

**APPROXIMATION ALGORITHMS AND HEURISTICS FOR A
2-DEPOT, HETEROGENOUS HAMILTONIAN PATH PROBLEM**

A Thesis

by

RIDDHI RAJEEV DOSHI

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

August 2010

Major Subject: Mechanical Engineering

**APPROXIMATION ALGORITHMS AND HEURISTICS FOR A
2-DEPOT, HETEROGENOUS HAMILTONIAN PATH PROBLEM**

A Thesis

by

RIDDHI RAJEEV DOSHI

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,	Darbha Swaroop
	Sergiy Butenko
Head of Department,	Dennis O' Neal

August 2010

Major Subject: Mechanical Engineering

ABSTRACT

Approximation Algorithms and Heuristics for a 2-Depot, Heterogenous Hamiltonian Path Problem. (August 2010)

Riddhi Rajeev Doshi, B.E., North Gujarat University, Patan, India

Chair of Advisory Committee: Dr. Sivakumar Rathinam

Various civil and military applications of UAVs, or ground robots, require a set of vehicles to monitor a group of targets. Routing problems naturally arise in this setting where the operators of the vehicles have to plan the paths suitably in order to optimize the use of resources available such as sensors, fuel etc. These vehicles may differ either in their structural (design and dynamics) or functional (sensing) capabilities. This thesis addresses an important routing problem involving two heterogeneous vehicles. As the addressed routing problem is NP-Hard, we develop an approximation algorithm and heuristics to solve the problem.

Our approach involves dividing the routing problem into two sub-problems: Partitioning and Sequencing. Partitioning the targets involves finding two distinct sets of targets, each corresponding to one of the vehicles. We then find a sequence in which these targets need to be visited in order to optimize the use of resources to the maximum possible extent. The sequencing problem can be solved either by Christofides algorithm or the Lin-Kernighan Heuristic (LKH). The problem of partitioning is tackled by solving a Linear Program (LP) obtained by relaxing some of the constraints of an Integer Programming (IP) model for the problem. We observe the performance of two LP models for the partitioning. The first LP model is obtained by relaxing only the integrality constraints whereas in the second model relaxes both integrality and degree constraints.

The algorithms were implemented in a C++ environment with the help of Con-

cert Technology for CPLEX, and Boost Graph Libraries. The performance of these algorithms was studied for 50 random instances of varying problem sizes. It was found that on an average, the algorithms based on the first LP model provided better (closer to the optimum) solutions as compared to those based on the second LP model. We also observed that for both the LP models, the average quality of solutions given by the heuristics were found to be better (within 5% of the optimum) than the average quality of solutions obtained from the approximation algorithm (between 30 - 60% of the optimum depending on the problem size).

To my parents,
my sister,
and
all my teachers

ACKNOWLEDGMENTS

I wish to thank my advisor Dr. Sivakumar Rathinam, without whom this work would not have been possible. His constant motivation and guidance have been critical in the materialization of this work. I will be forever grateful to him for being my mentor. I would also like to convey my gratitude to Dr. Darbha Swaroop and Dr. Sergiy Butenko for being a part of my committee.

I thank my parents Arati and Rajeev and my sister Nidhi for their constant love and support. I also wish to thank my dear friend Harsha for all the brainstorming sessions, constructive criticism and constant encouragement. Finally, I would like to acknowledge all the countless people who directly or indirectly supported my work by providing useful resources that were crucial for successful completion of this work.

TABLE OF CONTENTS

CHAPTER	Page
I INTRODUCTION	1
II RELATED WORK	4
2.1 Hamiltonian Path Problem and Traveling Salesman Problem . .	4
2.2 Single Vehicle Problems	5
2.3 Multi Vehicle Problems	6
III PROBLEM OVERVIEW	8
3.1 Formulation	8
3.2 Integer Programming Model	9
IV THE ALGORITHMS	13
4.1 Approximation Algorithm for the 2-HMSF	13
4.2 Approximation Algorithm for the 2DHHPP	14
4.3 Proof of 8 - Approximation	15
4.4 Heuristic	18
4.5 Alternative Approximation Algorithm and Heuristic	18
V COMPUTATIONAL STUDY	20
5.1 Details of Implementation	20
5.2 Results	21
5.3 Evaluation	27
VI CONCLUSION AND FUTURE WORK	29
REFERENCES	30
VITA	34

LIST OF FIGURES

FIGURE		Page
1	Comparison of the lower bounds obtained by the two LP relaxations(with functional heterogeneity in the vehicles)	22
2	Average Solution Quality obtained by the algorithms with the two LP relaxations(with functional heterogeneity in vehicles)	23
3	Average Solution time for the IP and the LP relaxations(with functional heterogeneity in vehicles)	24
4	Comparison of the lower bounds obtained by the two LP relaxations(with structural and functional heterogeneity in the vehicles)	25
5	Average Solution Quality obtained by the algorithms with the two LP relaxations(with structural and functional heterogeneity in vehicles)	26
6	Average Solution time for IP and the LP relaxations(with structural and functional heterogeneity in vehicles)	27

CHAPTER I

INTRODUCTION

Unmanned Aerial Vehicles (UAVs) and/or ground robots are routinely involved in military applications for border patrol, mine clearance, reconnaissance, maritime surveillance expeditions, etc. Their civil applications usually include remote sensing, traffic monitoring, search and rescue, scientific research, etc. The missions employing these vehicles usually operate with constraints on time and resource. It thus becomes very crucial, to find an optimal path of travel for the vehicles involved. In most cases, a heterogeneous collection of vehicles differing in either structure or function is employed for the completion of a mission. In this thesis, we mainly concern ourselves with routing heterogeneous vehicles for various applications. This problem is more realistic and more challenging than its homogeneous counterpart due to the inherent differences among the vehicles.

We begin by classifying the heterogeneity of these vehicles into two basic categories: *structural* heterogeneity and *functional* heterogeneity. Vehicles that are structurally heterogeneous mainly differ in the design and dynamics. Thus, they may differ based on their fuel consumption, the maximum speed at which they can travel, the maximum payload capacity, etc. The difference in minimum turning radius can be sometimes ignored by assuming that the targets are reasonably distant from each other. However, this assumption can not disregard other differences and the cost of traveling between two targets still depends greatly on the type of vehicle employed.

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

Thus, some structural heterogeneity is always present among the vehicles. Even the vehicles that are more or less identical structurally may occasionally be equipped with different sensors and hence additional vehicle-target constraints must be met by the vehicles. The targets may then be partitioned into disjoint subsets: targets to be visited by specific vehicles and targets that any of the vehicles can visit. While the cost of travel between two targets remains equal for all the vehicles, the vehicle-target assignments induce a functional heterogeneity.

In this thesis, we primarily concentrate on the following 2-Depot, Heterogeneous Hamiltonian Path Problem (2DHHPP):

Given a set of targets and two heterogeneous vehicles located at distinct depots, find a Hamiltonian path for each vehicle such that the sum of costs of paths traveled by both the vehicles is minimized and each target is visited by exactly one vehicle.

The 2DHHPP is a generalization of Hamiltonian Path Problem, which is known to be NP-Hard [1]. Hence, there are no constant factor approximation algorithms possible for a general case of this problem unless $P = NP$. However, if all the costs satisfy triangle inequality, an α -approximation algorithm [2] could possibly be used for finding solutions to such problems. An α - approximation algorithm is characterized by the following properties:

- The algorithm runs in polynomial time to find a solution to a given problem.
- It guarantees a solution that is at most α times the optimal or $\text{OPT}(\alpha)$ for instance of the problem.

There are also several heuristics available to solve these types of problems. Heuristics can find a feasible solution to the problem in polynomial time, however, there are no guarantees on the quality of the solution obtained. This thesis presents

some approximation algorithms and heuristics for the 2DHHPP. It also presents a computational study comparing the performance of these algorithms under various scenarios.

