University of Cincinnati

Date: 10/27/2011

I, Koushik Srinath Venkata Narasimha , hereby submit this original work as part of the requirements for the degree of Master of Science in Mechanical Engineering.

It is entitled:

Ant Colony Optimization Technique to Solve Min-Max MultiDepot Vehicle Routing Problem

Student's name: **Koushik Srinath Venkata Narasimha**

This work and its defense approved by:

Committee chair: Manish Kumar, PhD

Committee member: Sundararaman Anand, PhD

Committee member: Kelly Cohen, PhD

UNIVERSITY OF Cincinnati

2131

Ant Colony Optimization Technique to Solve Min-Max Multi Depot Vehicle Routing Problem

A thesis presented to

the faculty of

the College of Engineering & Applied Sciences

In partial fulfillment

of the requirements for the degree of

Master of Science

by

Koushik Srinath Venkata Narasimha

B.Tech, National Institute of Technology Karnataka(NITK), India

Autumn 2011

Committee Chair: Manish Kumar, Ph.D.

Abstract

This research focuses on solving the min-max Multi Depot Vehicle Routing Problem (MDVRP) based on a swarm intelligence based algorithm called ant colony optimization. A traditional MDVRP tries to minimize the total distance travelled by all the vehicles to all customer locations. The min-max MDVRP, on the other hand, tries to minimize the maximum distance travelled by any vehicle. The algorithm developed is an extension of Single Depot Vehicle Routing Problem (SDVRP) algorithm developed by Bullnheimer et al. in 1997 based upon ant colony optimization. In SDVRP, all the vehicles start from a single depot and return to the same depot, and solution aims at finding tours of vehicles so that every customer location is visited exactly once and that minimizes the total distance travelled. Building upon the SDVRP algorithm, this study first involves developing an algorithm for the min-max variant of SDVRP problem where the maximum distance travelled by any vehicle is minimized.

Later, the algorithm has been extended to address the Multi Depot variant of this problem. In this case, vehicles can start from multiple depots unlike SDVRP case and have to come back to their respective depot of origin once they visit a set of customer locations. The min-max multi-depot vehicle routing problem involves minimizing the maximum distance travelled by any vehicle in case of vehicles starting from multiple depots and travelling to each customer location at least once. This problem is of specific significance in case of time critical applications such as emergency response in large-scale disasters, and server-client network latency. The proposed algorithm uses an equitable region partitioning approach aimed at assigning customer locations to depots so that MDVRP is reduced to SDVRP. A background study on swarm intelligence based optimization techniques, region partitioning methods, approximation algorithms and also various techniques of optimization has been included in this research.

The proposed method has been implemented in Matlab for obtaining the solution for the min-max MDVRP with any number of vehicles and customer locations. A comparative study is carried out to evaluate the proposed algorithm's performance with respect to a currently available algorithm in literature in terms of the optimality of solution and time taken to reach the solution. Based on an extensive simulation study, it has been demonstrated that the ant colony optimization technique proposed in this thesis leads to more optimal results as compared to an existing method.

Acknowledgments

First and foremost, I owe my deepest gratitude to my research advisor Dr. Manish Kumar for his continuous support and encouragement throughout my research. His persistent guidance and motivation at various stages of my research transformed a challenging problem into this thesis.

I sincerely thank my thesis defense committee members Dr. Sam Anand and Dr. Kelly Cohen for their valuable suggestions and helpful hints in molding my thesis to its completion.

I would like to thank all my lab mates at the Cooperative Distributed Systems (CDS) lab for cooperating all through my study at UC and making my thesis completion successful. I specially thank Mr. Balaji Sharma, a PhD student at CDS and a good friend, for his suggestions and feedbacks in writing my thesis.

I would also like to thank my roommates and all buddies who made my stay at UC memorable with all the trips, fun-filled parties and provided support in innumerable ways to accomplish my goal of graduation.

Lastly, I thank my mom and dad, Mrs. Kumudini Srinath and Mr. Srinath, whose love and blessings were a constant driving factor in successful completion of the thesis. I also thank my elder sister Mrs. Rohini Ananthram, my cousins and other family members for their support and care while I am overseas.

Table of Contents

List of Figures

Chapter 1. Introduction and Motivation

Efficient transportation of products in and around the globe is an essential part of modern supply chain because of its high impact on cost and customer satisfaction by reduction of energy consumption and speedy delivery. With the growing business and hence increase in the complexity of transportation of materials, minimizing the cost of logistics becomes a significant factor in reducing the overall cost [1]. In the last decade, research [2] suggests that 10% to 15% of the traded goods correspond to the transportation costs. Therefore utilization of computerized methods for transportation will result in significant savings ranging from 5% - 20%. Recent study of U.S. Bureau of Labor Statistics estimates that transportation-related fields are growing by nearly 56,000 jobs a year thus showing an increase in trade and logistic businesses [3]. Realizing the importance of these factors, researchers have devoted a lot of effort in finding out novel and optimal ways for an efficient transportation. A well-known problem in this area, called Vehicle Routing Problem (VRP), has been studied for the past five decades. The problem involves finding tours of vehicles (with constraints on maximum distance they can travel) from a depot that visits a given number of delivery points and minimizes the total distance travelled. This problem was first proposed by Dantzig and Ramster in 1959 [4] as an extension of the classical Travelling Salesman Problem (TSP). Solving this problem, just like classical TSP, is computationally extensive and is known to be a Nondeterministic Polynomial (NP) hard problem [5]. NP Hard are those set of problems whose solutions require computational effort which increases in polynomial fashion with the size of problem. Hence an approximate solution is more desirable as it is considered to be a tradeoff to the computation time. The two basic types of VRP are: Single Depot (SDVRP) and Multi-Depot (MDVRP). SDVRP involves a single depot where all the vehicles are present initially while MDVRP consists of multiple depots. Multi-Depot Vehicle Routing Problem (MDVRP) extends the SDVRP by having multiple depots where multiple vehicles can originate from (and return to). In SDVRP, the objective is to find the minimum cost route that satisfies the three constraints, viz. every customer is visited exactly once, all vehicles begin and end at the same depot, vehicle capacity and distance restrictions are not violated. MDVRP as mentioned earlier consists of multiple depots with multiple vehicles and the objective remains the same as SDVRP. Interesting versions of these problems are the min-max version, where the objective of the problem is to minimize the maximum distance travelled by a vehicle (instead of total distance travelled as in a conventional VRP). This problem is often of interest when minimization of time taken to visit all points is more important than the total distance travelled, a usual factor of interest in emergency management situations. In emergency management, the objective is to use all available vehicles to minimize the time taken to attend to all points needing emergency resources. Other applications of this problem are in defense and computer networking. For example, assigning tours to a group of UAVs engaged in large scale surveillance operation by solving min-max MDVRP will minimize the maximum time of travel of UAVs, and hence help achieve desired objectives in time-critical scenarios. In computer networking, depots represent servers, vehicles represent data packets, and customers represent clients, a network routing topology generated by solving the min-max problem would result in minimizing the maximum latency between a server and a client. Traditional approaches to solving VRPs have involved the use of Linear Programming and Dynamic Programming based techniques. Some of the metaheuristic techniques have also been used lately. In recent years, optimization techniques that mimic swarming behavior in bio systems have gained lot of interest for applied problem solving. An artificial life program called ***Boids*** was developed by Craig Reynolds in 1986. This program simulates the flocking behavior of birds. In his paper published in 1987, he mentions an interesting observation about flocking of birds and the quote reads like this, "Perhaps most puzzling is the strong impression of intentional, centralized control. Yet all the evidence indicates that flock motion must be merely the aggregate result of the actions of individual animals; each acting solely on the basis of its own local perception of the world". The essence of this swarm optimization technique is the application of simple rules to individual agents with a decentralized control in order to achieve something that is of big picture. In nature, we observe several types of collective movements, for example, flocking of birds, fish schooling, and ant swarming. Foraging strategy in ants is a particularly interesting phenomenon. The branching bridge (See Figure 1-1) experiment by Deneubourg et al. in 1989 [6] demonstrated that the ants to reach point 2 from point 1 end up choosing one branch, the one which is shorter, over the other based on random fluctuations and pheromone guided selection of paths.. Pheromones are the chemicals deposited by certain species of ants. These pheromones are perceivable by other ants and also they evaporate over time. Over a period of time, due to back and forth travel of ants, pheromone trails are established and this determines the shortest path (optimal path) between nest and food source. On the basis of this research, Dorigo et al. successfully apply [42] this technique to solve various combinatorial optimization problems and thus establish this interesting metaheuristic optimization technique.

Applications of ant-colony based optimization technique are numerous. In the Harvard Business Review by Bonabeau [40], various possible applications of this method to practical business problems have been discussed. In the field of telecommunication network, to link a call from one place to another (say from Bangalore – San Jose), it has to be routed through various intermediate points (say Singapore, Honolulu etc.) to establish a connection. For such

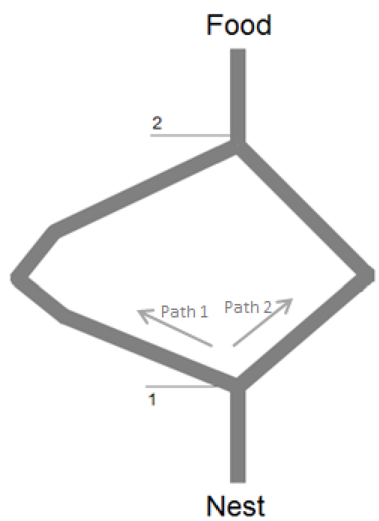


Figure 1-1: Binary Bridge Experiment: Ants Eventually Ended up Choosing Path 2 which was Shorter

systems, use of ant colony search techniques perhaps help in minimizing the latency between two stations. Also, consider dramatic phone traffic congestion scenarios because of inclement weather conditions or a phone-in television show which needs a quick rerouting of messages to a different network to handle this surge of traffic. As per the authors, Hewlett-Packard labs in Bristol, England have successfully applied the ant-foraging principles that routes these calls efficiently. Internet traffic is also similar to the telecommunication network and in fact on a larger scale. Ant-colony optimization technique has been applied successfully [40] to maximize throughput and minimize delays in handling internet traffic. Pena Petroli, a Swiss based heating oil distributor, uses ant-based programs [40] to direct the fleet of trucks in distributing oil to its customers. Variations of ant-foraging algorithms have also been applied to improve the factory efficiency. Unilever, a consumer goods company, were lacking the fast, efficient way to deploy equipment in one of its production facilities with the help of traditional optimization methods. Since the facility was very complex with lot of constraints including rates of operation of other equipment, changeover time for switching from one product to another, capacities, and preventive maintenance schedules, it was difficult to develop a traditional optimization program for this dynamic environment. Swarm intelligence technique was [40] used to develop software which adjusts the schedules quickly and automatically. This technique may also hold important applications in exploiting new markets. A classic example the authors provide is in the context of idea generation in Capital One. Here, the employees are responsible for their ideas and not the managers. People with better ideas should find sponsors, who are not necessarily their managers, and also recruit the people they need. In this system, the ideas which stand the best chance of revenue making will eventually attract more people, kind of analogy to a rich food source, and those ideas are transformed to new business units. This resembles a self-organizing behavior shown by ants. Thus there are myriads of applications of this swarm intelligence based techniques. ***These interesting behavioral consequences of biological swarming, specifically, phenomena such as ant foraging, and the immense potential these carry in solving complex optimization problems is the primary motivation for using this approach for solving the min-max MDVRP.***

Min-Max MDVRP was chosen as the thesis problem because of two reasons: i) the problem is very significant, and ii) the problem has been introduced recently, and very few prior researches have been conducted to address this problem. Considering the importance of this problem with various applications as mentioned in the earlier part of this chapter, a metaheuristic algorithm that can solve this problem has to be established in contrast to an exact algorithm in view of the nature of the problem. Ant colony optimization technique is a conspicuous option as this technique has been extensively applied to solve some of the well-known combinatorial optimization problems such as TSP and VRP. Some of the other approximation algorithms such as Genetic Algorithms, Tabu search, insertion algorithms, simulated annealing have already been applied to solve some of the variants of this problem. But, Ant colony technique, due to its guided search capability in addition to randomness and the usage of elitist phenomenon where the good solutions have more weights and influence search directions, is more suitable to solve the proposed min-max MDVRP.

The remainder of the thesis is organized as follows: First, the problem is introduced in Chapter 2. Later, an overview of the background of the problem and review of existing literature are presented with a brief introduction to various combinatorial optimization problems. A background study on different versions of the Vehicle Routing Problem and their formulations in Linear Programming framework has been carried out. Also, a discussion on the min-max MDVRP problem and methods available in literature has been discussed. Background studies on different types of optimization techniques and in particular, on ant colony optimization technique, have been done. Chapter 3 discusses the approach that was used in this thesis to solve the min-max MDVRP, which is the problem of interest. It starts with the approach to solve min-max SDVRP and has been extended to multi depot problem. This chapter also addresses some of the route improvement strategies that were used to obtain a better solution to the problem under consideration in this thesis. In Chapter 4, results and discussions that were obtained after rigorous simulation procedures have been studied. Following Results and Discussions chapter, the thesis has been concluded with a brief idea for future works that are possible on the basis of the proposed thesis in Chapter 5.

Chapter 2. Background and Literature Survey

2.1 A Typical Optimization Problem

Any decision making problem would involve finding out a best action to achieve certain goal. This best action needs to be performed with an objective function in place and finding out the maxima or minima of this function. These set of problems are typically called optimization problems. Mathematically, an optimization problem can be defined [7] as shown in Eq.(2.1)

If Objective function is $f(x)$, then

Find: *{min, max} $f(x)$*

$$\textbf{Subject to: } \text{Constraints - } g_i(x)\ \{\leq, =, \geq\}\ b_i,\ i = 1,2..m \quad (2.1)$$

$$\text{Variable bounds - } \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}$$

$$\text{Variable types} - x_j \text{ is } \{real, integer, binary\},\ j = 1,2...n$$

In short, an optimization problem involves finding out best solution in a set of feasible solutions, respecting certain constraints, bounds and variable types. To quote few applications - a decision making scenario that maximizes the business profit, to minimize the maintenance cost for a machinery, to minimize the total distance travelled in order to save fuel and time in case of a truck delivering goods.

2.2 NP Hard

Before the discussion of various optimization problems and techniques, it is better to know the term which describes the class of the problems based on the computational time and efficiency. Non-deterministic Polynomial (NP) is one of those terms which define the type of the optimization problem. A normally accepted algorithm for a given problem will generate a solution in a polynomial time. These classes of problems are called P-type problems. In layman terms, these are the problems that can be solved quickly. Most maxima-minima optimization problems come under this category. NP-Complete is another class of problems where there is no known algorithm that can generate the solution in a polynomial time. Hence, NP problems are the superset of NP-Complete and P-type problems. See Figure 2-1.

