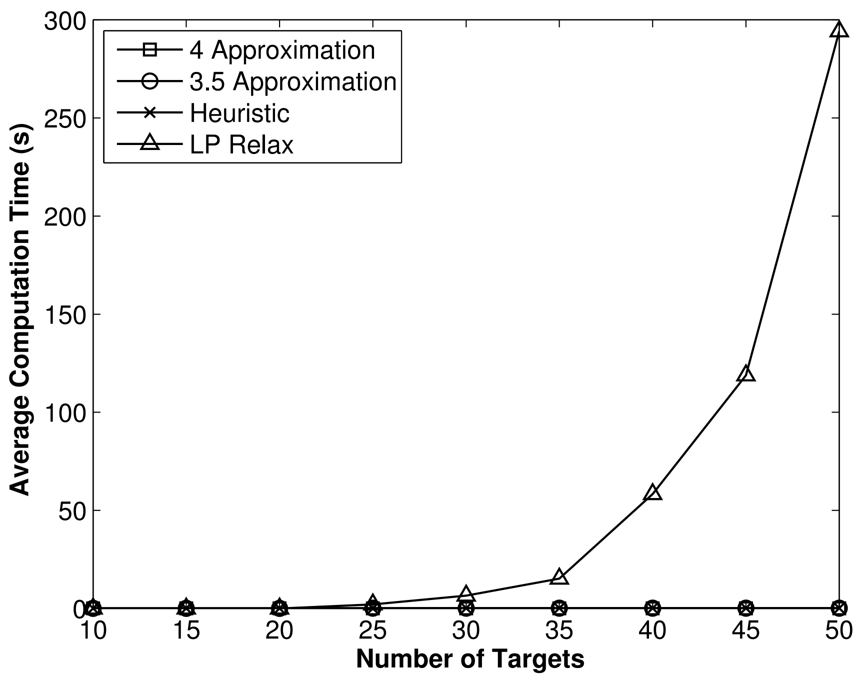
The qualitative performance and the average computation times of the algorithms and the heuristic for 2 vehicles in comparison to the relaxed linear program was found to be in accordance to Figs. 5.4(a) and 5.4(b) respectively. The performance of the 3 and 4 vehicles follows the pattern shown in Figs. 5.5 and 5.6. It is worth noting that the computation time for the relaxed LP is much lesser as compared to the integer program.

5.3 Evaluations

The LP relaxations provide a tighter lower bound for our problem. The curves follow a similar pattern for both the IP as well as the LP relaxation as the lower bound. This means that the LP formulation is close to the IP formulation. We can see that LP relaxation consumes very less time as compared to the IP counterpart. The time required to solve the LP formulations, increases exponentially with the increase in number of targets. Thus, there is a trade-off between the desired quality of the solution and the computation time available.