The rest of the thesis is organized as follows. In Chapter II, we review and discuss the prior work related to our problem. In Chapter III, we provide an overview of the problem formulation and describe the integer programming model of the 2DHHPP. In Chapter IV, we present the algorithms and heuristic along with a proof of approximation. In Chapter V, we perform a computational study explaining the simulation setup. We also present an evaluation of the performance of the algorithms based on the results obtained. In Chapter VI, we conclude the thesis by summarizing the work and discussing possible improvements that can be made.

CHAPTER II

RELATED WORK

In graph theory [3], a Hamiltonian path is defined as a path in a directed or an undirected graph that visits each vertex of the graph exactly once. A Hamiltonian Path Problem (HPP) is thus the problem to find a Hamiltonian path in a given graph. In this thesis, however, we use HPP to denote to a problem of finding a *minimum-cost* Hamiltonian path in the graph. The 2DHHPP is then a generalization of the HPP for two heterogeneous vehicles, each stationed at a distinct depot, subjected to additional constraints of assigning vehicles to targets and vice-versa.

2.1 Hamiltonian Path Problem and Traveling Salesman Problem

In a graph, when the end nodes of a Hamiltonian path are connected, a Hamiltonian cycle is formed. A Hamiltonian Cycle Problem (HCP) is then a problem of finding a tour (in a graph) that visits each vertex exactly once. The Traveling Salesman Problem (TSP) is equivalent to the problem of finding the *minimum-cost* Hamiltonian cycle in a graph and thus is a generalization of the HCP.

We observe that the problems of TSP, HCP and HPP belong to a same class of problems and hence are closely related. This family of problems along with its several variations [4] has been extensively studied in the past and several algorithms, heuristics and transformations have been developed to solve these problems. It is well known that approximation algorithms for this class of problems can exist if the costs satisfy the triangle inequality [2], (i.e., the cost of direct travel between any two targets is at most as expensive as cost of traveling through an intermediate target)

unless $P=NP$. Henceforth, unless otherwise mentioned, we assume that the costs corresponding to each vehicle satisfy the triangle inequality. The following sections will discuss some noted work done in the field of single and multi depot problems related to the TSP family.

2.2 Single Vehicle Problems

For a single vehicle TSP (STSP), there are two approximation algorithms that are commonly used: the 2-approximation algorithm that doubles the Minimum Spanning Tree (MST) to find a feasible tour and the algorithm given by Christofides [5], known to give a solution no worse than 1.5 times the optimal solution. The latter approximation factor is obtained by combining the MST with a weighted non-bipartite minimum cost perfect matching of its odd-degree nodes. The performance guarantee of the Christofides algorithm has not been narrowed down for over two decades now and finding a smaller approximation factor remains an open problem. However, for a specific case of TSP where the distances between the nodes are either one or two, a $\frac{7}{6}$ - approximation algorithm is presented in [6]. Also, for a Euclidean TSP, Arora presented a Polynomial Time Approximation Scheme (PTAS) which for a fixed $c > 1$ is known to give a $(1 + \frac{1}{c})$ - approximation to the optimum TSP tour in $O(n(\log n)^{O(c)})$ time [7].

For a single vehicle HPP (SHPP), Hoogeveen in [8] employs adaptations of Christofides' algorithm to prove a $\frac{3}{2}$ - approximation algorithm for cases where a starting node/depot for a path is specified and a $\frac{5}{3}$ - approximation algorithm for cases where both the starting point and the terminal point are specified. Chekuri and Pal [9] address the Asymmetric HPP (AHPP) and present a $O(\log n)$ - approximation algorithm for the AHPP.

Several linear/integer programming models like the ones described in [10, 11, 12] have also been successfully employed to solve the TSP, HPP and their variants. These combinatorial approaches are known to give close to optimal or optimal solutions however, the time taken to obtain a solution can be very large depending upon the number of targets and the resource constraints that the vehicle is subjected to. There are also heuristics like the Lin-Kernighan Heuristics (LKH) [13] and the ant-colony optimization [14, 15] which are relatively fast.

2.3 Multi Vehicle Problems

For a homogeneous Multi TSP (MTSP) and its variations, Kara and Bektas present some integer linear programming formulations in [16]. A branch-and-bound based method for large scale MTSP is described in [17]. A cutting plane algorithm for MTSP is presented in [18]. An overview of the commonly used formulations for the MTSP is described in [19].

For Multi Depot TSP (MDTSP), HPP (MDHPP) and their variants, 2-approximation algorithms are presented in [20, 21]. For a Euclidean multi depot vehicle routing problem, Cardon et al. have proposed a PTAS that ensures a $(1 + \epsilon)$ -approximation for $\epsilon > 0$ with running time $(qd/\epsilon)^{O(q^3d^2/\epsilon^2)} + O(n \log n)$, where q is the number on customers visited by each of the d depots [22]. Rathinam and Sengupta also present a $\frac{3}{2}$ - approximation algorithm in [23] for two variants of a 2-depot Hamiltonian Path Problem. An 8-approximation algorithm for the 2DHHPP with vehicular constraints was presented by Yadlapalli et al. in [24]. However, there are no computational results currently available to test its performance. In this thesis, we present slight variations of this algorithm along with LKH based heuristics. The approximation factor for these modified algorithms remains the same. The main focus of this thesis is

to present a computational study for the modified algorithms and heuristics in order to have a better insight on their actual performance.

CHAPTER III

PROBLEM OVERVIEW

One of the challenging aspects of solving a 2DHHPP problem is to find a partition that separates the targets into two sets such that the targets in each partition are visited by exactly one of the two vehicles. Once we have the partitions, the task that remains is that of finding a minimum cost path for each of the vehicles over their corresponding targets. In the following sections, we present the formulation of the problem and describe an integer programming approach to find an appropriate partition of the targets.

3.1 Formulation

Let us assume that all the depots and targets are the vertices of a graph and all the paths joining these targets are the edges joining the corresponding vertices. Let $D = \{d_1, d_2\}$ be the set of vertices corresponding to the two depots and $T = \{1, 2, \dots, N\}$ be the set of vertices denoting all the targets to be visited by the two vehicles. Then, $V_1 = T \cup \{d_1\}$ is the set of all the vertices corresponding to vehicle 1 and $V_2 = T \cup \{d_2\}$ is the set of all vertices corresponding to vehicle 2. The two depots are not connected by any edge. Each vehicle, starts from its corresponding depot and ends some final target. The vehicles are free to choose their final destinations and hence they are not prescribed.

Suppose, E_1 and E_2 are the sets containing the edges that join any two vertices in V_1 and V_2 respectively. Let c_{ij}^1 be the cost incurred by vehicle 1 when traveling from vertex i to vertex j and c_{ij}^2 be the corresponding cost for vehicle 2. Then,

$G_1 = (V_1, E_1, c^1)$ and $G_2 = (V_2, E_2, c^2)$ are the weighted graphs corresponding to vehicles 1 and 2 respectively. The overall graph that the vehicles then have to search together in order to find their respective shortest Hamiltonian paths is $G = (G_1 \cup G_2)$. We assume that the costs satisfy the triangle inequality, i.e., for every $i, j, k \in V_1$, $i \neq j \neq k$, $c_{ij}^1 + c_{jk}^1 \leq c_{ik}^1$. Similarly, for every $l, m, n \in V_2$, $l \neq m \neq n$, $c_{lm}^2 + c_{mn}^2 \leq c_{ln}^2$. We also assume a functional heterogeneity in the vehicles owing to which, vehicle 1 is required to visit a set of vertices $R_1 \subseteq T$ and vehicle 2 is required to visit $R_2 \subseteq T$.