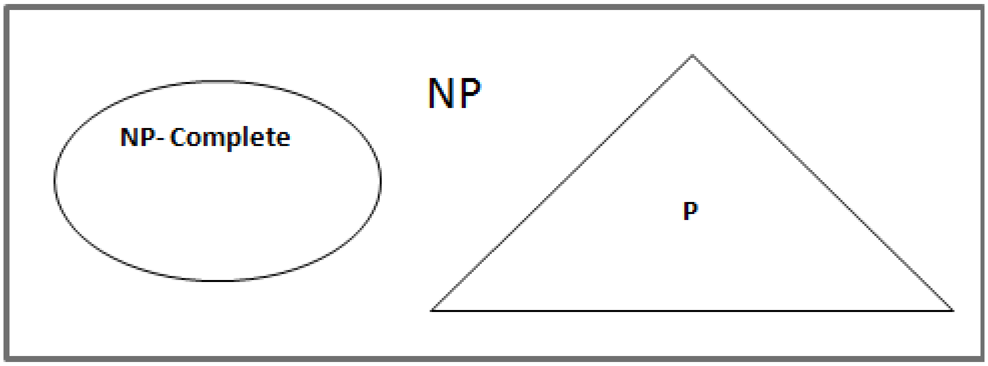


Figure 2-1: Classification of mathematical problems based on computational complexity

According to [5], a problem ***'Ω'*** is called a ***NP-hard,*** if a polynomial-time algorithm for problem ***'Ω'*** would imply a polynomial-time algorithm for every problem in NP. In short, these are a different set of problems that can have few NP type problems and these are actually NP-complete. See Figure 2-2.

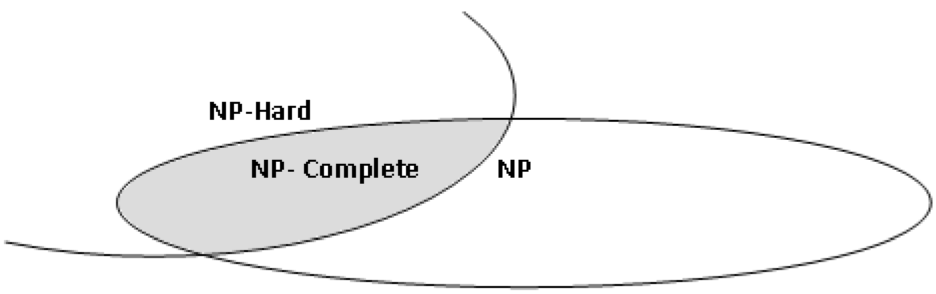


Figure 2-2: NP-complete are the intersection of NP and NP-Hard type of problems

2.3 Combinatorial Optimization Problems

In Applied Mathematics, a combinatorial optimization problem involves finding an optimal solution given a set of feasible solutions that are discrete. As per [8], a discrete-optimization problem is a problem of maximizing a real-valued objective function *f(x)*, on a finite set of feasible solutions *S.* This set S arises from a set of 2^E (the set of all subsets of E), for some finite ground set E, in which case it is called combinatorial-optimization problem. The applications of this type of problems are in operations research, machine learning, and artificial intelligence.

Discussed in the sections below are some of the well-known combinatorial optimization problems and their extensions in terms of dimensionality of the problem.

2.3.1 TSP

Travelling Salesman Problem (TSP) is a well-known combinatorial optimization problem. Here, the goal is to find a closed tour of minimal length connecting 'n' given cities or customer locations. Each city must be visited once and only once. The origin of this problem was from Irish Mathematician W.R. Hamilton and British Mathematician Thomas Kirkman. But this problem was formulated mathematically in 1930 by Karl Menger in Vienna and Harvard. This was later expanded by Whitney and Flood in Princeton [9]. A typical solution to TSP can be seen in Figure 2-3. TSP is considered as NP-hard problem in terms of computational complexity [5]. Starting from the solution to 49 cities in 1954 by Dantzig et al. [10], TSP has come a long way with the recent solution from Cook et al. in 2005 to solve 33810 cities instance [10]. This is shown in Table 1.

Table 1: Historical Solutions to TSP provided by various research teams

Year	Research Team	Size of Instance
1954	G Dantzig, R Fulkerson and S. Johnson	49 cities
1971	M Held and R. M. Karp	64 cities
1975	P.M. Camerini, L. Fratta,and F. Maffioli	67 cities
1977	M. Grötschel	120 cities
1980	H. Crowder and M.W. Padberg	318 cities
1987	M. Padberg and G. Rinaldi	532 cities
1987	M. Grötschel and O. Holland	666 cities
1987	M. Padberg and G. Rinaldi	2392 cities
1994	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	7397 cities
1998	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	13509 cities
2001	D. Applegate, R. Bixby, V. Chvátal, and W. Cook	15112 cities
2004	D. Applegate, R. Bixby, V. Chvátal, W. Cook, and K. Helsgaun	24978 cities
2005	W. Cook et al.	33810 cities*

A Linear Programming formulation of TSP as explained in [11] can be represented in the following way. Considering the TSP defined on n nodes belonging to set $N = \{1, 2... n\}$, arc set $E = N \times N$ with travel costs t_{ij} $((i,j) \in E; t_{ii} = \infty, \forall i \in N)$. In a regular TSP, if city 1 is the starting point of the travel, then the ending point is also city 1. Some of the other definitions pertaining to the problem formulation are:

M, which denotes the set of the remaining cities, $M = N \setminus \{1\}$,

$S = N \setminus \{n\}$ as the index set for the stage of travel corresponding to the order of visit of the cities in M and

$R = S \setminus \{n-1\}$

Let x_{is} $(i \in M, s \in S)$ be a binary variable $(0, 1)$ that takes the value *'1'* if city $i \in M$ is visited at stage $s \in S$.

Thus, travel costs considering all travel stages is shown in Eq. (2.2).

$$c_{isj} = \begin{cases} t_{ij} + t_{1,i}, s = 1, (i,j) \in M \times M; \\ t_{ij}, s \in R \setminus \{1, n-2\}, (i,j) \in M \times M; \\ t_{ij} + t_{j,1}, s = n-2, (i,j) \in M \times M. \end{cases} \tag{2.2}$$

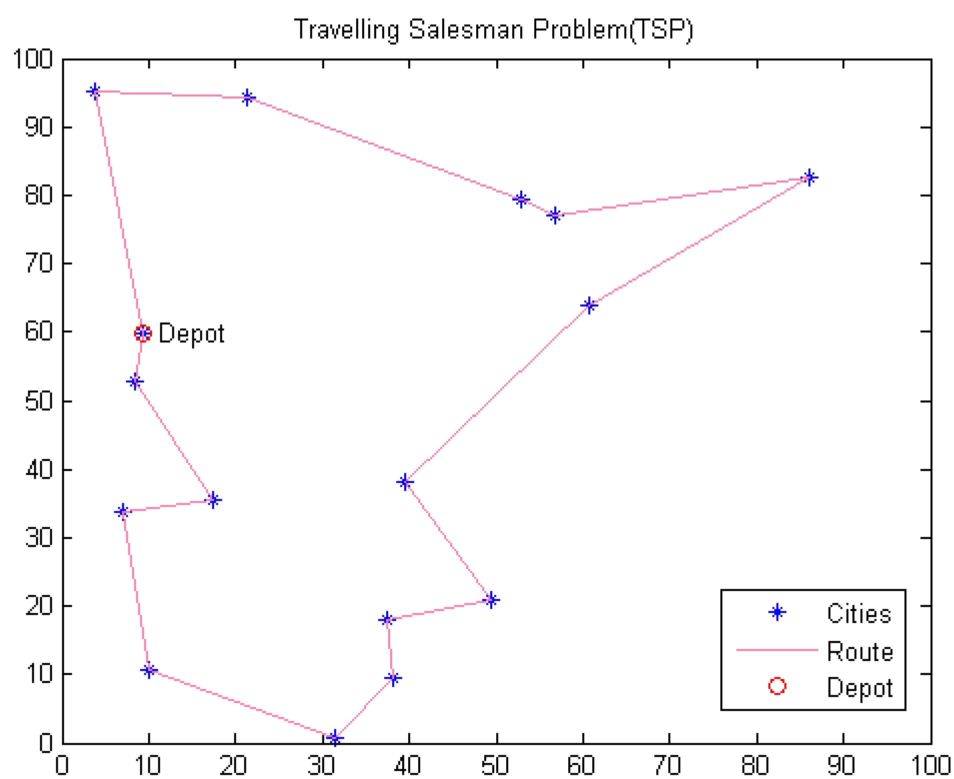


Figure 2-3: A typical TSP solution and also VRP solution with an unlimited fuel capacity

LP Formulation is represented in Eq. (2.3).

Minimize

$$Z_{TSP} = \sum_{s \in R} \sum_{i \in M} \sum_{j \in (M \setminus \{i\})} c_{isj} x_{is} x_{j,s+1} \tag{2.3}$$

subject to

$$\sum_{i \in M} x_{is} = 1 \quad s \in S \tag{2.4}$$

$$\sum_{s \in S} x_{is} = 1 \quad i \in M \tag{2.5}$$

$$x_{is} \in \{0,1\} \quad i \in M; \;\; s \in S \tag{2.6}$$

Eq. (2.3) aims at minimizing the total cost of all travels. Constraint (2.4) with (2.6) represents that only one city can be visited from city 1 and only one city is visited at each stage of travel. Constraint (2.5) enforces that a given city is visited at exactly one stage of travel. The terms in objective function represents that a travel cost is incurred between city *i* and city *j* if and only if those are visited at consecutive stages of travel.

2.3.2 VRP

Vehicle Routing Problem (VRP) is closely related to TSP because it consists of many TSPs with common start and end cities. In VRP, there is single depot (hence sometimes called single depot vehicle routing problem or SDVRP), and 'k' vehicles with capacity and distance restrictions. The objective is to find the minimum cost (of minimum overall travel distance) vehicle route so that: i) every customer location is visited exactly once; ii) all vehicles routes begin and end at the depot; iii) vehicle capacity and distance restrictions are not violated. VRP was first proposed by Dantzig in 1959 [4]. Since then, it has been studied extensively and serves as one of the benchmark problems in the field of optimization. These problems have influenced the emergence of fields such as operations research, polyhedral theory and complexity theory.

VRP in general can be formulated using Binary - Linear Programming technique. A generalized VRP formulation has been provided by Laporte in [21] and Kara in [22] . Here we discuss the formulation for a classical single depot vehicle routing problem with a constraint on the vehicle capacity (eg. fuel).

Consider a directed graph $G = (V,A)$, where $V = \{0,1,2,...n\}$ as set of vertices and $A = \{(i,j) : i,j \in V, i \neq j\}$ as the set of arcs. Node 0 represents the depot and remaining n nodes represent the customer points (cities to be visited). Hence $|V| = n+1$. The node set V is clustered into k mutually exclusive subsets where $V_0 = \{0\}$. There are m identical vehicles, each with capacity Q. c_{ij} is a non-negative cost associated with each arc $(i,j) \in A$.

Decision Variable x_{ij} can be defined as

$$x_{ij} = \begin{cases} 1, & \textit{if arc } (i,j) \textit{ is on the tour}, i \in V_p, j \in V_l, p \neq l, p,l = 1,2....,k \\ & 0, \textit{otherwise} \end{cases} \quad (2.7)$$

$$Minimize \sum_i \sum_j c_{ij} x_{ij} \quad (2.8)$$

Subject to

$$\sum_{i=1}^{n} x_{0i} = m \quad (2.9)$$

$$\sum_{i=1}^{n} x_{i0} = m \quad (2.10)$$

$$x_{ij} = 0 \; or \; 1 \quad \forall (i,j) \quad (2.11)$$

$$\begin{aligned} &|V_i| = 1, \quad \forall i \in V, \quad k = n, \\ &y_{ij} = x_{ij}, \quad \forall (i,j) \; pairs \end{aligned} \quad (2.12)$$

$$\sum_{j=0}^{n} x_{ij} = 1, \quad i = 1,2,\ldots,n \quad (2.13)$$

$$\sum_{i=0}^{n} x_{ij} = 1, \quad j = 1,2,\ldots,n \quad (2.14)$$

$$u_i + (Q - \overline{q_i} - q_i) x_{0i} - \overline{q_i x_{i0}} \leq Q - \overline{q_i}, \quad i = 1,2,\ldots,n \quad (2.15)$$

$$u_i + \overline{q_i} x_{0i} + (q_i + \overline{q_i}) x_{i0} \geq q_i + \overline{q_i}, \quad i = 1,2,\ldots,n \quad (2.16)$$

$$x_{0i} + x_{i0} \leq 1, \quad i = 1,2,\ldots.n \quad (2.17)$$

$$u_i - u_j + Q x_{ij} + (Q - q_i - q_j) x_{ji} \leq Q - q_j, \quad i \neq j, i,j = 1,2,\ldots,n \quad (2.18)$$

Where

$$\overline{q_i} = \min_{j, j \neq i} \{q_j\} \quad (2.19)$$

u_p: *Load of a vehicle just after leaving cluster p or unloaded amount from the vehicle just leaving cluster p,*

p =1, 2… k

q_l: *demand of cluster l,*

i.e. $q_l = \sum_{i \in V_l} d_i, \quad l = 1,2,\ldots,k$

d_i: *demand of customer i, i = 1,2,…n*

In the above formulation, since the cardinality of cluster is 1, demand of the customer and demand of the cluster are the same. Also capacity Q can be the distance constraint on the vehicle due to fuel or other reasons. And c_{ij}, the cost variable is the distance between the customer i and customer j. Thus the load u_p of the customer is nothing but the difference between total capacity of the vehicle and the distance travelled to reach the customer prior to the current customer. Also, demand of the customer is nothing but the distance travelled by the vehicle from the previous position to current position. Equations (2.9) - (2.11) represent the degree constraints on the problem. It implies that there are *'m'* leaving arcs from and *'m'* entering arcs to the home city. Equations (2.13) - (2.14) also represent the degree constrains implying that there can only be a single incoming or outgoing arc to a cluster from any other node belonging to other clusters except V_0. Equations (2.15) - (2.18) represent the well-known subtour elimination constraints.