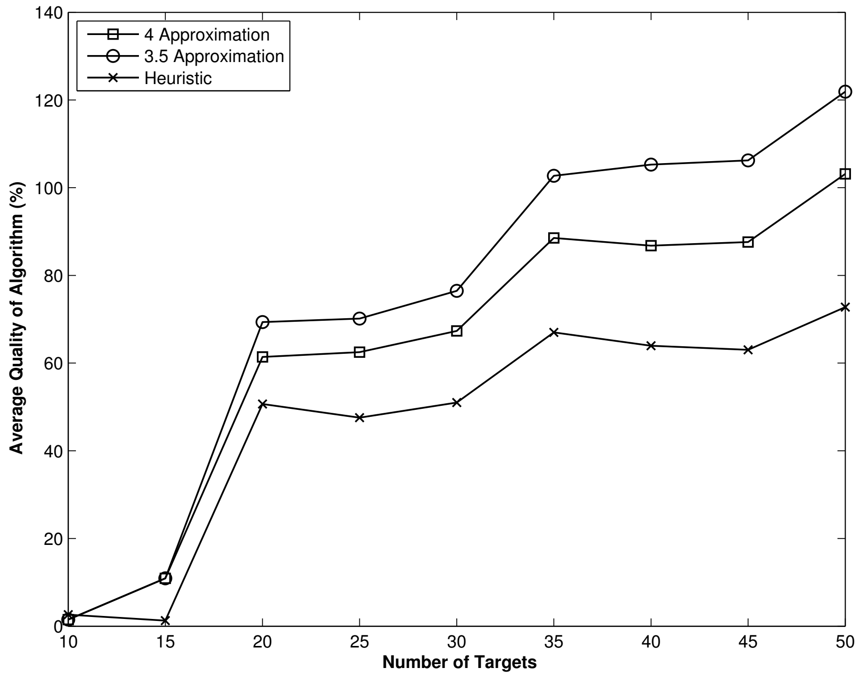


(a) Average Quality

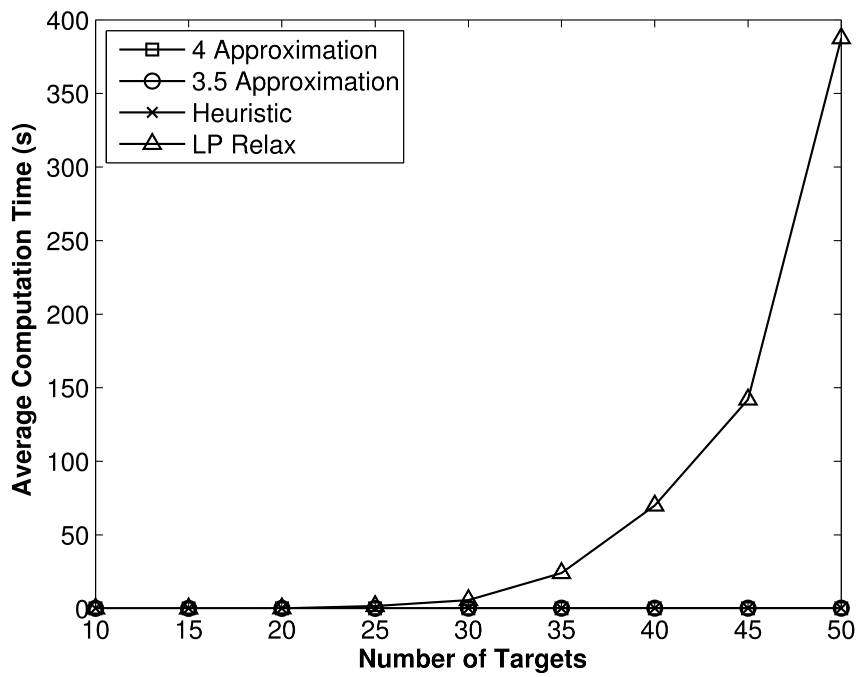


(b) Average Time

Fig. 5.5. Performance for 3 Vehicles (with LP relaxation)



(a) Average Quality



(b) Average Time

Fig. 5.6. Performance for 4 Vehicles (with LP relaxation)

CHAPTER VI

FUTURE WORK AND CONCLUSIONS

In this thesis we presented a detailed computation study of two approximation algorithms and a heuristic to solve the MDMHTSP. The approximation algorithms developed found solutions of better quality than the worst case guarantees. The heuristic developed performed very well and mostly found solutions of better quality as compared to the solutions found by the approximation algorithms. It can also be concluded that the quality of the solutions obtained by the algorithms and the heuristic reduce as the number of targets increase. The solutions produced by the integer program is very time consuming. Hence a trade off between the desired solution quality and the computation time needs to be found.

This problem can be generalized more by having different costs for different vehicles between targets. This will add another constraint to be satisfied while trying to solve the MDMHTSP. Developing approximation algorithms for such a case could be a future consideration as an extension of this problem. Adding motion constraints and changing the objective to minimizing the maximum distance could also be considered for future work.

REFERENCES

[1] J. Bellingham, M. Tillerson, A. Richards, and J. P. How, "Multi-task allocation and path planning for cooperating UAVs," in *Conference on Coordination, Control and Optimization, Gainesville, FL*, 2001.

[2] R. Kumar and Z. Luo, "Optimizing the operation sequence of a chip placement machine using tsp model," *IEEE Transactions on Electronics Packaging Manufacturing,*, vol. 26, no. 1, pp. 14 – 21, 2003.

[3] M. R. Bowers, W. M. Ng, K. Groom, and G. Zhang, "Cluster analysis to minimize sequence dependent changeover times," *Mathematical and Computer Modeling*, vol. 21, no. 11, pp. 89 – 95, 1995.

[4] G. Gutin and A. Punnen, *The Traveling Salesman Problem and Its Variations*, Boston: Kluwer Academic Publishers, 2002.

[5] P. Oberlin, S. Rathinam, and S. Darbha, "A transformation for a heterogeneous, multiple depot, multiple traveling salesman problem," in *American Control Conference, St. Louis, MO*, 2009, pp. 1292 –1297.

[6] G. Laporte, "The traveling salesman problem: An overview of exact and approximate algorithms," *European Journal of Operational Research*, vol. 59, no. 2, pp. 231–247, June 1992.

[7] G. J. Woeginger, "Combinatorial optimization - eureka, you shrink!," chapter Exact algorithms for NP-hard problems: a survey, pp. 185–207. New York: Springer-Verlag, 2003.

[8] S. Lin and B. W. Kernighan, “An effective heuristic algorithm for the traveling-salesman problem,” *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.

[9] A. Dumitrescu and J. S. B. Mitchell, “Approximation algorithms for TSP with neighborhoods in the plane,” *Journal of Algorithms*, vol. 48, no. 1, pp. 135 – 159, 2003.

[10] D. Du and P. M. Pardalos, *Handbook of Combinatorial Optimization*, vol. 1 - 3, New York: Springer Verlag, 1999.

[11] R. R. Mettu, “Approximation Algorithms for NP-Hard Clustering Problems,” Dissertation, University of Texas, Austin, 2002.

[12] V. V. Vazirani, *Approximation Algorithms*, New York: Springer-Verlag, 2001.