Let the number of vertices visited by the i^{th} vehicle be n_i , then the sequence of vertices visited by the i^{th} vehicle can be given by $S_i = \{d_i, a_1^i, a_2^i, \dots, a_{n_i}^i\}$, where $a_1^i, \dots, a_{n_i}^i \in T$ and $R_i \subseteq S_i$. The cost of travel for the i^{th} vehicle is given by $Cost_i = c_{d_i a_1^i}^i + \sum_{j=2}^{n_i} c_{a_{j-1}^i, a_j^i}^i$. Our problem is to find the sequences S_1 and S_2 such that each vertex is visited exactly once by exactly one of the vehicles and the total cost of travel, $Cost_1 + Cost_2$, is minimized.

3.2 Integer Programming Model

The 2DHHPP can be formulated as a multi-commodity flow problem (MCFP). Let there be n distinct commodities corresponding to the n targets. It is required that at least one unit of each commodity be delivered to its corresponding target by either of the two vehicles. Any vehicle i which delivers the commodity to the targets, will do so using the edges that carry the commodities from the i^{th} depot. Thus, the 2DHHPP becomes the problem of delivering at least one unit of each commodity to its corresponding destination by either of the two vehicles such that the total cost of building the transportation network is a minimum, and the degree constraints and the vehicle-target assignments are met.

We now formulate this MCFP as an integer programming problem. Let f_{ij}^k and

p_{ij}^k be the flows of the k^{th} commodity originating from depot 1 and depot 2 respectively and flowing from the i^{th} node to the j^{th} node. Though both f_{ij}^k and p_{ij}^k are free to flow on a given $arc(i, j)$, the total amount of flow is restricted by the capacity of the arc.

Let x_{ij} be a binary variable that decides whether $arc(i, j)$ is traveled by vehicle 1 and let y_{ij} be a similar variable corresponding to vehicle 2. The following capacity constraints then arise:

$$0 \leq f_{ij}^k \leq x_{ij} \quad \forall i \in V_1, j, k \in T \quad (3.1a)$$

$$0 \leq p_{ij}^k \leq y_{ij} \quad \forall i \in V_2, j, k \in T \quad (3.1b)$$

Let ψ_k be a binary variable that denotes the demand of commodity k fulfilled by depot 1. The demand fulfilled by the second depot is η_k . Thus, we get the following flow constraints:

$$\sum_{j \in V_1} f_{ji}^k - \sum_{j \in T} f_{ij}^k = -\psi_k \quad \forall k \in T \text{ and } i = d_1 \quad (3.2a)$$

$$\sum_{j \in V_1} f_{ji}^k - \sum_{j \in T} f_{ij}^k = 0 \quad \forall i, k \in T \text{ and } i \neq k \quad (3.2b)$$

$$\sum_{j \in V_1} f_{ji}^k - \sum_{j \in T} f_{ij}^k = \psi_k \quad \forall i, k \in T \text{ and } i = k \quad (3.2c)$$

$$\sum_{j \in V_2} p_{ji}^k - \sum_{j \in T} p_{ij}^k = -\eta_k \quad \forall k \in T \text{ and } i = d_1 \quad (3.3a)$$

$$\sum_{j \in V_2} p_{ji}^k - \sum_{j \in T} p_{ij}^k = 0 \quad \forall i, k \in T \text{ and } i \neq k \quad (3.3b)$$

$$\sum_{j \in V_2} p_{ji}^k - \sum_{j \in T} p_{ij}^k = \eta_k \quad \forall i, k \in T \text{ and } i = k \quad (3.3c)$$

The constraints (3.2a) and (3.3a) state that the flow of the k^{th} commodity out of the depots is equal to the demand of the commodity fulfilled by each of the de-

pots. The constraints (3.2b) and (3.3b) ensure that no part of the k^{th} commodity is supplied to intermediary nodes by any of the vehicles. Finally, the demand of the k^{th} commodity is completely supplied to the k^{th} target due to the constraints (3.2c) and (3.3c).

Since, the capacity of the j^{th} target is ψ_j for the first vehicle and (η_j) for the second vehicle, we need to ensure that the total capacity of the edges entering the j^{th} target is equal to its capacity and the total capacity of edges leaving this target is also at most equal to the capacity of the target j . The following equations implement these constraints.

$$\sum_{i \in V_1} x_{ij} = \psi_j \quad \forall j \in T \quad (3.4a)$$

$$\sum_{i \in V_2} y_{ij} = \eta_j \quad \forall j \in T \quad (3.4b)$$

$$\sum_{i \in V_1} x_{ji} \leq \psi_j \quad \forall j \in T \quad (3.5a)$$

$$\sum_{i \in V_2} y_{ji} \leq \eta_j \quad \forall j \in T \quad (3.5b)$$

To make sure that only one vehicle leaves each of the depots, we have an additional degree constraint on each of the depots as follows

$$\sum_{j \in T} x_{ij} \leq 1 \quad i = d_1 \quad (3.6a)$$

$$\sum_{j \in T} y_{ij} \leq 1 \quad i = d_2 \quad (3.6b)$$

Since, both the vehicles are required to visit their assigned targets R_1 and R_2 ,

the following vehicle-target constraints arise:

$$\psi_k = 1 \quad \forall k \in R_1 \quad (3.7a)$$

$$\eta_k = 1 \quad \forall k \in R_2 \quad (3.7b)$$

Furthermore, since the total demand of commodity k is 1 unit, the following coupling constraint arises.

$$\psi_k + \eta_k \geq 1 \quad \forall k \in T \quad (3.8)$$

Finally, it is required that x_{ij}, y_{ij} be binary integers, i.e.,

$$x_{ij}, y_{ij} \in \{0, 1\} \quad \forall i, j \in V_1 \cup V_2 \quad (3.9)$$

The variables $f_{ij}^k, p_{ij}^k, \psi_k$ and η_k are non-negative real numbers.

$$f_{ij}^k, p_{ij}^k, \psi_k, \eta_k \in \mathbb{R}^* \quad (3.10)$$

Thus, the problem of finding the minimum cost path for each vehicle can be defined as

$$C_{2DHHP} = \min \sum (c_{ij}^1 x_{ij} + c_{ij}^2 y_{ij}) \quad (3.11)$$

subject to the capacity constraints (3.1), flow constraints (3.2, 3.3), degree constraints (3.4, 3.5), vehicle-target constraints (3.7), coupling constraints (3.8), integrality constraints(3.9) and non-negativity constraints (3.10).

This integer program is a binary integer program (BIP). By removing the degree constraints (3.4, 3.5, 3.6) we obtained a relaxed problem which calculates the Heterogeneous Minimum Spanning Forest(HMSF). The role of the HMSF algorithm in our algorithms will be emphasized in the next chapter.

CHAPTER IV

THE ALGORITHMS

In order to find a solution to the 2DHHPP, we first partition the targets into two disjoint sets, each corresponding to one of the vehicles. To obtain an approximation algorithm we then apply the Christofides heuristic to each of these partitions in order to find a hamiltonian path for each vehicle.

Before we present the approximation algorithm for the 2DHHPP, we present an approximation algorithm for a relaxed problem, i.e., 2-HMSF. The basic ideas presented for the 2-HMSF, later form the crux of the approximation algorithm for the 2DHHPP.

4.1 Approximation Algorithm for the 2-HMSF

1. Relax the integrality constraints (3.9) and the degree constraints (3.6, 3.4, 3.5) from the IP in section 3.2 to get a relaxation of the 2-HMSF problem. This relaxed Linear Program (LP_{HMSF}) can be solved in polynomial time [25]. Assume that the optimal solution of this relaxation is $(x_{ij}^*, y_{ij}^*, f_{jk}^{i*}, p_{jk}^{i*}, \psi^*, \eta^*)$.
2. Find the optimal fractional amounts of the commodities shipped to each of the targets from both the depots. Assign the targets to the depot that ships the maximum amount of their corresponding commodities. Break the ties arbitrarily. Thus, $U_1 = \{k : k \in T, \psi_k \geq \eta_k\}$ is the set of targets assigned to the first vehicle, and $U_2 = T \setminus U_1$ is the corresponding set for the second vehicle. Denote the partition for each vehicle as $P_i (= U_i \cup \{d_i\})$
3. Compute the minimum cost spanning tree (MST_i) spanning over P_i (rooted

at d_i) using the costs associated with the corresponding vehicle i .