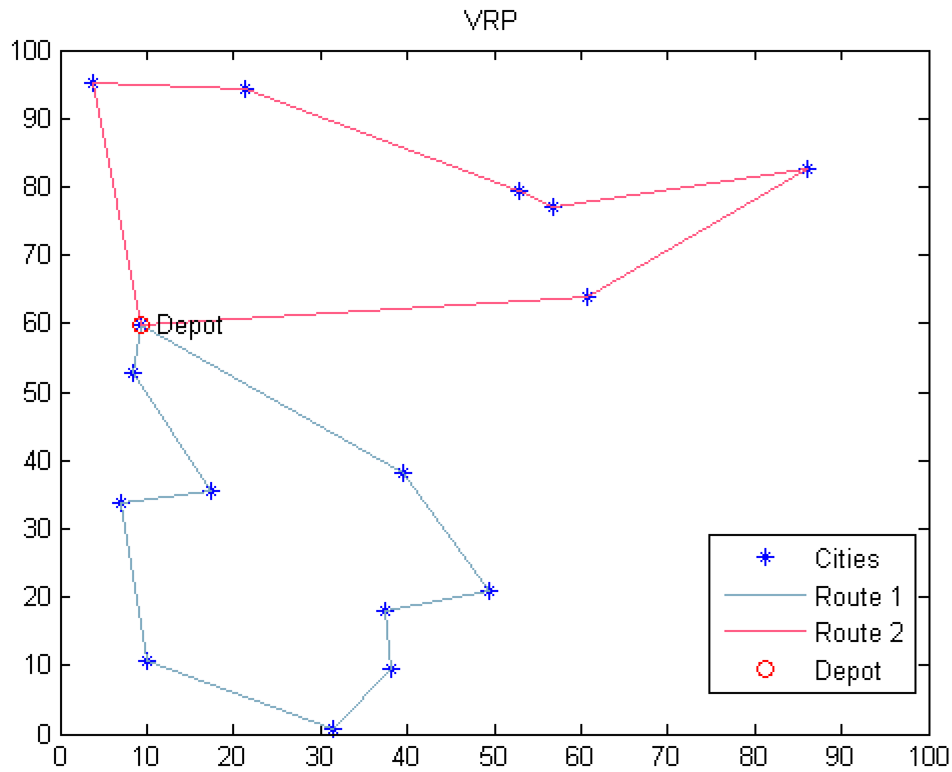


Figure 2-4: A VRP solution for 2 vehicle problem

2.3.3 Variants of VRP: MDVRP, VRPTW

The common form of VRP is the capacitated VRP (CVRP) wherein a set of vehicles of uniform capacity must visit customer places once and only once from a common depot at a minimum transit cost [12]. Multiple Depot VRP (MDVRP) is an extension of the CVRP with more number of depots. The objective here is to visit all the customer locations once and only once while minimizing the number of vehicles and travel distance. MDVRP can be traced back to 1976 when Gillet and Johnson published a paper on Multi Terminal Vehicle-Dispatch Algorithm [13]. In this paper, a heuristic algorithm was developed to obtain an approximate solution. Their purpose was to determine a set of routes by which vehicles from two or more terminals can service a collection of demand points keeping in mind the total distance traveled needs to be minimized. They employed a sweep algorithm where in the strategy was to partition the problem to single-terminal problem to significantly reduce the computation time. The solution was also extended to satisfy some of the constraints such as the vehicle load and the length of each route. After this paper, much effort has been dedicated by researchers around the globe and many have come up with different methods to solve this problem in an optimal way. [14], [15], [16] are some of the different methods that were used to solve MDVRP. In 2005, Lim and Wang [17] proposed a more practical variant of this problem and it was named MDVRP with fixed distribution of vehicles (MDVRPFD). They proposed this problem with a bound on the number of vehicles in a depot unlike the traditional MDVRP where the limit was unrealizable infinite number of vehicles. With an assumption of exactly one vehicle in each depot, they developed a binary programming model to obtain the solution and generalized the solution for *n* number of vehicles in a depot. The LP formulation of MDVRP as shown in [17] is as follows.

Consider *M* depots denoted by set *D*, and each depot *k* has exactly one vehicle v_k. Each vehicle v_k has a route R_k. There are *N* customers denoted as set *C*. The total demand on each route R_k does not exceed the capacity of the serving vehicle, Q_k. Distance between the points (customers and depots) is $c_{i\,j}(i,,j \in C \cup D)$. Demands of the customers is d_i $(i \in C)$.

$$\text{Minimize} \sum_{i \in C \cup D} \sum_{j \in C \cup D} c_{ij} \sum_{k \in D} x_{ijk} \tag{2.20}$$

Subject to

$$\sum_{i \in C \cup D} \sum_{k \in D} x_{ijk} = 1 \quad \forall j \in C \tag{2.21}$$

$$\sum_{i \in C} \sum_{j \in D} x_{ijk} \leq 1 \quad \forall k \in D \tag{2.22}$$

$$\sum_{j \in C \cup D} x_{ijk} = \sum_{j \in C \cup D} x_{jik} \quad \forall k \in D, \quad i \in C \cup D \tag{2.23}$$

$$\sum_{j \in C} d_j \sum_{i \in C \cup D} x_{ijk} \leq Q_k \quad k \in D \tag{2.24}$$

$$\sum_{i \in S} \sum_{j \in S} x_{ijk} \leq |S| - 1 \quad \forall S \subseteq C, \quad |S| \geq 2, \quad k \in D \tag{2.25}$$

$$x_{ijk} \in \{0,1\} \quad \forall i, \quad k \in D, \quad j \in C \tag{2.26}$$

The decision variable $x_{ijk} = 1$ *iff i* immediately precedes customer *j* on the route R_k $(i,j \in C \cup D,\ k \in D)$; 0 otherwise. In the above formulation, constraints (2.21) - (2.23) imposes that each customer is visited exactly once, each route is served by at most one vehicle and same vehicle enters and leaves the customer respectively. Eq. (2.24) is for the capacity restriction for each vehicle. Eq. (2.25) is for the generalized sub-tour elimination constraint which restricts that each vehicle v_k has at least one route that leaves each customer set S visited by v_k . In this paper, they also proposed a new one-stage approach where in the assignment of customers to depots and route calculations were done at the same level. VRP with time windows (VRPTW) is also a similar type of problem with an additional restriction in terms of time window associated with each customer. The objective here is to minimize the number of vehicles used and sum of travel time and waiting time needed to visit all customers.

2.3.4 Min-Max VRP

There is another important variant of VRP closely related to VRPTW called Min-Max VRP. This problem was first formulated Carlsson et al. in 2007 [18]. Unlike the traditional objective of minimizing the tour lengths of the vehicles, the purpose of this problem is to minimize the maximum length of a tour in VRP. Carlsson et al. developed two different heuristics to solve min-max MDVRP. The first heuristic is a linear programming based technique. Using this algorithm, the customers are assigned to depots and TSP routes are generated for each vehicle and its respective customers. Second heuristic namely region partition method is also developed which helps in generating fast approximate initial solutions. On the basis of a popular BHH theorem (1959) by Beardwood, Halton and Hammersley, they show that theoretically there exists a lower and upper bounds to the longest tour length, L, which denotes the optimal solution to min-max MDVRP. Using this upper bound, they developed a tour partition heuristic which generates a feasible solution for the MDVRP. Building upon this, they developed an asymptotic bound for L and they conclude that the optimal solution to min-max MDVRP with uniformly distributed points will numerically approach to a value proportional to $\sqrt{n}/k$, which is the value of optimal TSP tour of all customers split by number of vehicles, under the constraint. Later they developed two different heuristics to solve the min-max MDVRP.

The first heuristic was a linear programming based load balancing technique. In this heuristic, it is believed that the load on the vehicle must be balanced in terms of the nodes that each vehicle has to serve. This intuition is valid considering the uniform distribution of cities (or nodes). This technique is further enhanced with global adjustments and generate near-optimal route with concorde TSP solver.

To mathematically formulate this problem, consider 'n' customer/delivery points, 'm' depot points and a total number of *'k'* vehicles at the depots. All the vehicles are initially located at their respective depots. The vehicles are required to visit all customer points and return to the same depot from where they started their journey. The problem as stated earlier is to decide the tours of each vehicle so that the maximum distance travelled by any vehicle is minimized. Mathematically, the aim is to

$$\begin{aligned} &\text{Minimize } \lambda \\ &\text{subject to } TSP(S_i) \le \lambda, \forall i \\ &\cup S_i = N \end{aligned} \tag{2.27}$$

where N is the set of all customer locations, |N| = n, S_i $(\subset N)$ is the subset of customers assigned to vehicle *'i'* and $TSP(S_i)$ is the length of the optimal tour for a single Traveling Salesman Problem applied to vehicle i and the customer set S_i.

Similarly, the min-max MDVRP can also be formulated in LP framework. Consider uniformly distributed customer and depot locations. Initial assumption is depot has exactly one vehicle but this can be generalized later to any number of vehicles. Each node X_j is located at (x_j, y_j) where $1 \le j \le n$ and each vehicle V_i is located at (v_i, w_i) where $1 \le i \le k$. Here *'k'* is number of depots and *'n'* is number of customer points. The distance c_{ij} between X_j and V_i is given by $c_{ij} = \left\| (x_j, y_j) - (v_i, w_i) \right\|$

Consider decision variable x_{ij} to be a binary variable indicating if node X_j is assigned to vehicle V_i or not.

Thus,

$$\text{Minimize } \sum_{1 \le i \le k, 1 \le j \le n} c_{ij} x_{ij} \tag{2.28}$$

Subject to,

$$\sum_{1 \le j \le n} x_{ij} = \frac{n}{k}, \quad \forall i, \tag{2.29}$$

$$\sum_{1 \le i \le k} x_{ij} = 1, \quad \forall j, \tag{2.30}$$

$$x_{ij} \in \{0,1\}, \quad \forall i, j, \tag{2.31}$$

In (2.29), *n/k* fraction is assumed as an integer. This can also be generalized any fractional case.

The second and an intriguing heuristic is based on the region partitioning based method. In this technique, noticing that a convex equitable partition yields an even division of points, they divided the service region into a set of subregions with equal area and generate good initial solutions by assigning the customer points in the depot region to the respective depot.

2.4 Optimization Techniques

Optimization techniques are the means by which the objective function can be maximized or minimized.

2.4.1 Traditional Techniques

The basic optimization problem being unconstrained optimization problems can be solved using **Differential Calculus** techniques. Other simple problems which have equality constraints (=) can be solved using techniques such as **Lagrangian Multipliers**. A bit more complex versions of optimization problems would involve inequality (≤ or ≥) relationships for the constraints. These kinds of problems can be mathematically computed using **Linear Programming techniques**. In linear programming technique, the objective function and constraints can be expressed as linear function of decision variables. These equations can be iteratively solved using well known methods such as simplex method. Integer programming problems, quadratic programming problems are variants of Linear programming techniques that can be used to solve some of the constrained optimization problems. These solutions are considered are *'exact'* algorithms to solve optimization problem. As the current research is focused on combinatorial optimization problems such as TSP and VRP, it is important to know some of the previously attempted exact algorithms to solve these problems. In [20] and [21], Laporte covers various exact algorithms that were attempted to address TSP and VRP. For TSP, listed are some of the algorithms. a) Integer linear programming formulations [23] b) the branch-and-bound algorithms [24] c) the shortest spanning tree bound and related algorithms [25][26] d) the 2-matching lower bound and related algorithms [27][28]. For VRP, Laporte discusses following exact algorithms. a) The k-degree center tree and a related algorithm [29], and b) A three-index and a two-index vehicle flow formulation [30].

2.4.2 Approximation Algorithms

To counter the difficulty in generating exact solutions for NP-hard problems, approximate algorithms are developed which are suboptimal, for example within 2~3% of optimal solution. These algorithms are the methods through which a set of approximate solutions are recognized for an optimization problem defined in 2.1. Some of the initial approximate solutions for NP problems such as Subset-Sum, Bin-Packing, Maximum Satisfiability, Set Covering, Graph Coloring, and Maximum Clique can be seen in [19]. To solve combinatorial optimization problems such as TSP and VRP, plenty of approximation algorithms have been used. In [20] and [21], Laporte discussed various approximate algorithms that were used to solve TSP and VRP respectively. Here is a brief overview of some of them. He divided the heuristic or approximation algorithms into three classes: (i) tour construction procedure, (ii) tour improvement procedure, and (iii) composite algorithms. In tour construction, TSP solution is built by choosing next vertex at each step. In tour improvement procedure, the solution is improved by performing various exchanges. Composite algorithm would involve combining these two. Some of the tour construction procedures which the paper deals with are: a) the nearest-neighbor algorithm [31] – in this technique, next vertex is chosen based on the decision which is most advantageous; b) Insertion algorithms [31] – choose the third vertex based on some criteria like vertex yielding the least distance increment, vertex closest to the current tour; c) The patching algorithms[32] – this employs the notion that the assignment relaxation of the TSP provides a near optimal solution if the elements c_{ij} in cost matrix *'c'* are uniformly distributed. Some of the tour improvement procedures that Laporte discusses include: a) the r-opt algorithm [33] – in this case, for any given tour, any *'r'* arcs are tentatively removed from the tour and they are reconnected in all possible ways to find a better solution. This strategy has been used in the proposed research and has been explained in the form of 2-opt heuristic route improvement strategy under Chapter 3. b) Simulated annealing [34] – this technique is used in analogy to the annealing technique used to obtain material stability. Here, the objective is to gradually perform changes to an initial solution in order to obtain minimum cost solution. c) Tabu search [35] – in this method, a list called 'tabu list' is constantly updated by inserting those examined solutions that are forbidden or suboptimal in order to avoid recycling of these sub-optimal solutions. Composite algorithms utilize both tour building and tour improvement strategies. In the current proposed research, composite algorithm has been used to achieve an optimal solution. Some of the VRP approximation algorithms which are discussed in Laporte's paper [21] are a) Clark and Wright algorithm [36] – in this method, vehicle routes start with a vertex and the depot. At each iterative step, two routes are merged based on the largest saving that can be generated. b) The Sweep algorithm [37] - here they represent the vertices by their polar coordinates (θ_i, ρ_i) where θ_i is the angle and ρ_i is the ray length. By assigning the initial angle of 0^o to an arbitrary vertex and calculating the remaining angles, this algorithm ranks the vertices in increasing order of their θ_i. c) The Christofides-Mingozzi-Toth two phase algorithm [38] – this algorithm solves CVRP by constructing algorithm in two phases namely sequential route construction and parallel route construction.

In late 90s, biologically inspired algorithms were published motivated by some of the naturally observed phenomena such as intelligence of swarm of birds in synchronized motion or swarm of ants in foraging etc. The sections which follow discuss in detail the theory and background of some of these algorithms as the proposed thesis attempts to solve the min-max MDVRP problem using one of the swarm intelligence techniques called Ant Colony Optimization. Therefore it is necessary to know the basics of these algorithms.