[13] N. Christofides, *Worst-case analysis of a new heuristic for the traveling salesman problem*, Tech. Rep., Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA,, 1976.

[14] Y. Gurevich and S. Shelah, “Expected computation time for Hamiltonian path problem,” *SIAM J. Comput.*, vol. 16, no. 3, pp. 486–502, June 1987.

[15] J. A. Hoogeveen, “Analysis of Christofides’ heuristic: some paths are more difficult than cycles,” *Operations Research Letters*, vol. 10, no. 5, pp. 291 – 295, 1991.

[16] N. Guttmann-Beck, R. Hassin, S. Khuller, and B. Raghavachari, “Approximation algorithms with bounded performance guarantees for the clustered traveling salesman problem,” *Algorithmica*, vol. 28, no. 4, pp. 422–437, 2000.

[17] C. Chekuri and M. Pál, "An $o(log\ n)$ approximation ratio for the asymmetric traveling salesman path problem," *Theory of Computing*, vol. 3, no. 1, pp. 197–209, 2007.

[18] S. Rathinam W. Malik and S. Darbha, "An approximation algorithm for a symmetric generalized multiple depot, multiple travelling salesman problem," *Operations Research Letters*, vol. 35, no. 6, pp. 747 – 753, 2007.

[19] S. Rathinam, R. Sengupta, and S. Darbha, "A resource allocation algorithm for multi-vehicle systems with non holonomic constraints," *Institute of Transportation Studies, University of California at Berkeley*, vol. 4, pp. 2007 – 2012, 2005.

[20] S. Rathinam S. Yadlapalli and S. Darbha, "3-approximation algorithm for a two depot, heterogeneous traveling salesman problem," *Optimization Letters*, vol. 4, no. 1, pp. 1–12, 2010.

[21] Z. Xu and B. Rodrigues, "A 3/2-approximation algorithm for multiple depot multiple traveling salesman problem," in *Algorithm Theory - SWAT 2010*, Haim Kaplan, Ed., vol. 6139 of *Lecture Notes in Computer Science*, pp. 127–138. Springer, Berlin, 2010.

[22] S. Rathinam and R. Sengupta, "Lower and upper bounds for a multiple depot UAV routing problem," in *45th IEEE Conference on Decision and Control, San Diego, CA*, 2006, pp. 5287 –5292.

[23] S. Rathinam and R. Sengupta, "3/2-approximation algorithm for two variants of a 2-depot Hamiltonian path problem," *Operation Research Letters*, vol. 38, no. 1, pp. 63–68, 2010.

[24] S. Rathinam S. K. Yadlapalli and S. Darbha, "An approximation algorithm for a 2-depot, heterogeneous vehicle routing problem," in *Proceedings of the 2009 conference on American Control Conference*. pp. 1730–1735, IEEE Press, Piscataway, NJ, 2009.

[25] G. B. Dantzig, D. R. Fulkerson, and S. M. Johnson, "On a linear-programming, combinatorial approach to the traveling salesman problem," *Operations research*, vol. 7, no. 1, pp. 58–66, 1959.

[26] T. Stützle, A. Grün, S. Linke, and M. Rüttger, "A comparison of nature inspired heuristics on the traveling salesman problem," in *Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*, London, UK, 2000, pp. 661–670, Springer-Verlag.

[27] M. Middendorf M. Guntsch and H. Schmeck, "An ant colony optimization approach to dynamic TSP," in *Conference on Genetic and Evolutionary Computation, San Francisco, CA*, 2001, pp. 860–867.

[28] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computing*, vol. 1, no. 1, pp. 53 –66, 1997.

[29] K. Helsgaun, "An effective implementation of the Lin-Kernighan traveling salesman heuristic," *European Journal of Operational Research*, vol. 126, no. 1, pp. 106–130, 2000.

[30] K. Helsgaun, "General k-opt submoves for the Lin-Kernighan TSP heuristic," *Mathematical Programming Computation*, vol. 1, no. 2-3, pp. 119–163, 2009.

[31] V. Kolmogorov, "Blossom V: a new implementation of a minimum cost perfect

matching algorithm," *Mathematical Programming Computation*, vol. 1, no. 1, pp. 43–67, 2009.

[32] J. Edmonds, "Paths, trees, and flowers," *Canad. J. Math.*, vol. 17, no. 1, pp. 449 – 467, 1965.

[33] K. Helsgaun, *Lin-Kernighan Heuristic*, [Online]. Available: http://www.akira.ruc.dk/ keld/Research/LKH/, 2009.

VITA

Rahul Rangarajan was born in Madras, India. After graduating from D.A.V. Senior Secondary School, Chennai, India he attended the National Institute of Technology, Jalandhar from August 2004 to May 2008. He majored in mechanical engineering and received a Bachelor of Technology in June 2008. He attended Texas A&M University in College Station from August 2008 to May 2011. He graduated with a Master of Science in mechanical engineering. He may be reached at:

Office Address:
425, James J. Cain
3123, Texas A&M University,
College Station, TX 77843 - 3123

The typist for this thesis was Rahul Rangarajan.