Algorithm 1 Christofides Algorithm

for Partition $P_i : i \leftarrow \{1, 2\}$ **do**

procedure CHRISTOFIDES(P_i)

$MST \leftarrow \text{FindKruskalMST}(P_i)$ \triangleright Find a MST that spans across P_i

$M \leftarrow \emptyset$ \triangleright M is a set of odd degree nodes

for $each v \in G$ **do**

if degree is odd **then**

$M \leftarrow \{v\}$

end if

end for

$M^* = \text{BlossomVmatch}(M)$ \triangleright M^* is the result of minimum matching on M

$EulerGraph = \text{Add}(MST, M^*)$ \triangleright combine M^* and MST to form an Euler

Graph

$\text{ConstructEulerWalk}(EulerGraph)$ \triangleright See Algorithm 2

 Transform the walk into a Hamiltonian tour(path) by short-cutting the visited vertices.

end procedure

end for

4.2 Approximation Algorithm for the 2DHHPP

1. Partition the nodes into sets P_1 and P_2 using steps (1) and (2) in the 2-HMSF algorithm described in section 4.1. Denote the feasible 2-HMSF solution by $HMSF_{feasible}$
2. For each minimum spanning tree MST_i , find the nodes with odd-degree. The

Handshaking lemma [3] , states that there will be an even number of such nodes. Let the M_1 and M_2 be the sets of odd-degree nodes for the two MSTs.

3. Perform a minimum cost perfect matching on sets M_i and add the matching edges to the corresponding MST to obtain a connected Eulerian subgraph.
4. Find an Euler Walk in the subgraph using Fleury's algorithm [26] or the cycle finding algorithm (see algorithm 2). The Eulerian walk traverses each edge in the subgraph exactly once.
5. Shortcut the Eulerian walk to obtain a Hamiltonian Path for each vehicle. This assures that each target is visited exactly once.

Algorithm 2 Algorithm for Euler Walk

procedure CONSTRUCTEULERWALK($Graph$)

$walkPos \leftarrow 0$

EulerWalk($start$)

▷ The starting vertex

end procedure

procedure EULERWALK(u)

for each vertex v adjacent to u **do**

RemoveEdge(u, v)

EulerWalk(v)

end for

$Walk[walkPos + +] \leftarrow u$

end procedure

4.3 Proof of 8 - Approximation

We now prove an 8 - Approximation for our algorithm. For this, we first prove that the 2-HMSF algorithm is a 4-approximation algorithm.

Theorem 4.3.1 *The cost of feasible solution produced by the 2-HMSF is at the most four times the cost of the relaxed linear program. It is thus, less than $4C_{opt}(HMSF)$, i.e., the 2-HMSF algorithm has an approximation factor of four.*

At the end of the 2-HMSF algorithm, let the cost of MST for the two vehicles be denoted by $C_{feasible}(MST_1)$ and $C_{feasible}(MST_2)$. Let the optimal cost of the relaxed linear program for the 2-HMSF problem be $C_{opt}(HMSF)$. We see that the variables corresponding to both the vehicles are coupled only through constraint (3.8). Consider another LP relaxation where (3.8) is replaced by the following constraints:

$$\psi_k \geq 1 \quad \forall k \in U_1 \quad (4.1a)$$

$$\eta_k \geq 1 \quad \forall k \in U_2 \quad (4.1b)$$

This new LP formulation($HMSF_{new}$) now allows us to split the main problem into two sub-problems. The first sub-problem is a relaxation of the HPP for the first vehicle over P_1 and the second is relaxation for the second vehicle over P_2 . Thus, if we assume the optimal cost of the new LP formulation to be $C_{opt}(HMSF_{new})$, then we have

$$C_{opt}(HMSF_{new}) = C_{opt}(P_1) + C_{opt}(P_2) \quad (4.2)$$

where, $C_{opt}(P_1)$ and $C_{opt}(P_2)$ are the optimal costs of the LP relaxations of the HPPs corresponding to vehicle 1 and 2 respectively.

We can thus prove theorem 4.3.1 by proving the lemmas 4.3.2 and 4.3.3.

Lemma 4.3.2

$$C_{opt}(HMSF_{new}) \leq 2C_{opt}(HMSF)$$

Lemma 4.3.3

$$\begin{aligned}\frac{1}{2}C_{feasible}(MST_1) &\leq C_{opt}(P_1) \\ \frac{1}{2}C_{feasible}(MST_2) &\leq C_{opt}(P_2)\end{aligned}$$

A detailed proof of these lemmas and theorem 4.3.1, can be found in [24]. Thus, finally we can deduce from these two lemmas that

$$\begin{aligned}C_{opt}(P_1) + C_{opt}(P_2) &\leq 2C_{opt}(HMSF) \\ \frac{1}{2}C_{feasible}(HMSF) &\leq 2C_{opt}(HMSF) \\ C_{feasible}(HMSF) &\leq 4C_{opt}(HMSF)\end{aligned}$$

Now, to prove the 8-approximation, let $C(S_i)$ be the total cost of all edges on the path S_i . Since, the costs satisfy the triangle inequality, we may apply lemma 4.3.4 described below for our proof.

Lemma 4.3.4 *If the costs satisfy the triangle inequality, then*

$$C_{matching} \leq C_{feasible}(HMSF)$$

The proof of lemma 4.3.4 can be found in [27]. Thus, we have

$$\begin{aligned}C(S_1) + C(S_2) &\leq C_{feasible}(HMSF) + C_{matching} \\ &\leq 2C_{feasible}HMSF\end{aligned}$$

Therefore,

$$\begin{aligned}\sum_{i=1,2} C(S_i) &\leq 8C_{opt}(HMSF) \text{ (from theorem 4.3.1)} \\ &\leq 8C_{opt}(2DHPP)\end{aligned}$$

where, $C_{opt}(2DHPP)$ is the optimal cost of the 2DHPP.

4.4 Heuristic

The 2DHHPP can also be solved by applying the Lin-Kernighan Heuristic (LKH) instead of the Christofides algorithm to the partition obtained by the 2-HMSF. The Lin-Kernighan Heuristic(LKH), employs a *variable k-opt* algorithm to find the shortest tour/path in a given graph. It begins by randomly selecting a feasible tour on the graph. At each iteration, for increasing values of k , the heuristic checks whether an interchange of k -edges between the current tour and the rest of the graph will result in a shorter tour. These edges are selected such that a feasible tour maybe formed at any stage of the algorithm. This process is continued until no further improvement is possible or until all possible exchanges are exhausted.

The search strategy and the termination criterion for the LKH play a very crucial role in its performance. Also, the selection of a good starting tour may reduce the computation time for finding the final solution. Helsgaun in [28] devised a better implementation which restricted the search for the replacement edges within the set of edges that are ranked according to their closeness to the dual solution of the TSP obtained by the Held-Karp relaxation. This reduces the search set drastically. Also, the starting tour is calculated such that most of the edges belong to the optimal dual solution or are close to it. These improvements along with many others described in [28, 29] have drastically improved the performance of the LKH and enabled it to provide optimal solutions for many large instances of TSP.

4.5 Alternative Approximation Algorithm and Heuristic

For both the algorithms mentioned in the previous sections, the HMSF algorithm used for partitioning the targets into two disjoint sets, utilized an LP relaxation which had both degree constraints and integrality constraints relaxed. In an alternative

approach, we relax only the integrality constraints thus keeping the relaxation closer to the actual IP formulation.