2.5 Swarm Intelligence Based Algorithms

Swarm intelligence is term used to describe a phenomenon associated with the naturally occurring intelligent behavior of biological species. This term was first used by Gerardo Beni and Jing Wang in [39]. Some of the behaviors like decentralization, cooperation, self-organization are typical for species such as ants, birds, and bacteria. These behaviors can be applied to the field of artificial intelligence to develop intelligent algorithms that can be applied to solve certain optimization problems in many fields. Some of the notable algorithms are Particle Swarm Optimization, Artificial Immune Systems, and Ant Colony Optimization. In [40], Bonabeau et al. applied optimization algorithms on businesses such as Southwest airline and Unilever in which they utilized social behavior of insects in order to optimize the cargo handling and factory equipment scheduling respectively. According to them, the term swarm intelligence can be paraphrased in the following way – "Social insects work without supervision. They achieve teamwork by individual interactions and with no connection to a non-neighbor. But collective behavior emerged from this group can be used to find efficient solutions to various problems and this emergent behavior can be termed as Swarm Intelligence". In [41], the authors enlist some common properties found in the swarm intelligent systems. They are a) system composes of many relatively homogenous individuals b) simple rules of interaction among individuals with each other or via the environment (stigmergy), and c) Absence of a coordinator or an external controller. Swarm intelligence has been applied in various fields such as airline cargo scheduling, crowd simulation in movies, communication networks, truck dispatching and many more. In fact, in the paper [40], it is said that applications of swarm intelligence are limited only by the imagination. This thesis uses one of swarm intelligence techniques called Ant Colony Optimization which originates from Ant System to solve the min-max MDVRP. In what follows, a brief explanation of various concepts involved in Ant Colony Optimization is provided.

2.5.1 Ant System

The concept of Ant System was first introduced by Dorigo et al. in [42]. The basic idea underlying this ant based algorithm is to use a positive feedback mechanism, based on an analogy with the trail laying trail following behavior of some species of ants and some other social insects, and to reinforce those portions of good solutions that contribute to the quality of these solutions [41]. Since the algorithm involves simple rules of each ant with decentralized control, this technique is computationally efficient and fairly easy to implement. Some of the desirable characteristics of these ant systems as highlighted by [42] are versatility, robustness and decentralization. The search activity is carried using agents called ants. It has been observed in ant colonies that ants, in presence of multiple food sources, eventually pick one that is optimal in some manner such as proximity to the nest or the quality of food. This is an interesting observation that was studied by ethologists and it was proven that the ants leave a trail that consists of pheromone, a chemical which evaporates as a function of time. These pheromones act as a medium of communication between individual ants. An isolated, randomly moving ant encountering this previously laid pheromone trail will choose this path with high probability. Thus the more they travel along this path, the more pheromone is deposited and hence reinforcing the pheromone concentration. In [42], Dorigo et al. also showed this idea of ant foraging experimentally in analogy to the short distance problem. Figure 2-5 and Figure 2-6 explains the analogy between the real ants and the corresponding simulated ants' scenario. Essentially, the idea behind this algorithm is that, at any given instance, the probability of an ant choosing a path out of many paths will be based on path chosen by its preceding ants. This idea has been efficiently used in developing algorithms for various combinatorial optimization problems such as Travelling Salesman Problem (TSP), Quadratic Assignment Problem (QAP), Job-Scheduling Problem, Vehicle Routing Problem etc. Considering the relevancy to the proposed thesis, a brief explanation of Ant System/Ant Colony algorithm that was used to solve TSP and VRP has been provided in section 2.6.

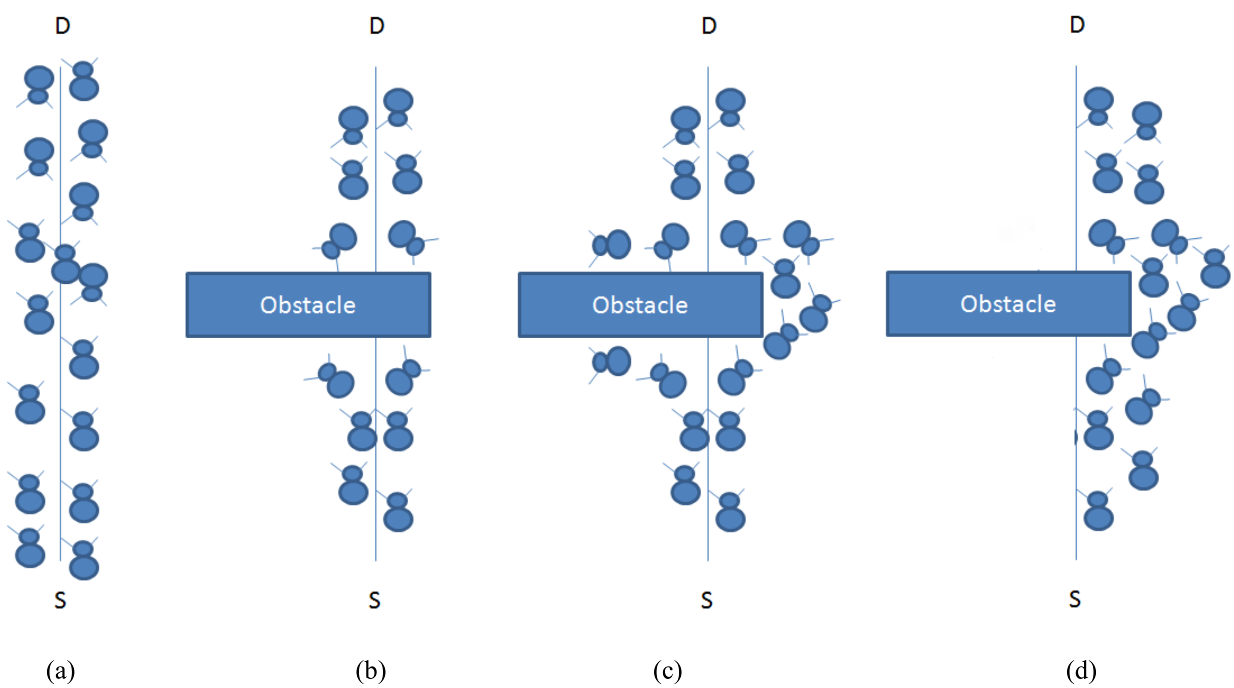
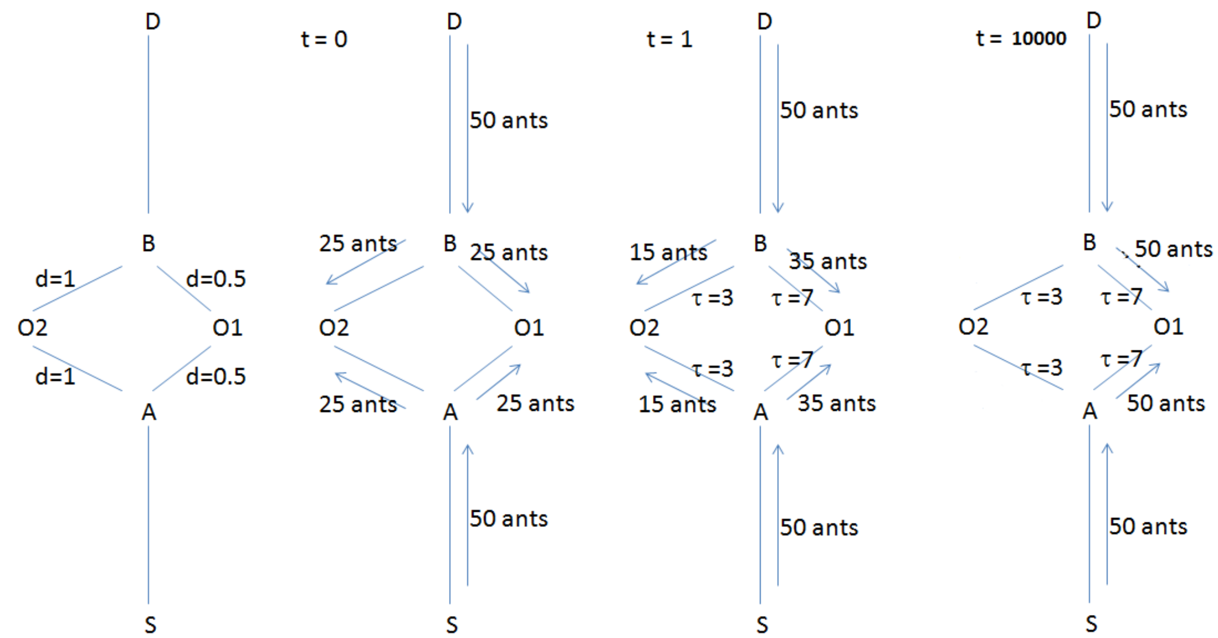


Figure 2-5: Ant Foraging Behavior

(a) Ants moving from Start point (S) to Destination (D) (b) An obstacle is introduced (c) Ants try to find a path avoiding the obstacle (d) All ants converge to the same path



(a) (b) (c) (d)

Figure 2-6: Probabilistic Representation of Ant Foraging Behaviour in terms of Pheromone Concentration

(a) **Hypothetical distances from A-O1 is 0.5, and A-O2 is 1. (b) At t=0, 50 ants choose either of the paths AO1 or AO2 with equal probability (c) At t=1, 35 ants choose path AO1 as the pheromone concentration τ is more (d) At t = large, all the ants choose this path**

2.6 Application of Ant System based Algorithms to solve TSP and VRP

2.6.1 Application to TSP

Swarm intelligence using the foraging strategies of ants was first applied to TSP [43]. This optimization technique is called ant-colony optimization. In this technique, as explained in the earlier section, artificial ants search the solution space. Artificial ants are slight different than biological ants. For instance, artificial ants are endowed with working memory which can be used to memorize the already visited cities. This memory M_K is updated after every time step. In the ant colony system, according to Dorigo et al., an ant '*k'* in city '*r'* chooses the city '*s*' based on three variables. First, the tabu list, which is nothing but the list of cities that are not visited: the set of cities J_i^k that ant *k* still has to visit when it is at city *'r'*. Second is visibility, which represents the local information and is the heuristic desitability to visit city *'s'* when in city *'r'*. Third, pheromone trail *φ(r,s)* at time *'t'* for each edge measure the learned desirability to visit city *'s'* when in city *'r'*. Following probabilistic formula shown in Eq. (2.32) is universally applied to different applications with small modifications.

$$s = \arg\max\left\{[\tau(r,u)]\cdot[\eta(r,u)]^{\beta}\right\} \text{ if } q \le q_0$$
$$else\ s = S \tag{2.32}$$

where $\tau(r,u)$ is the amount of pheromone trail on edge *(r,u)*, $\eta(r,u)$ is a heuristic function, which is inverse of distance between cities *r* and *u*, *β* is a parameter which weighs relative importance of pheromone trail and distance, *q* is a random value, q_0 is a parameter ($0 \le q_0 \le 1$). Function *argmax* in (2.32) represents the exploitation which favors transition towards cities that are at shorter distances and the edges which have large amount of pheromone. Biased exploration is represented by the *'else'* term in (2.32). The factor *β* represents weightage of heuristic desirability in comparison to pheromone concentration. The parameter q_0 determines the relative importance of exploitation versus exploration. *S* is a random variable selected according to the probabilistic distribution function shown in Eq. (2.33)

$$p_k(r,s) = \begin{cases} \dfrac{[\tau(r,s)][\eta(r,s)]^{\beta}}{\sum_{u \notin M_k}[\tau(r,u)]\ [\eta(r,u)]^{\beta}} & if\ \ s \notin M_k \\ else \quad p_k(r,s) = 0 \end{cases} \tag{2.33}$$

where $p_k(r, s)$ probability with which ant k chooses to move from city *r* to city *s*.

Pheromone trail is updated locally as well as globally. Global updating is required in order to reward edges that belong to short tours. It is similar to reinforcement learning where the better solutions get reinforced. Global pheromone update formula is given by Eq. (2.34)

$$\varphi(r,s) \leftarrow (1-\alpha)\cdot\varphi(r,s) + \alpha\cdot\Delta\varphi(r,s)$$
$$\text{where } \Delta\varphi(r,s) = (shortest\_tour)^{-1} \tag{2.34}$$

Local trail updating is similar to the pheromone evaporating behavior. It is applied in order to avoid greedy search and to restrict the ants from choosing the strongest edge always. This is given by the formula as shown in Eq. (2.35).

$$\tau(r,s) \leftarrow (1-\alpha)\cdot\tau(r,s)+\alpha\cdot\tau_0 \tag{2.35}$$

where τ_0 is a parameter.

In general, any ant system algorithm typically consists of heuristic desirability, probabilistic transition rule (ant interactions), and global and local pheoromone update rule. Using these functions, in [43], Dorigo et al. solves the TSP with much optimized results upto 1577-city problem. In another paper by Dorigo et al. [44], they share the complete detailed algorithm that was used to solve the TSP. This algorithm forms the basis of the proposed thesis in order to solve VRP and consequently min-max MDVRP. In this paper, they analyze the computational performance in comparison with different algorithms such as simulated annealing, elastic net, self-organizing maps etc. Based on the performance analysis, a disadvantage they found is getting stuck in local minima. This issue has been addressed using several approaches such as introduction of randomness in choosing the next city as well as via mechanisms such as *r*-opt techniques [33] as explained in 2.4.2. These 2-opt and 3-opt techniques are explained in detail in section 3.3.2 of this thesis. Also they mention these heuristics in order to classify ant colony system which according to them is tour constructive heuristics and these algorithms such as *r*-opt techniques are tour improvement strategies. This paper brings out the idea of merging tour constructive and tour improvement heuristics in order to obtain an optimal solution. In [45], Stutzle clearly distinguishes the differences between Ant System and Ant Colony Optimization techniques. Also, he explains couple of tour improvement strategies such as *Candidate lists* and *Fastening the trail update*. Stutzle quotes three versions of Ant System (AS) when it was first introduced. They are *ant-density, ant-quantity and ant-cycle.* In *ant-density* and *ant-quantity,* the pheromone is updated after every ant transition from one city to another. But in *ant-cycle,* the pheromone update is done only after the ants construct the complete tour and the amount of pheromone deposited is a function of the tour quality. Since *ant-cycle* performed better compared to the other two, this was eventually called as Ant System technique. Ant Colony System (ACS or ACO) is an improvement to the performance of Ant System. The improvement is in terms of the pheromone update. In ACO, the concept of *elitist strategy* was introduced which was already discussed at the beginning of this section called global pheromone update rule. The notion is to reinforce those set of arcs that belong to the best tour since the start of the algorithm. Thus Stutzle in [45] bring out three major distinctions that can be marked from AS to ACO. First, an aggressive action rule of ACS. Second, the pheromone-update methodology. Third, pheromone evaporation, which is proportional to time, for all the arcs.

Even though the improvisation techniques will be in discussed in detail in further sections, a brief explanation of two improvisation strategies dealt in this paper [45] can be provided at this point of time. Candidate lists, for every city, typically consists of a fixed number of neighboring cities in increasing order of distances. Instead of choosing a next city from an entire set of cities, using this list for choosing a city from the current city will considerably improve the performance Another strategy called fastening the trail update is application of pheromone evaporation to the arcs that connect from city *'i'* to *i's* candidate list. This reduces the computational time from order $O(n^2)$ to $O(n)$.

2.6.2 Application to VRP

ACO algorithm was applied to solve SDVRP by Bullnheimer et al. in 1997 [46], where a hybrid ant system algorithm was presented and then improved using specific information (savings, capacity utilization). In their approach, the artificial or simulated ants construct vehicle routes by successively choosing the next cities using a probabilistic transition rule similar to TSP. If the probability *'p'* value for transition to a particular j^{th} city from the i^{th} city is higher, then chances of that city getting chosen as the next city is more. In 2004, Bell and McMullen [47] developed multiple ant colony method to solve the vehicle routing problem. This method used separate specialized ant groups with unique pheromone deposits for each vehicle to solve the VRP. This separation was intended to differentiate paths typically used in first vehicle route from those used by subsequent vehicles. It is believed that this technique is more useful when the size of the problem and the number of vehicles required increase. Also, their paper reinstated the route improvement strategies such as 2-opt heuristic and candidate list technique which were initially proposed by Bullnheimer et al. The same has been applied in this thesis in obtaining more accurate results for the VRP solution. In [48], Rizzoli et al. studied ACO application to variants of VRP such as CVRP, VRPTW, TDVRP and DVRP. They presented two case studies for VRP with time windows. And finally they showed the application of ACO to industry size problems for a super market chain in Switzerland and Pick-up and Delivery problem for an industry in Italy. They showed the improvement in vehicle route performances by ACO application and thus proving that this is a process for strategic planning in real world applications. They used an interesting heuristic desirability function based on the savings algorithm in [49]. This is given by Eq. (2.36)

$$\eta_{ij} = d_{i0} + d_{0j} - g \cdot d_{ij} + f \cdot \left| d_{i0} - d_{0j} \right| \quad (2.36)$$

where g and f are parameters.

2.7 Introduction to Computational Complexity

Study of computational complexities of different techniques to solve these combinatorial optimization problems is by itself an important area of research. To give a basic example of the determination of computational complexities, let us consider a TSP example. Through a brute force method, the possible number of tours and thus the order of complexity is shown in the below table.

No. of cities	Possible number of tours
3	1
5	12
10	181440
15	43589145600
35	14761639951980200000000000000000000000
50	304140932017134000000000000000000000000000000000000000000000000

Through is method as we can observe that the complexity is of the order of *O(n!)*. Similarly, It can be shown that the use of dynamic programming method to calculate TSP is of the order of $O(c^n)$ $c>1$. But the application of ant colony method to TSP reduces the number of function evaluations to $O(r*n^2)$. This is because the objective function considered for every local search in terms of Ant Colony Optimization is linear, which means the order of complexity is *O(n),* and this kind of search is carried out *n* times because of *n* ants(considering no. of ants = no. of cities) which makes it order $O(n^2)$ complexity. Considering *r* iterations, the number of function evaluations is at most by a factor of $O(r*n^2)$. It is interesting to analyze the computational complexity of ACO techniques for this min-max MDVRP. A thorough study in the similar area was conducted by Neumann et al. in [59]. Here they show the analysis of computational complexities of simple ACO algorithms applied to variants of TSP.

Chapter 3. Approach

3.1 Min-Max SDVRP

Ant colony optimization technique is used in this research to solve the min-max MDVRP. The approach is based on the ant colony optimization method used by Bullnheimer et al. [46] in solving the traditional SDVRP. The traditional SDVRP can be represented by a complete weighted digraph $G = (V,A,d)$ where $V = \{v_0,v_1,..,v_n\}$ is a set of vertices and $A = \{(v_i,v_j) : i \neq j \}$ is a set of arcs. The vertex v_0 denotes the depot; the other vertices of V represent cities or customers. A cost variable is associated with each arc called d_{ij} that can represent distance or travel time between city i and city j. The aim is to find minimum cost vehicle routes. Artificial ants searching the solution space simulate real ants searching the environment; the objective value corresponds to the quality of food sources and an adaptive memory corresponds to the pheromone trails. To aid the search procedure through a set of possible solutions, the artificial ants are provided with local heuristic functions. Ant colony algorithms typically use some parameters that include heuristic desirability, pheromone updating rule, and probabilistic transition rule as explained in 2.6.1. The definitions of these parameters are problem specific and are critical to solve the optimization problem in consideration. For the min-max SDVRP, the definitions of these parameters are similar to the TSP problem. The heuristic desirability of visiting city j from city i (or the visibility) is represented by η_{ij} and is equal to reciprocal of d_{ij} as shown in Eq. (3.1).

$$\eta_{ij} = \frac{1}{d_{ij}} \tag{3.1}$$

Another heuristic desirability function that can also be implemented to solve this problem is given by Eq. (3.2)

$$\eta_{ij} = d_{i0} + d_{0j} - g \cdot d_{ij} + f \cdot \left| d_{i0} - d_{0j} \right| \tag{3.2}$$

Where d_{i0} is the distance between depot and city i. g and f are optimizing parameters.

The probabilistic transition rule is indicated by p_{ij} which represents the probability of choosing city j from city i, and is given by Eq. (3.3)

$$p_{ij} = \begin{cases} \dfrac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{h\in\Omega}[\tau_{ih}]^{\alpha}[\eta_{ih}]^{\beta}} & if \quad v_j \in \Omega \end{cases}$$
$$else \quad p_{ij} = 0 \tag{3.3}$$

Here, τ_{ij} is the pheromone concentration on path from *i* to *j*, $\Omega = v_j \in V : v_j$ feasible cities, α and β represent the biases for pheromone trail and visibility respectively, and they represent parameters to weigh pheromone concentration (which represents learnt knowledge for more global solution) with respect to visibility (which represents local heuristic desirability) in the transition rule.

The pheromone update rule is given by Eq. (3.4)

$$\tau_{ij}^{new} = \rho\tau_{ij}^{old} + \sum_{k=1}^{m}\Delta\tau_{ij}^{k} + \sigma\Delta\tau_{ij}^{*} \tag{3.4}$$

The details of individual terms of Eq. (3.4) are explained in the algorithm in the Table 2

The artificial ants construct vehicle routes by successively choosing the next cities using probabilistic transition rule shown in Eq. (3.3). If the probability *'p'* value for transition to a particular j^{th} city from the i^{th} city is more, then the chances of that city getting chosen as the next city is high. During every transition, the tour length is calculated and whenever the choice of next city would lead to an infeasible solution, i.e., if the tour length exceeds the vehicle distance constraint (the maximum distance that a vehicle can travel) *L*, then the depot is chosen as the next city (so that the tour of that vehicle is finished) and a new tour is started for a new vehicle. The vehicle capacity *L* is a critical parameter in finding out the min-max solution to the proposed problem. This parameter, as proven by Carlsson et al. in [18], is bounded between two values as per the following lemma:

Lemma 1: For a general planar graph representing the VRP,

$$\frac{TSP(D \cup N) - TSP(D)}{k} \le L \le \frac{TSP(N)}{k} + 2 * d(D, N) \tag{3.5}$$

In Eq. (3.5), N denotes the set of customer points, *D* denotes set of depots, *k* denotes number of vehicles in a depot, *d(A,B)* denotes the largest distance between an arbitrary pair of points in two different sets *A* and *B* as shown in Eq. (3.6), i.e.,

$$d(A,B) = \max_{x \in A, y \in B} \| x - y \| \tag{3.6}$$

A proof of the above lemma is provided in reference [18].

Since *L* represents the distance constraint, it is the maximum distance travelled by a vehicle. The approach here is based upon finding the least value of *L*, called *L**, that will still lead to a valid solution of the traditional SDVRP. This solution of traditional SDVRP will also be the solution of the min-max SDVRP. Thus this constraint on *L* acts as a critical parameter in finding out the in finding out the min-max solution. Solving the min-max SDVRP boils down to finding the minimum *L*, called *L**, that can still solve the traditional SDVRP using the existing vehicles. A high level description of the algorithm used to solve the min-max SDVRP is shown in Table 2. Explained in Table 2, is the proposed algorithm to obtain an approximate solution to min-max SDVRP. It may be mentioned that text between /* and */ represent the comments explaining the steps.

Table 2: Ant Colony Optimization Algorithm to Solve the Min-Max SDVRP

Ant Colony Optimization Algorithm to Solve the Min-Max SDVRP

/*Initialization*/

1. V = set of x,y coordinates of the cities, $V_0 = x,y$ coordinate of the depot
2. $L\_max = \dfrac{TSP(V)}{k} + 2 * d(V_0, V)$
3. $L\_min = \dfrac{TSP(V \cup V_0)}{k}$
4. **While** *(L>L_min)*

   a. **For** every edge *(i,j)* do

      $\tau_{ij}(0) = \tau_0$

      End For

/*Main Loop*/

b. **For** *t = 1* to *t_max* do

Initialize *Length_tour* and tour T^k

For *k = 1* to *m* do

Build tour T^k (t) by applying *n - 1* times the following step

Choose the next city *j* with probability, $$p_{ij} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum_{h\in\Omega}[\tau_{ih}]^{\alpha}[\eta_{ih}]^{\beta}} \quad \text{if } v_j \in \Omega$$
$$else \quad p_{ij} = 0$$

Calculate length of the tour *Length_tour*

If *Length_tour + d0(j) > L*

Choose depot as the next city

Calculate the *Length_tour* accordingly

End If

End For

Find minimum of *Length _tour* and tour *T*

If an improved tour is found **then**

Update it as *T_opt* and *L_opt*

End If

Count the number of vehic, *no_of_vehic* based on number of times the ant has visited depot in the tour *T_opt* and also find tour of each vehicle, *L_opt_vehic*

Using 2-opt Heuristic, if possible Obtain a better *T_opt* and *L_opt*

For every edge *(i,j)* do

Update pheromone trails by applying the rule: $$\tau_{ij}^{new} = \rho\tau_{ij}^{old} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} + \sigma\Delta\tau_{ij}^{*}$$

Where $$\Delta\tau_{ij}^{k} = \begin{cases} \dfrac{Q}{Length\_tour} & if\,(i,j) \in T \\ 0 & \text{otherwise} \end{cases}$$ and $$\Delta\tau_{ij}^{*} = \begin{cases} \dfrac{Q}{L\_opt} & if\,(i,j) \in T\_opt \\ 0 & \text{otherwise} \end{cases}$$

End For

End For

c. **If** *no_of_vehic* $\leq k$, where k is the number of vehicles in the depot

$L = L - \delta L$ *and* ***go to*** 'b' until *no_of_vehic* $= k+1$ and the respective L is L^*

Else $L = L + \delta L$

End If

End While

5. **Print** *T_opt, L_opt,* L^*

6 .**Stop**

$\alpha = 1, \beta = 5, \rho = 0.25, Q = 100, \tau_0 = 10^{-6}, \sigma = 5$

3.2 Extension of min-max SDVRP to min-max MDVRP

The approach used in this thesis to solve the min-max MDVRP is to decompose the problem into multiple SDVRPs and use the method proposed in 3.1 to solve those SDVRPs. The decomposition is carried out by dividing the region into equitable convex partitions. Once equitable partitions are found, solving min-max MDVRP reduces to solving min-max SDVRP for each partitioned region. However, it is necessary to achieve the partitioning in a way such that solving min-max SDVRP for partitions would correspond to solving the complete min-max MDVRP. In order to show that, first present a result proven by Beardwood et al. in [50] as a lemma below which can be used for arguing that, under certain assumptions, partitioning the polygon (convex polygon hull of the entire domain of customer points) equally would lead to an optimal partitioning for our problem.

Lemma 2: If X_i ,$1 \leq i \leq \infty$ are independently and identically distributed (i.i.d) random variables with bounded support in R^d, then the length L_n under the usual Euclidean metric of the shortest path through the points $\{X_1, X_2, \ldots. X_n\}$ satisfies

$$L_n \Big/ n^{\frac{(d-1)}{d}} \rightarrow \beta_{TSPd} \int_{R^d} f(x)^{\frac{(d-1)}{d}} dx \qquad (3.7)$$

In Eq. (3.7), $f(x)$ is the density of the absolutely continuous part of the distribution of the X_i, and $\beta_{TSP,d}$ is a positive constant that depends on d but not on the distribution of the X_i.

Based on the above lemma, we can deduce a proof for our 2-dimensional problem [51]. Assuming the cities are uniformly distributed and the function *Σf(x) dx* is the area of the partitioned subregion *A*, Ln for our problem is the route length that was calculated and it is apparent with a few substitutions *(d=2, n = 'k' no. of cities)*, that route length of each vehicle is asymptotically proportional to the square root of the area of subregion.

Mathematically it can be represented as shown in Eq. (3.8),

$$L_n \Big/ n^{\frac{1}{2}} \rightarrow \beta_{TSPd} A^{\frac{1}{2}} \tag{3.8}$$

Thus establishing an asymptotic relation between tour length and area, it is appropriate to divide the area into equitable partitions as the lengths of optimal tours in each partitions would approximately be the same. Because, for example, if an area of subregion *(A_1)* is larger than subregion *(A_2)*, then route length *(L_1)* of that area A_1 becomes longer and hence the time taken to cater to that area increases which should not be the case since it is a min-max problem. Hence, the most optimal heuristic is to divide the area into equal partitions.

3.2.1 Region Partitioning

A brief overview of the studies that have been carried out in finding equitable partitions of a region is presented in this section. This is pertinent as our solution implements these methods in generating a solution to the min-max MDVRP. In 1942, Stone and Tukey [52] proposed a basic partition theorem called Ham Sandwich Theorem (HST) which states that "given *n* measurable objects in n-dimensional space, it is possible to divide all of them in half with respect to their measure with a *(n-1)-dimensional* hyperplane". In a computational geometry perspective as proposed by [53], a discrete HST can be interpreted as follows. For a planar case of finite sets of red and blue points, there exists a line dividing both red and blue points into sets of equal size.