The proof of 8-approximation factor for the approximation algorithms still remains the same. However, one can expect that the feasible solutions given by both the algorithms may be improved by using this alternative relaxation.

CHAPTER V

COMPUTATIONAL STUDY

5.1 Details of Implementation

The implementation of our algorithms has been done in *C++*. The CPLEX callable libraries available from IBM's ILOG Concert Technology have been embedded in the implementation for solving the relaxed linear and integer programs.

The Boost Graph Library(BGL) [30] was used to implement the Christofides algorithm. The minimum cost matching required for this algorithm is obtained through *Blossom V* [31], an implementation of Edmonds' *blossom* algorithm [32]. For details, see algorithm 1

For applying LKH to the partitions, we used the implementation developed by Helsgaun [28, 29], which is available online [33]. This implementation of the LKH is known to give very high quality solutions in a reasonably short time. In order to be able to use this implementation, the costs of traveling between any two nodes was rounded up to a positive integer value. Also, since this implementation doesn't solve an HPP. We used the implementation for the Asymmetric TSP assuming the costs of all the edges incoming to the depot are zero.

The algorithms were applied to a test area of 5000 by 5000 sq. units. Fifty random instances were generated for each problem size ranging from 15 to 50 nodes (in increments of 5).

Two test cases were designed based on the heterogeneity in vehicles. In the first case, we simulated a functional heterogeneity by assigning about first fifth of all the targets to the first depot and the last fifth of the targets to the second depot. For

the second test case, we multiplied the cost matrix of the first vehicle with a factor of 1.5. This simulated the differences in the costs of travel that arise due to structural heterogeneity. Thus, in the second case both structural and functional heterogeneity were introduced.

For both these cases, the Integer Programming solutions were obtained using the CPLEX solver. Two different relaxations for the Linear Program were tested. The first according to section 4.5 had only the integrality constraints relaxed. We denote this relaxation by LP_1 . The second as described in the HMSF algorithm(section 4.1), had both the integrality and the degree constraints relaxed. This is denoted by LP_2 .

The time taken by CPLEX to solve each problem instance was noted for all the methods. However, since the time taken by the Christofides algorithm and LKH heuristics was almost negligible, we compare only the time taken to solve the two LP relaxations with time taken to solve the integer program.

5.2 Results

All the tests were implemented on an Intel[®] Xeon[®] X5450 3.00GHz/16GB machine. For the first case with only functional heterogeneity introduced, the average quality of lower bound was found to be tighter in case of LP_2 . This is indicated in Figure 1.

Given an algorithm a and an instance I , the following equation was used to calculate the quality of the solutions produced by the algorithm on I.

$$Quality_I = \frac{Cost_I(a) - Cost_I(IP)}{Cost_I(IP)}.100\% \quad (5.1)$$

where,

$Cost_I(a)$ = Cost of the solution obtained by the algorithm a on the instance I

$Cost_I(IP)$ = Optimal cost of the 2DHHPP obtained by solving the Integer Program-

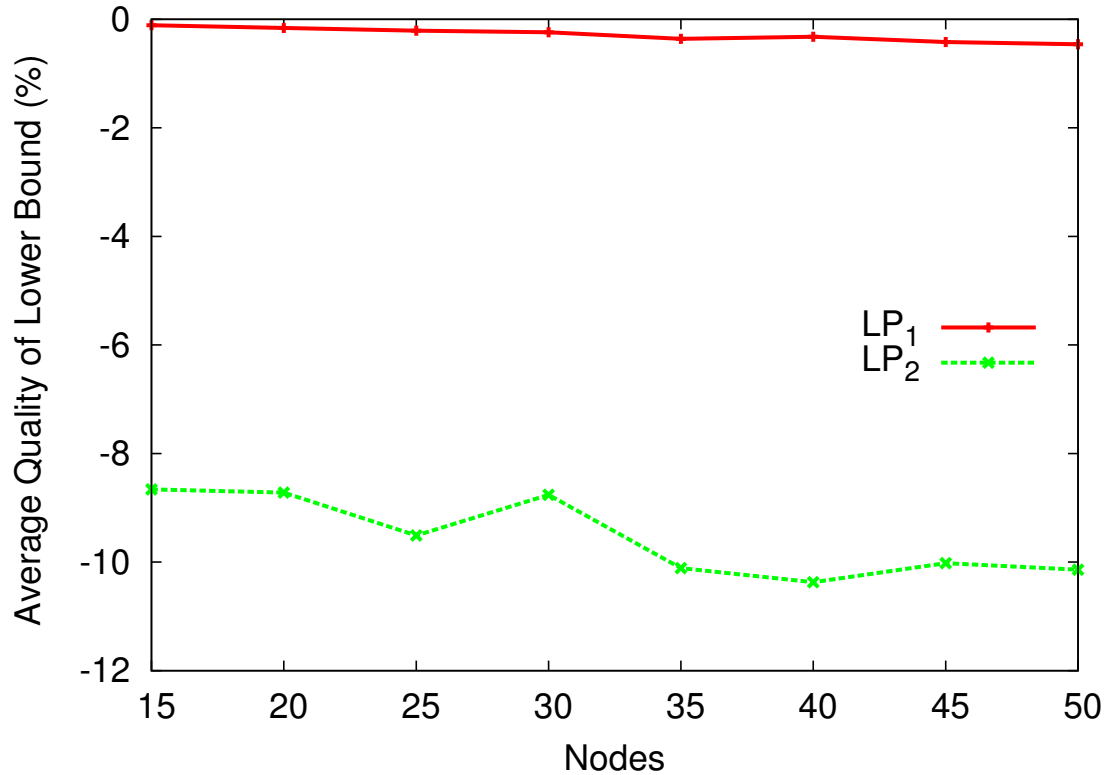
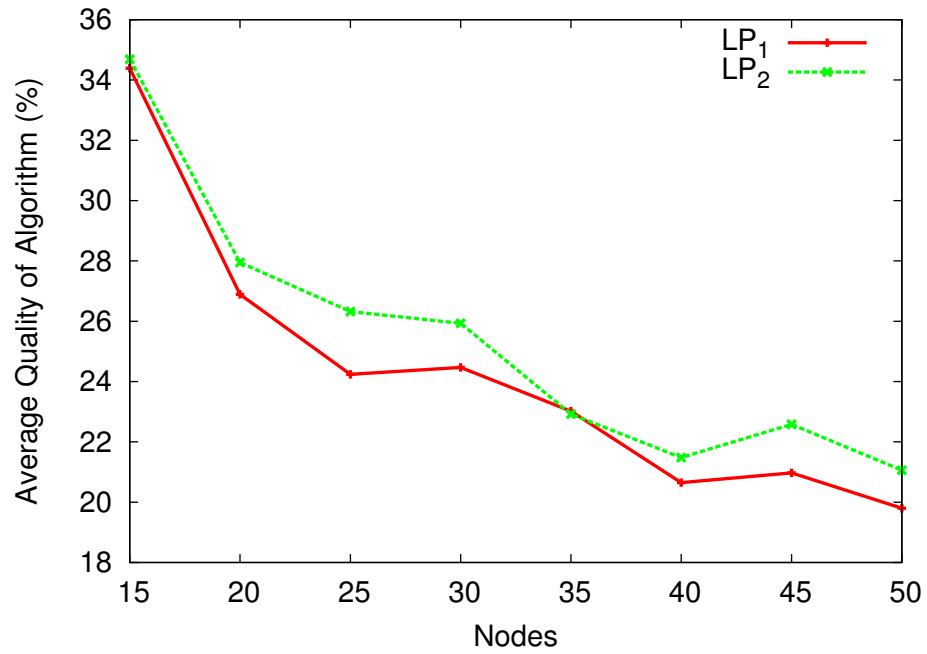