A generalized Ham Sandwich theorem was proven by Bespamyatnikh et al. in [54]. In this paper, they demonstrated that given *gn* red points and *gm* blue points in the plane in general position, there existed an equitable subdivision of the plane into *g* disjoint convex polygons each of which contains *n* red points and *m* blue points. The Ham Sandwich theorem is a subset of this problem given the specific case of *g=2*. Other notable partitioning techniques that has been discussed in [60] are a) Triangulation method, b) Hertel-Mehlorn(HM algorithm, c) Chazelle's complex cubic algorithm. Although these methods are relevant, the most relevant method to solve the proposed thesis was chosen from Carlsson's region partitioning method. On the basis of the generalized Ham Sandwich Theorem, Carlsson et al. developed an algorithm to find equitable convex partitions in a polygon [51]. This problem was addressed with a motivation to obtain a solution to the MDVRP. Region partition is a key step in this vehicle routing problem as the customers need to be allocated to vehicles in a load balancing manner. For a min-max type of a problem, since the travel times of the vehicle has to be minimum in order to obtain optimal solution, partitioning the polygon equitable was the primal factor.

Going back to our step of partitioning the given area, based on the technique used by the Carlsson in [51], we have the following important assumption: customer points in the *2-d* plane are uniformly distributed. A well-known partitioning method in the field of computational geometry is the Voronoi diagram. The method in particular is the Centroidal Voronoi Tessellation. Here, given points are separated based on the relative closeness/density and a centroidal point is calculated and a hyperplane is drawn by considering the fact that the set of points that are closer

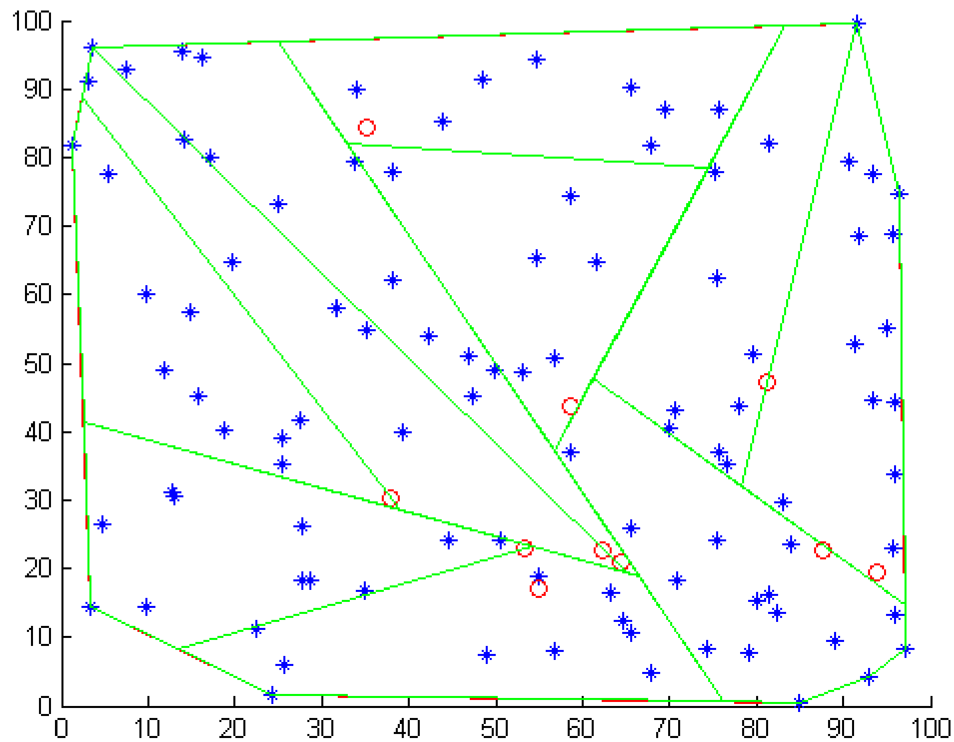


Figure 3-1: A convex equitable partition for a case of 10 depots

to centroid come under one voronoi cell. In our case these cells can be considered as subregions. But the drawback of this method is that it does not always divide the area into equal partitions. And if the partitions are not of same area, then we have already proved that it will be a suboptimal solution. Hence we implement a different method that was proposed by Carlsson et al. This method uses an approximation algorithm to find the location of partition by performing binary searches over the given set of depot points and customer points [18]. Based on this method any given polygon can be partitioned so that

a) All the partitions are convex polygons.

b) All the partitions contain exactly one depot

c) All the partitions have equal area

A sample figure representing a partitioned region that was obtained using the above method is shown in Figure 3-1.

3.2.2 Min-Max MDVRP Algorithm

Explained in Table 3 is the proposed algorithm to obtain an approximate solution to min-max MDVRP. It may be mentioned that text between /* and */ represent the comments explaining the steps. Also, the subroutine sdvrp(V,V0) is the subroutine representing the min-max SDVRP solution shown in Table 2.

Table 3: Ant Colony Optimization Algorithm to Solve the Min-Max MDVRP

Ant Colony Optimization Algorithm to Solve the Min-Max MDVRP
/*Inputs*/
1. *cust_points = get_custpoints()* ***/*set of x,y coordinates of the cities */***
2. *depot = get_depotpoints()* ***/*set of x,y coordinates of the depots*/***
/*Generate convex hull*/
3. *poly_points = [cust_points; depot]* ***/*set of combination of depot and cust_points*/***
4. *poly_vertices = convhull(poly_points)* ***/*get the vertices of the convex hull using convhull function*/***
/*Region Partitioning using Carlsson algorithm*/
5. *subregions = region_partition(poly_vertices,depot)* ***/*get the vertices of the partitioned polygon and also get the***

```
depot points corresponding to these set of subregion vertices*/

/*Main Loop*/

no_of_depots = size(depot)

6. For i = 1 to no_of_depots do

      IN = inpolygon() /*assign the cust_points that are in partition i to depot i, basically check the points if it
      inside or outside the polygon which is in focus*/

      V = cust_points(IN)  /*V is the customer points that are inside region 'i'*/

      V0 = depot(i)  /*depot(i) corresponds to the x,y coordinate of the ith depot*/

/*Generate the min-max SDVRP tour and calculate optimum tour length and optimum vehicle tour for each
vehicle*/

[opt_vehic_tour opt_tour_length] =  sdvrp(V,V0)

   End For

/*Plot the Vehicle Tours*/

7. For i = 1 to no_of_depots do

      For k=1 to no_of_vehic do

         plot() /*Plot the vehicle tour*/

      End For

   End For
```

3.3 Tour Improvement Strategies

3.3.1 Candidate List

Candidate list technique was first shown by Bullnheimer et al. in [55]. In this technique, a list of cities in the increasing order of distance from the current city will be assigned to each city. Only the closest cities will be included in the candidate list and are made available for selection of the next city to be visited in the route.

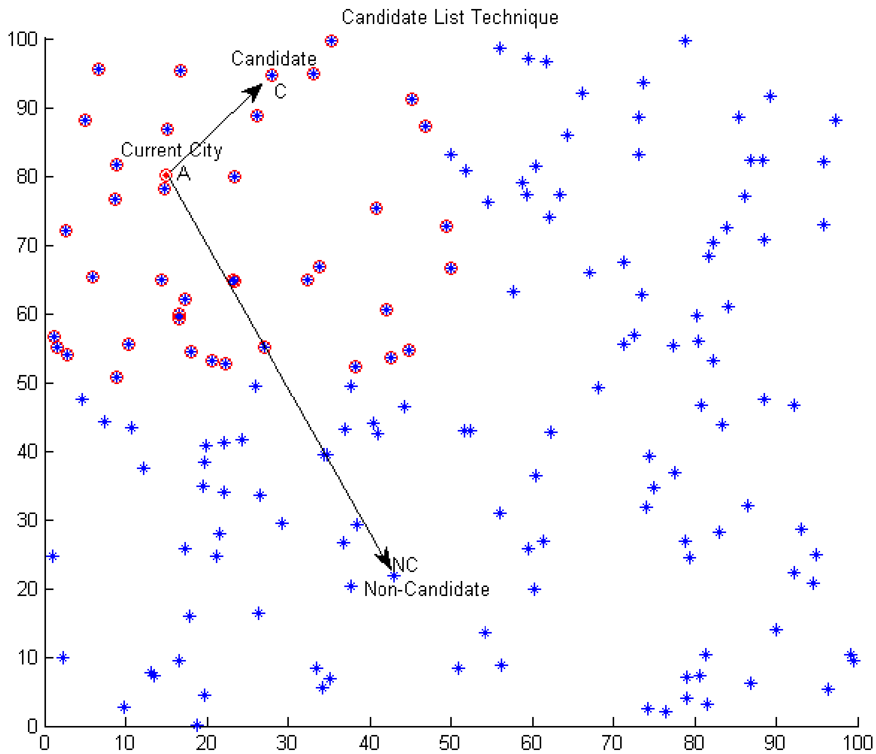


Figure 3-2: Candidate List Technique

Ant in City ‘A’ can choose City ‘C’ which is in City ‘A’s candidate list, while ‘NC’ is not in candidate list

An example is provided in Figure 3-2, where the candidate cities are marked with red circles for a current city A. Hence an ant in the city ‘A’ can’t choose city ‘NC’ which is far away from the city ‘A’ unlike city ‘C’. This is a logical assumption as it is most unlikely that an ant will choose city ‘NC’ in building the route. Usually, the candidate list size will be set to [n/4] where ‘n’ is the total number of customers. This application of candidate list route improvement technique has two advantages. First, the calculation time is less because the probability calculation for all the cities is not necessary. Second, the solution quality is improved because selection probabilities are no longer undermined by highly unlikely customers. In [48], Bell et al. experimented with different candidate list sizes and study the impact of the selection size with respect to the problem size.

3.3.2 *r*-opt heuristic

This tour improvement strategy was shown by Lin et al. in [56]. They implemented a local exchange procedure after a locally optimum tour is found. This technique is commonly called r-opt heuristic and 'r' can be 2, 3... In this thesis, 2-opt heuristic is used wherein all possible pairwise exchange of cities are made in order to find any better solution.

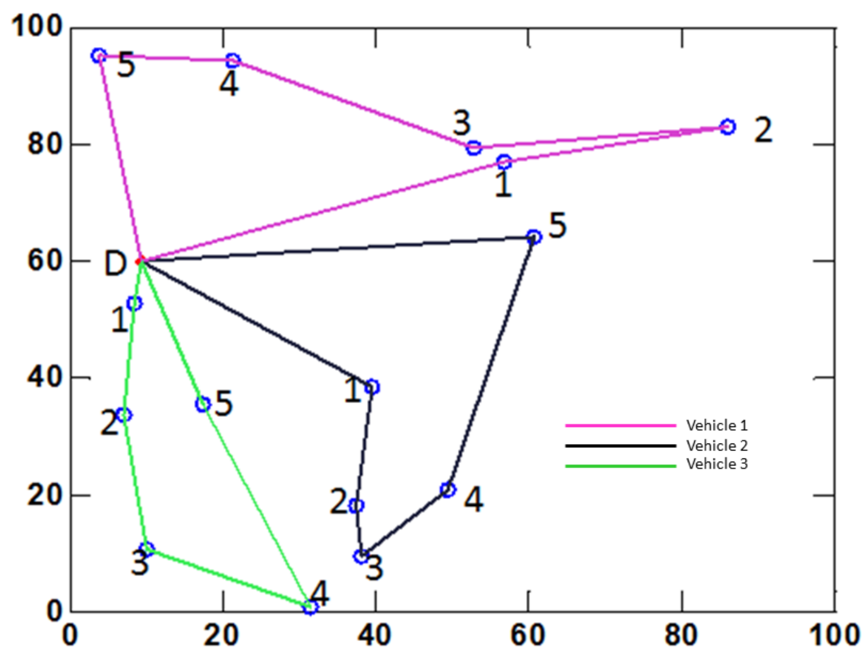


Figure 3-3: Vehicle routes in a typical VRP solution and sequential local exchange procedure

For example, consider Figure 3-3 and say for 1st vehicle the optimal tour obtained using ant colony optimization technique is D-1-2-3-4-5. This 2-opt heuristic technique performs a sequential exchange of the cities and calculate the distances for all the pairwise permutations like for the set D-1-2-3, distances for D-1-3-2, D-2-3-1 and D-3-2-1 are calculated and if an overall improvement of tour length is obtained, the respective tour is used as an optimal tour. This adds greatly to the number of combinations that are explored by ant colony search technique and can be used in finding the best solution before applying the pheromone update rule.

Chapter 4. Results and discussions

A MATLAB based program was developed to implement the proposed solution for min-max variant of SDVRP and MDVRP. The proposed algorithm was verified using a number of simulations. In this chapter, results are presented starting from min-max SDVRP solution and extending to min-max MDVRP solution. To validate the results obtained, a comparison with the existing method proposed by Carlsson et al [18] has been carried out. A MATLAB program that implements the Carlsson's Linear Programming based algorithm is freely available over the internet [57]. Going forward, this Carlsson's Linear Programming based method is acronymically mentioned as LP based method. It may be noted that their program does not work for 2 vehicle problems, and hence the results provided here do not include their results for 2 vehicle problems.

4.1 Min-Max SDVRP Results

To demonstrate the performance of our algorithm, we present results from three simulated scenarios: i) Scenario 1 with 15 cities; ii) Scenario 2 with18 cities; and iii) Scenario 3 with 25 cities. In each of these scenarios, the number of vehicles varies from 1 to 4/5. The second set of scenarios has no. of vehicles fixed: (iv) Scenario 4 with 150 cities and 7 vehicles; and (v) Scenario 5 with 300 cities and 13 vehicles.

Figure 4-1 shows the solution obtained for scenario (i) and the comparison in terms of distance with LP based method has been made in Table 4. Figure 4-1a was obtained when the vehicle distance constraint (the distance it can travel) was *L_max*. This particular case is equivalent to TSP because there is only one vehicle involved with a capacity more than TSP. The same can be observed through the results obtained. The distance travelled by that vehicle is the TSP distance which is *323.59*. Figure 4-1b was obtained when the number of vehicles *(no_of_vehic)* was set to *2*. Here the *L** was found to be *230*. In this case, vehicle 1 travels *173* units of distance, while vehicle *2* travels *205* units of distance with the total distance travelled being *378*. Figure 4-1c was obtained when the number of vehicles was set to *3*. In this case, *L** obtained was *180*. The total distance travelled, however, increases to *427* from *378*. Here, vehicle *1* travels *173* units, vehicle *2* travels *164* units, and vehicle *3* travels *89* units. Figure 4-1d was obtained when the number of vehicles was set to *4*. *L** obtained in this case was *170* with vehicle *1* travelling *154* units, vehicle *2* travelling *89* units, vehicle *3* travelling *164* units, and vehicle *4* travelling *112* units.

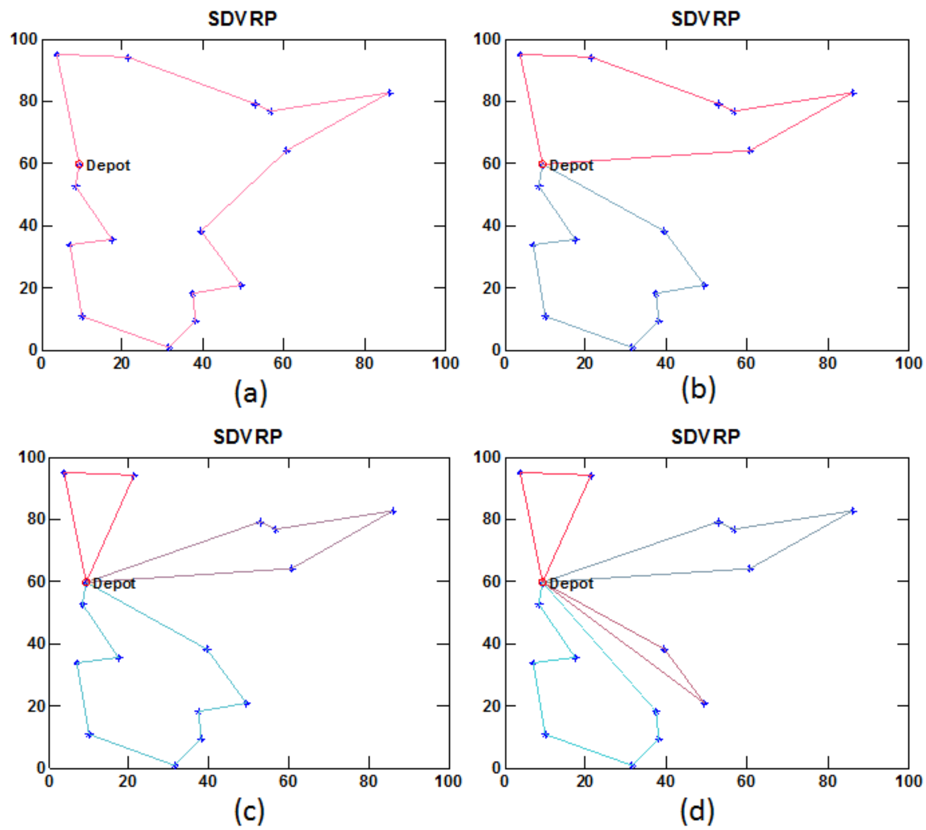


Figure 4-1: Results obtained using the proposed ant colony based approach for Scenario 1 (1 Depot 15 City problem)

Figure 4-2 show the corresponding results obtained from LP based method. Figure 4-2a shows the *1* vehicle case and it can be observed that it is same as Figure 4-2a which is nothing but the TSP solution and the total distance being TSP length which is *323.59* units. Figure 4-2b shows the *3* vehicles case where the lengths traversed by vehicles are: vehicle *1*: *178* units, vehicle *2*: *201* units, and vehicle *3*: *136* units Hence, total length is *515* and maximum length being *201* which is of vehicle *2*. Figure 4-2c shows the *4* vehicles case. The lengths traversed in this case are vehicle *1*: *143* units, vehicle *2*: *211* units, vehicle *3*: *101* units and vehicle *4*: *133* units. Thereby, total length is equal to *588* units and maximum length is *211* units. Table 4 compares the results obtained from the proposed ant colony based method with respect to the LP based method.

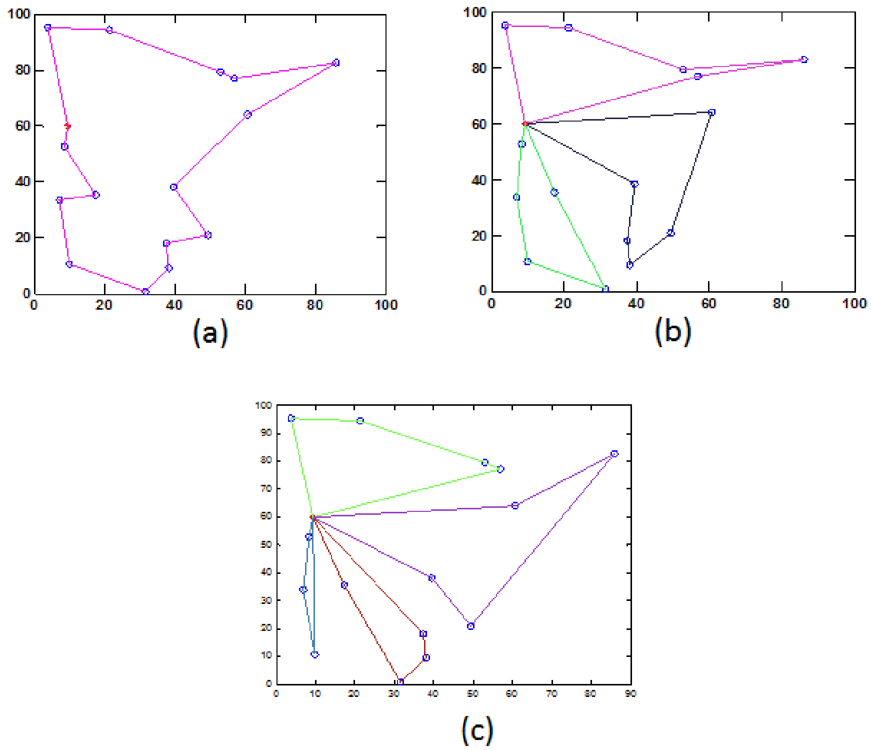


Figure 4-2: Results obtained using LP based approach developed by Carlsson et al. in case of Scenario 1 (1 Depot 15 Customer problem)

Table 4: Comparison of Ant Colont Based and LP Based technique to solve min-max SDVRP for Scenario 1

No. of Vehicles	Distance Travelled (Ant Colony Based)	Distance Travelled (LP Based Method)
1	Vehic 1 = 323.59*	Vehic 1 = 323.59
2	Vehic 1 = 173	
	Vehic 2 = 205**	
3	Vehic 1 = 173**	Vehic 1 = 178
	Vehic 2 = 164	Vehic 2 = 201**
	Vehic 3 = 89	Vehic 3 = 136
4	Vehic 1 = 154	Vehic 1 = 143
	Vehic 2 = 89	Vehic 2 = 211*
	Vehic 3 = 164**	Vehic 3 = 101
	Vehic 4 = 112	Vehic 4 = 133

* denotes the TSP solution for the problem
** denotes the maximum distance travelled in a set of vehicle routes

Table 5: Comparison of Ant Colont Based and LP Based technique to solve min-max SDVRP for Scenario 2

No. of Vehicles	Distance Travelled (Ant Colony Based)	Distance Travelled (LP Based Method)
1	Vehic 1 = 340.14*	Vehic 1 = 340.14
2	Vehic 1 = 258**	
	Vehic 2 = 189	
3	Vehic 1 = 135	Vehic 1 = 184
	Vehic 2 = 219**	Vehic 2 = 284
	Vehic 3 = 197	Vehic 3 = 229**
4	Vehic 1 = 135	Vehic 1 = 168
	Vehic 2 = 189	Vehic 2 = 181
	Vehic 3 = 183	Vehic 3 = 213**
	Vehic 4 = 197**	Vehic 4 = 198

* denotes the TSP solution for the problem
** denotes the maximum distance travelled in a set of vehicle routes

Table 5 shows the comparison of results obtained for scenario (ii). The plot is not shown in order to avoid repetition of similar type of figures. It is observed that in each set of tours starting from 1 vehicle (nothing but a *TSP*) to 4 vehicle tour, maximum distance travelled by a vehicle in case of Ant colony optimization method is lower than the LP based method.

For scenarios (iii), a similar simulation exercise has been done and the results obtained are shown in Figure 4-3 and Figure 4-4 and the comparison carried out is also tabulated in Table 6.

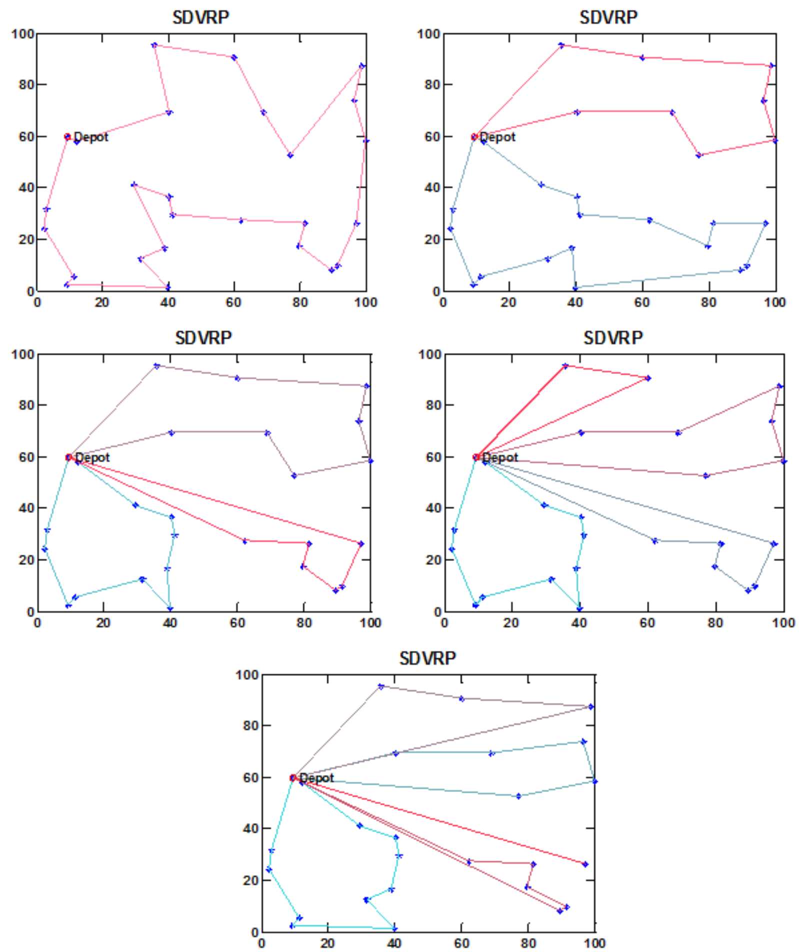


Figure 4-3: Results obtained using the proposed ant colony based approach for Scenario 3 (1 Depot 25 City problem)

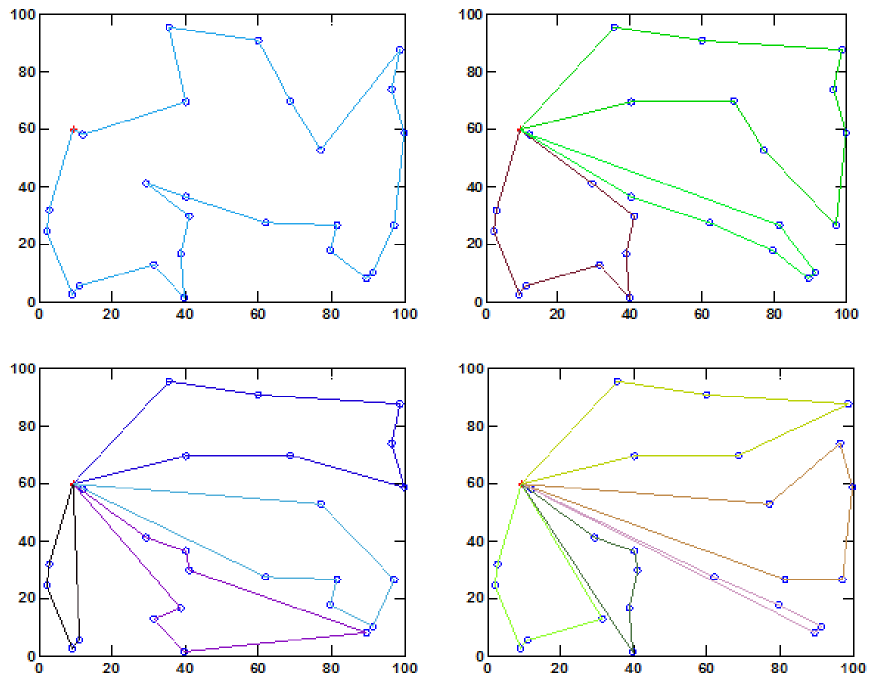


Figure 4-4: Results obtained using LP based approach for Scenario 3 (1 Depot 25 City problem)

Table 6: Comparison of Ant Colony Based and LP Based technique to solve min-max SDVRP for Scenario 3

No. of Vehicles	Distance Travelled (Ant Colony Based)	Distance Travelled (LP Based)
1	Vehic 1 = 471.17*	Vehic 1 = 470.89*
2	Vehic 1 = 290**	
	Vehic 2 = 240	
3	Vehic 1 = 173	Vehic 1 = 171
	Vehic 2 = 240**	Vehic 2 = 282**
	Vehic 3 = 217	Vehic 3 = 197
4	Vehic 1 = 173	Vehic 1 = 231
	Vehic 2 = 217**	Vehic 2 = 224**
	Vehic 3 = 216	Vehic 3 = 222
	Vehic 4 = 128	Vehic 4 = 117
5	Vehic 1 = 173	Vehic 1 = 136
	Vehic 2 = 196	Vehic 2 = 239**
	Vehic 3 = 201	Vehic 3 = 194
	Vehic 4 = 202**	Vehic 4 = 203
	Vehic 5 = 187	Vehic 5 = 140