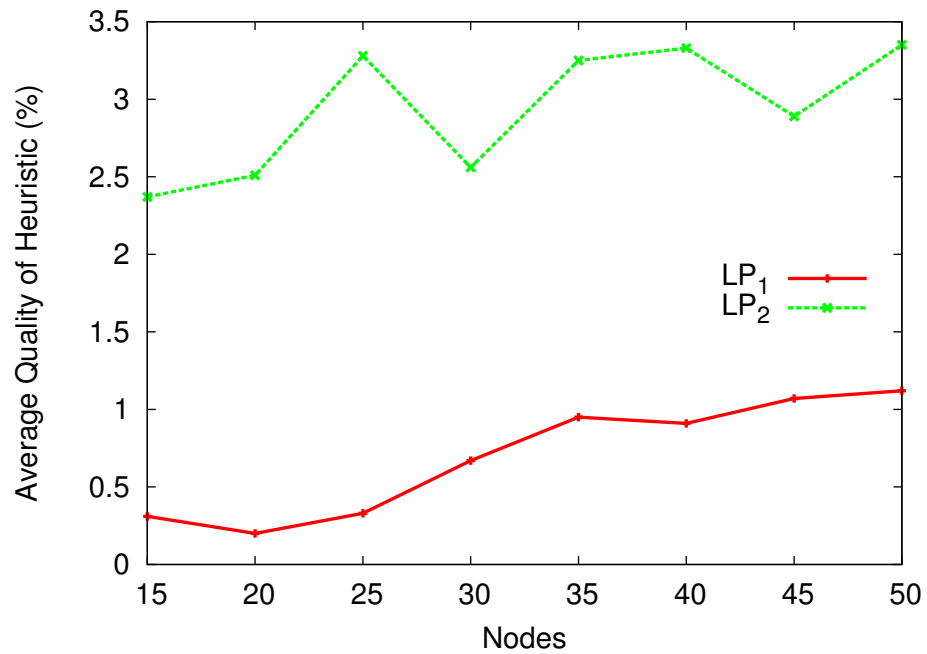
Fig. 1. Comparison of the lower bounds obtained by the two LP relaxations (with functional heterogeneity in the vehicles)

ming problem on the instance I .

When the qualities were compared for the approximation algorithm and the heuristic the results improved in the cases with only the integrality constraints relaxed. Figure 2 shows a clear comparison of the average quality of solution provided by each of the algorithms.



(a) Average Quality of the Approximation Algorithm



(b) Average Quality of the Heuristic

Fig. 2. Average Solution Quality obtained by the algorithms with the two LP relaxations (with functional heterogeneity in vehicles)

The time study for each of the LP relaxations and IP is presentend in Figure 3.

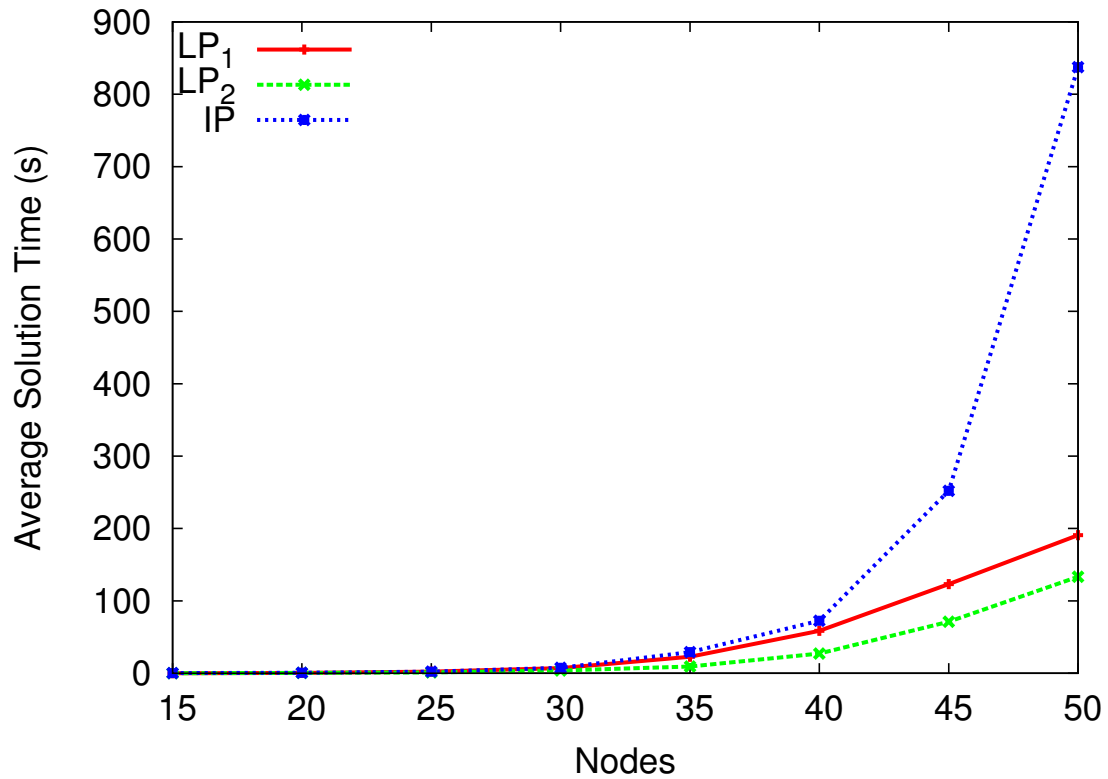


Fig. 3. Average Solution time for the IP and the LP relaxations(with functional heterogeneity in vehicles)

Similar results were also observed for the case with both structural and functional heterogeneity among the vehicles. The lower bound was again found to be better in case of LP_1 as can be seen from Figure 4.

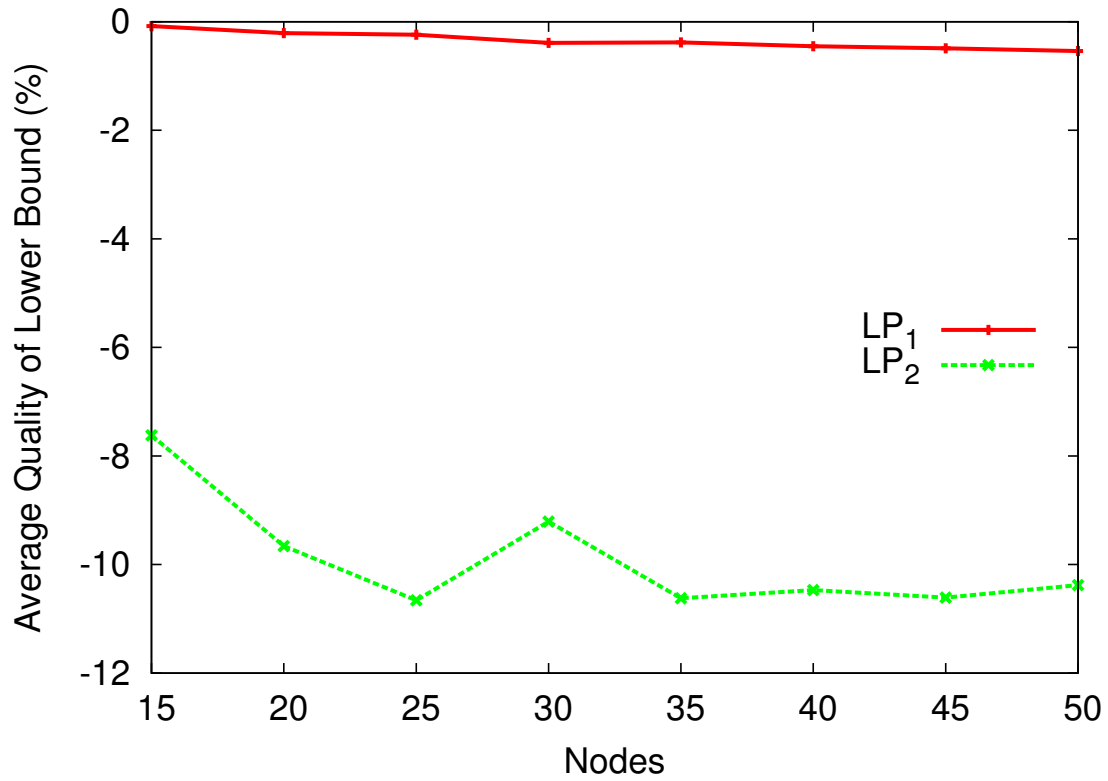
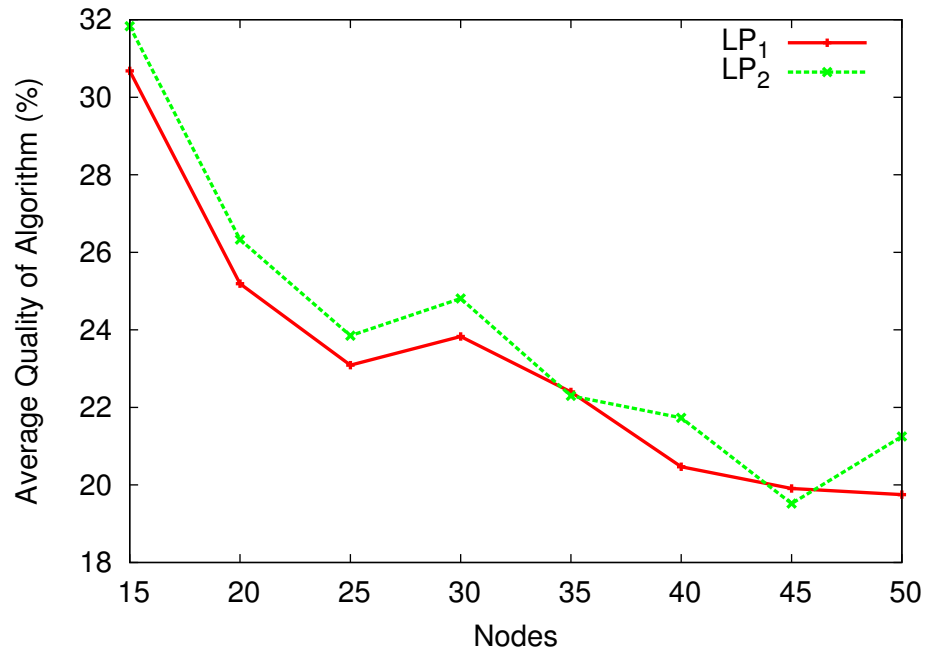
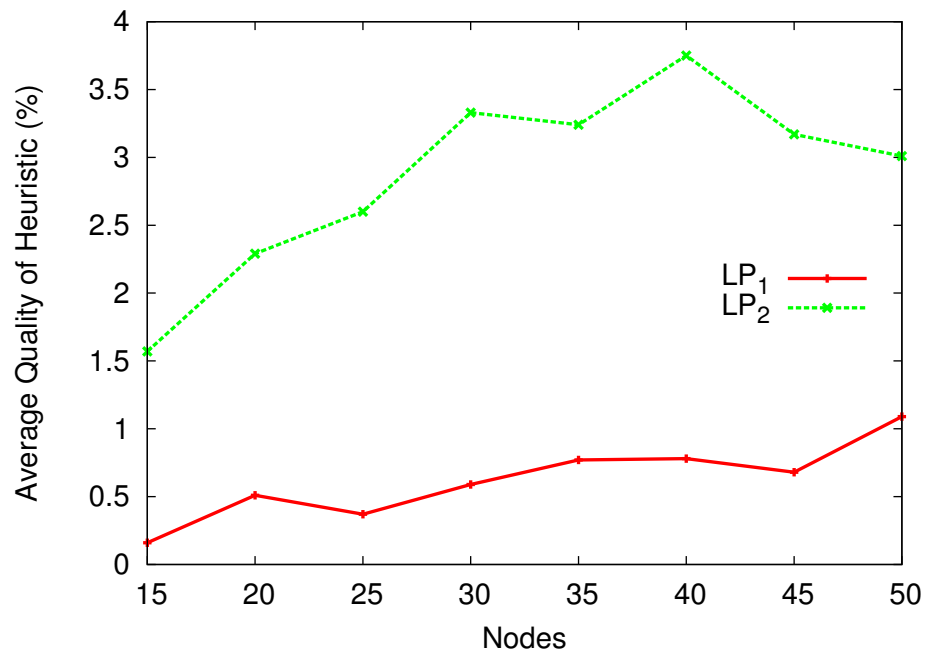


Fig. 4. Comparison of the lower bounds obtained by the two LP relaxations (with structural and functional heterogeneity in the vehicles)

The quality of the approximation algorithms and heuristics was found to be in accordance to Figure 5.



(a) Average Quality of Approximation Algorithm



(b) Average Quality of Heuristic

Fig. 5. Average Solution Quality obtained by the algorithms with the two LP relaxations (with structural and functional heterogeneity in vehicles)

Figure 6 shows the time comparison for the performance of the IP formulation and the LP relaxations in case when both structural and functional heterogeneity is present among the vehicles.

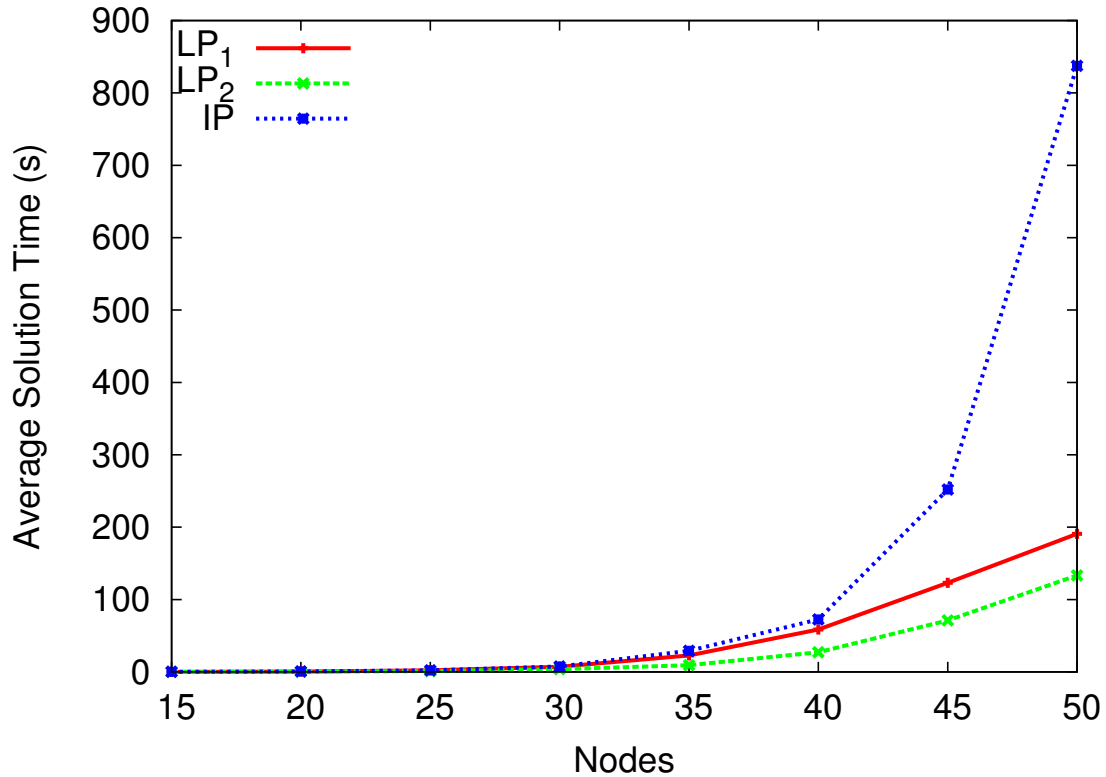


Fig. 6. Average Solution time for IP and the LP relaxations(with structural and functional heterogeneity in vehicles)

5.3 Evaluation

The LP relaxations provide a lower bound for our problem. We find that the relaxation with only the integrality constraints relaxed, provide a tighter lower bound on the solution as compared to the case where the degree constraints are also relaxed. Moreover, the solutions obtained by the heuristic also improves when only the degree

constraints are relaxed. The improvement can also be seen in the results obtained from the approximation algorithm even though the improvement is not drastic. This improvement in the results may be due to the fact that the formulation of LP with only the integrality constraints relaxed shares more similarities with the IP formulation.

However, the improvement comes at an expense of solution time. The solution time for the relaxation without the degree constraints is lower than the one with degree constraints. Since, the time required to solve the LP formulations, increases exponentially with the increase in number of nodes, the difference in the time taken by the two relaxations even though smaller for the observed instances, may increase to a much higher value with the increase in number of nodes. Thus, a trade-off between the quality of the solution and the time taken needs to be found.

CHAPTER VI

CONCLUSION AND FUTURE WORK

In this thesis we present a detailed computational study of the algorithms and the heuristic for the 2DHHPP. This study helps us get a better idea of the actual performance of these algorithms. So far, no computational studies like the one presented in this work, have been available for the 2DHHPP. The approximation algorithm can be easily extended to cases with more than two vehicles. In that case, the approximation factor will be $4m$, where m is the number of depots.

The performance of our algorithm may be improved by adopting a slightly modified version Christofides algorithm for finding a Hamiltonian Path [8]. However, this hypothesis has not been tested as yet. In future, we plan to test if this hypothesis is true. The case of 2DHHPP with both the end-points of the path are fixed is also quite interesting. This appears to be a more challenging case as compared to the current case with just the starting point fixed. Finding algorithms to solve this new case, is also left for future consideration.

REFERENCES

- [1] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W. H. Freeman & Co., 1990.
- [2] V. V. Vazirani, *Approximation Algorithms*. Berlin: Springer-Verlag, 2001.
- [3] D. B. West, *Introduction to Graph Theory*. Upper Saddle River, NJ: Prentice Hall, 1996.
- [4] G. Gutin and A. P. Punnen, *The Traveling Salesman Problem and Its Variations*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 2002.
- [5] N. Christofides, “Worst-case analysis of a new heuristic for the travelling salesman problem,” Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep., 1976.
- [6] C. H. Papadimitriou and M. Yannakakis, “The traveling salesman problem with distances one and two,” *Mathematics of Operations Research*, vol. 18, no. 1, pp. 1–11, 1993.
- [7] S. Arora, “Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems,” *J. ACM*, vol. 45, no. 5, pp. 753–782, 1998.
- [8] J. Hoogeveen, “Analysis of christofides heuristic: Some paths are more difficult than cycles.” *Operations Research Letters*, no. 10, pp. 291– 295, 1991.
- [9] C. Chekuri and M. Pål, “An $O(\log n)$ approximation ratio for the asymmetric traveling salesman path problem,” *Theory of Computing*, vol. 3, no. 1, pp. 197–209, 2007.

- [10] 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.
- [11] C. E. Miller, A. W. Tucker, and R. A. Zemlin, “Integer programming formulation of traveling salesman problems,” *J. ACM*, vol. 7, no. 4, pp. 326–329, 1960.
- [12] D. Applegate, R. Bixby, V. Chvatal, and W. Cook, “Implementing the dantzig-fulkerson-johnson algorithm for large traveling salesman problems,” *Mathematical Programming*, vol. 97, no. 1, pp. 91–153, July 2003.
- [13] S. Lin and B. Kernighan, “An effective heuristic algorithm for the traveling-salesman problem,” *Operations Research*, vol. 21, pp. 498–516, 1973.
- [14] M. Dorigo and L. Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem,” *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, April 1997.
- [15] M. Dorigo and T. Sttzle, *Ant Colony Optimization*. Boston, MA: MIT Press, 2004.
- [16] I. Kara and T. Bektas, “Integer linear programming formulations of multiple salesman problems and its variations,” *European Journal of Operational Research*, vol. 174, no. 3, pp. 1449–1458, 2006.
- [17] B. Gavish and K. Srikanth, “An optimal solution method for large-scale multiple traveling salesmen problems,” *Operations Research*, vol. 34, no. 5, pp. 698–717, 1986.
- [18] G. Laporte and Y. Nobert, “A cutting planes algorithm for the m-salesmen problem,” *The Journal of the Operational Research Society*, vol. 31, no. 11, pp.

1017–1023, 1980.

- [19] T. Bektas, “The multiple traveling salesman problem: an overview of formulations and solution procedures,” *Omega*, vol. 34, no. 3, pp. 209 – 219, 2006.
- [20] W. Malik, S. Rathinam, 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.
- [21] 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.
- [22] S. Cardon, S. Dommers, C. Eksin, R. Sitters, A. Stougie, and L. Stougie, “A PTAS for the multiple depot vehicle routing problem,” Technische Universiteit Eindhoven, report, 2008.
- [23] S. Rathinam and R. Sengupta, “ $3/2$ -approximation algorithm for two variants of a 2-depot hamiltonian path problem,” *Operations Research Letters*, vol. 38, no. 1, pp. 63 – 68, 2010.
- [24] S. K. Yadlapalli, S. Rathinam, and S. Darbha, “An approximation algorithm for a 2-depot, heterogeneous vehicle routing problem,” in *ACC’09: Proceedings of the 2009 conference on American Control Conference*, St. Louis, MO, 2009, pp. 1730–1735.
- [25] N. Karmarkar, “A new polynomial-time algorithm for linear programming,” *Combinatorica*, vol. 4, no. 4, pp. 373–395, 1984.
- [26] H. A. Eiselt, M. Gendreau, and G. Laporte, “Arc routing problems, part i: The chinese postman problem,” *Operations Research*, vol. 43, no. 2, pp. 231–242,

1995.

- [27] S. Rathinam, “Routing and Monitoring Algorithms for UAVs,” Dissertation, University of California, Berkeley, 2007.
- [28] 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.
- [29] K. Helsgaun, “General k-opt submoves for the lin-kernighan tsp heuristic,” *Mathematical Programming Computation*, vol. 1, no. 2, pp. 119 – 163, 2009.
- [30] J. Siek, L. Lee, and A. Lumsdaine, *Boost Graph Library: The User Guide and Reference Manual*. Reading, MA: Addison-Wesley Professional.
- [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, July 2009.
- [32] J. Edmonds, “Paths, trees and flowers,” *Canadian Journal of Mathematics*, vol. 1, no. 17, pp. 449–467, 1965.
- [33] K. Helsgaun, “LKH,” July 2009, <http://www.akira.ruc.dk/keld/research/LKH/>.

VITA

Riddhi Rajeev Doshi was born in Petlad, India. After completing her schooling at Maharaja Agrasen Vidyalaya, Ahmedabad, India, she attended the North Gujarat University, Patan, India from 2003 to 2007. Here she majored in Mechatronics Engineering and received the degree of Bachelor of Engineering in June, 2007. She entered the graduate school at Texas A&M University, College Station in August, 2007 to pursue her interests in robotics and path-planning. She graduated with her Master of Science in 2010. She may be reached at:

Office Address:

55B Zachry Engineering Center

3127 - Texas A&M University

College Station, TX 77843-3127

Email: riddhi.r.doshi at tamu.edu

The typist for this thesis was Riddhi Rajeev Doshi.