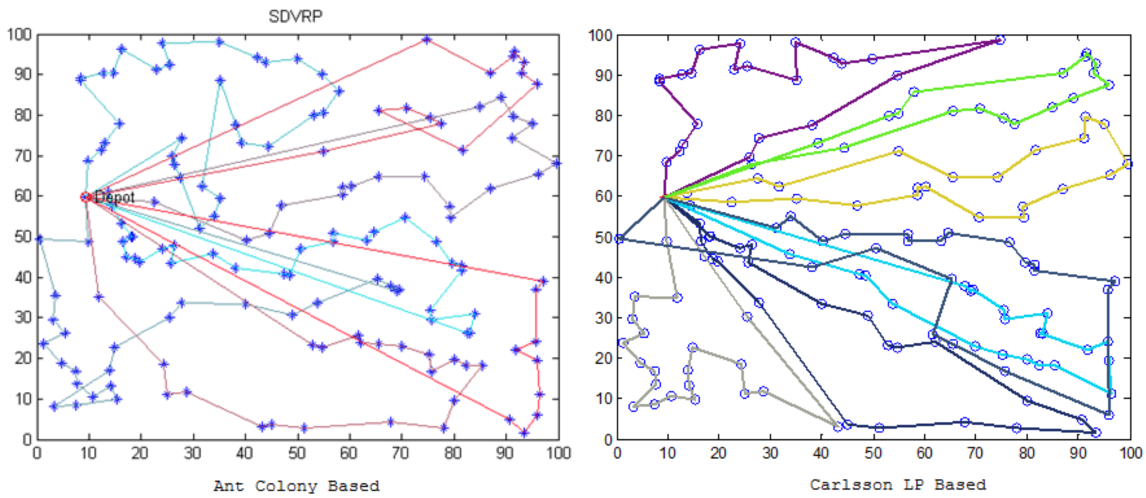


Figure 4-5: Comparison of Results obtained for Ant Colony Based and LP Based technique for Scenario 5 (7 Vehicles – 150 Cities)

Table 7: Comparison of Vehicle Distances Ant Colony Based and LP Based technique for Scenario 4

Distance travelled by each vehicle (Ant Colony Based)	Distance travelled by each vehicle (LP Based)
Vehic 1 = 219.75	Vehic 1 = 226.82
Vehic 2 = 229.00	Vehic 2 = 209.97
Vehic 3 = 232.20	Vehic 3 = 217.36
Vehic 4 = 220.68	Vehic 4 = 197.77
Vehic 5 = 222.36	Vehic 5 = 269.93**
Vehic 6 = 231.88	Vehic 6 = 202.36
Vehic 7 = 235.81**	Vehic 7 = 196.87

** denotes the maximum distance travelled in a set of vehicle routes

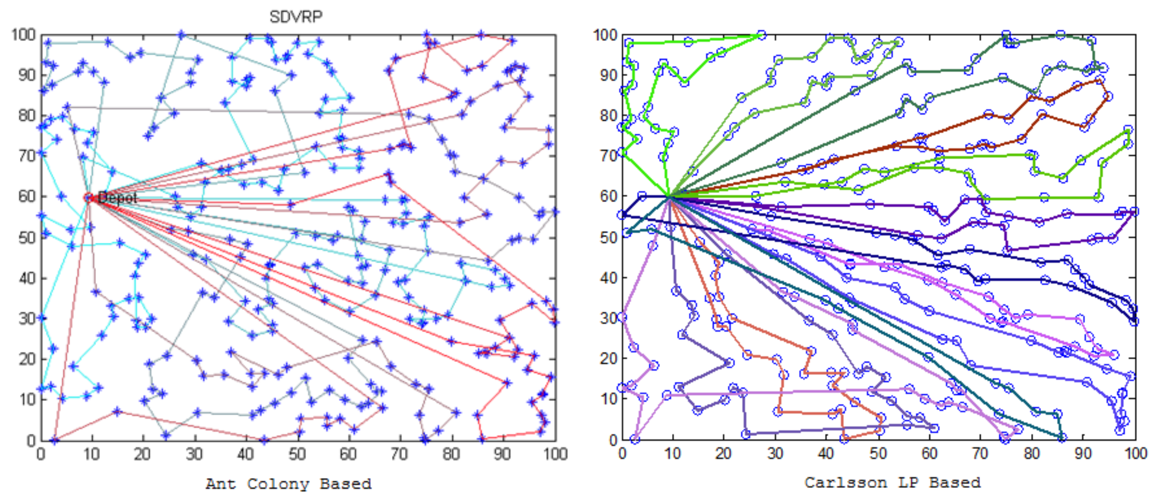


Figure 4-6: Comparison of Results obtained for Ant Colony Based and LP Based technique for Scenario 5 (13 Vehicles – 300 Cities)

Table 8: Comparison of Vehicle Distances Ant Colony Based and LP Based technique for Scenario 4

Distance travelled by each vehicle (Ant Colony Based)	Distance travelled by each vehicle (LP Based)
Vehic 1 = 220.03	Vehic 1 = 135.44
Vehic 2 = 229.26	Vehic 2 = 139.82
Vehic 3 = 239.90**	Vehic 3 = 173.13
Vehic 4 = 236.38	Vehic 4 = 199.97
Vehic 5 = 228.82	Vehic 5 = 196.41
Vehic 6 = 212.26	Vehic 6 = 208.86
Vehic 7 = 232.30	Vehic 7 = 192.78
Vehic 8 = 235.73	Vehic 8 = 226.17
Vehic 9 = 228.43	Vehic 9 = 208.88
Vehic 10 = 228.99	Vehic 10 = 208.10
Vehic 11 = 231.57	Vehic 11 = 246.47**
Vehic 12 = 231.53	Vehic 12 = 219.66
Vehic 13 = 226.17	Vehic 13 = 211.91

** denotes the maximum distance travelled in a set of vehicle routes

Figure 4-5 represents the scenario (iv) and the corresponding comparison that has been carried out in terms of distance travelled by each vehicle is shown in Table 7. It can be observed that the maximum distance travelled in ant colony based algorithm(235.81) is lesser than the LP based algorithm(269.93) and thus strengthening the fact that ant colony optimization fairs well even for large number of cities in terms of accuracy of the solution. A similar observation extends to scenario (v) which is shown in Figure 4-6 and tabulated in Table 8.

To further validate the repeatability of the optimal results, a number of simulations were performed for a typical SDVRP problem of 1 Depot - 3 Vehicle - 30 City scenario and results were compared between ant colony based approach with LP based approach. Figure 4-7 shows the result obtained using Carlsson LP based method. Figure 4-8 represents a sample result obtained using Ant Colony based approach. A total of 40 simulation runs were performed to validate the correctness and repeatability of the solution obtained through the proposed ant colony based algorithm. Comparison in terms of this maximum distance travelled by a vehicle in a simulation run has been made in Figure 4-9 between ant colony based and LP based techniques. It is observed that the curve obtained from Ant colony based method (shown as *dashed red line*) is always lower that the curve obtained using LP based method (shown as *continuous blue line*). This establishes that Ant Colony based method consistently performs well in terms of the optimality of the results.

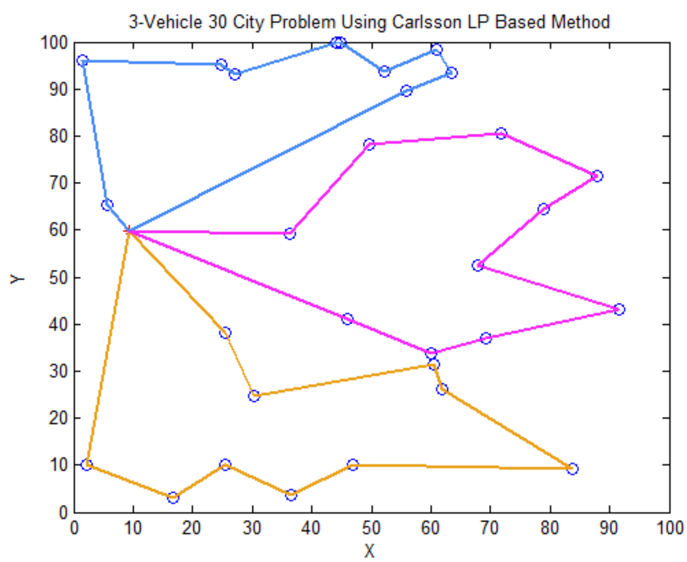


Figure 4-7: Result Obtained for 3-Vehicle 30 City Problem using LP Based Method

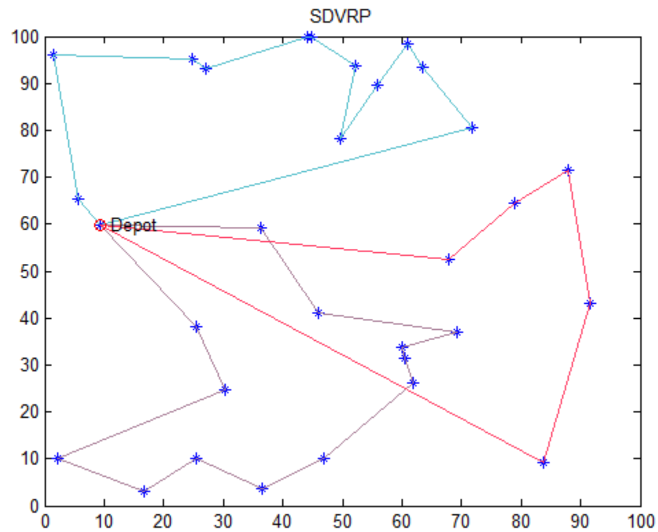


Figure 4-8: A Sample Result* obtained for the same 3-vehicle 30 city scenario using Ant Colony Based approach during the Simulation Exercise

*Result May not be Optimal.

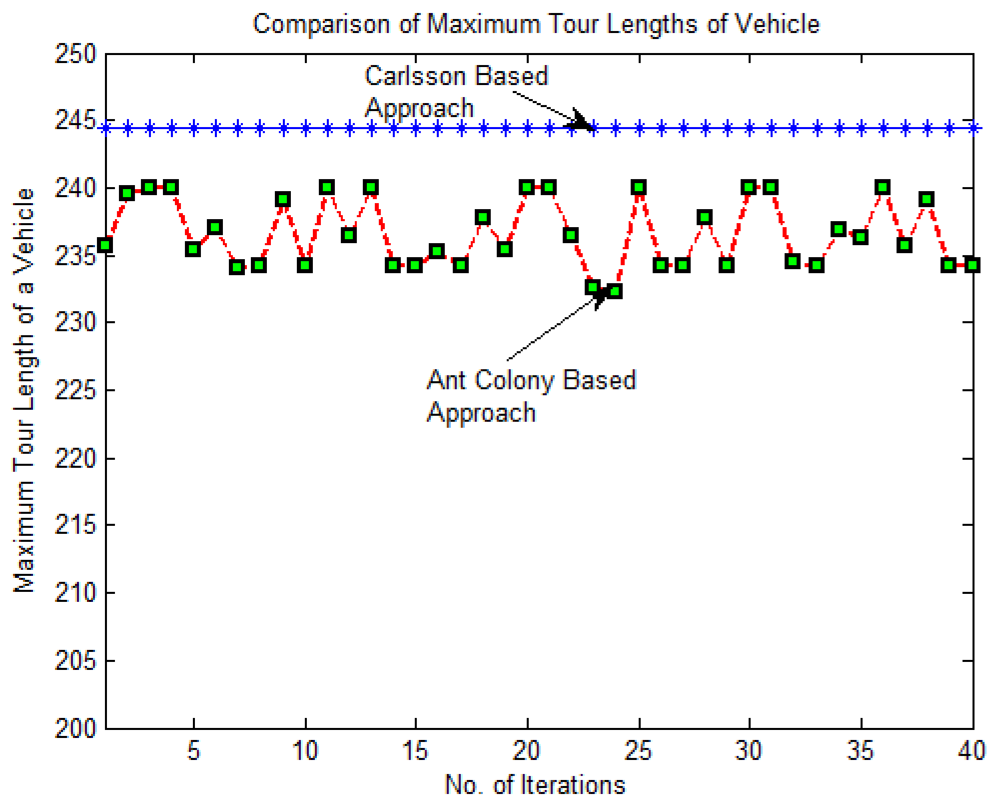


Figure 4-9: Comparison of Maximum Tour Lengths of Vehicle for a 3 Vehicle, 30 City Problem

4.2 Min-Max MDVRP Results

Min-Max SDVRP was extended to Min-Max MDVRP as explained in the section 3.2. Before we discuss the results, let us capture some of the assumptions that are made: a) Customer points and depot points are uniformly distributed over *2-d* space b) the vehicle capacities remain the same throughout all depots. This thesis presents results for three scenarios: i) Scenario 1 with 4 depots and 80 cities; ii) Scenario 2 with 4 depots and 140 cities; iii) Scenario 3 with 5 depots and 140 cities

Figure 4-10 compares the solution for scenario (i) between Carlsson based method and the Ant Colony based method. Each depot has two vehicles.

Table 9 compares the distances traversed by each vehicle for the scenario. The maximum vehicle distance in ant colony based method is by vehicle 1 in depot 3 which is *125.35* units. Comparing with the Carlsson's LP-based method where the maximum distance is *131.47* units (vehicle 2 in 2nd depot), it is clear that ant colony based method improves on LP-based method in this case. Two more cases demonstrating improved performance of the ant colony solution over the Carlsson's algorithm in terms of optimality of solution are shown with the results tabulated in Table 10 and Table 11. Figure 4-11and Figure 4-12 represent these cases respectively.

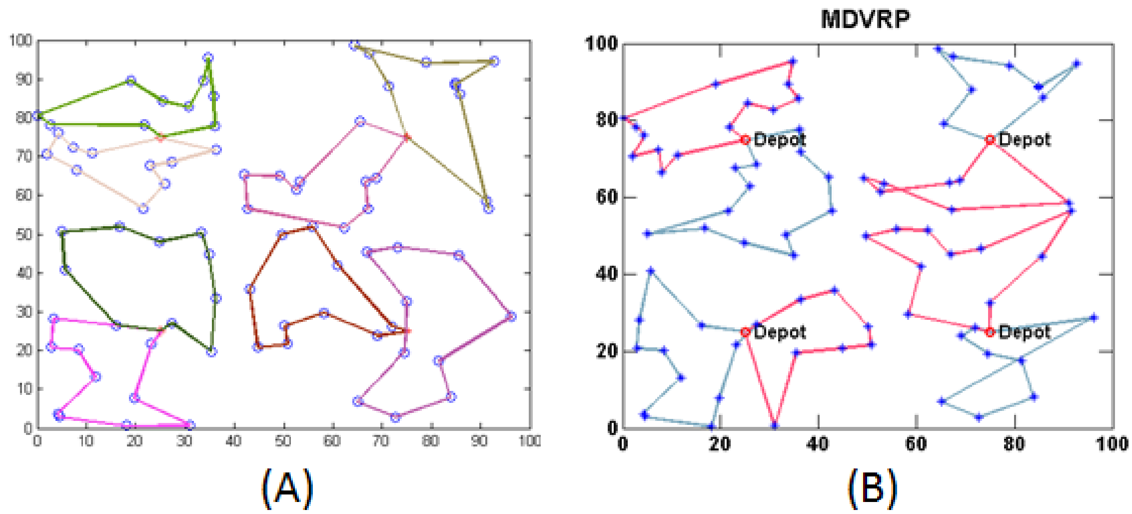


Figure 4-10: For the case of Scenario 1 (80 Customers and 4 depots)

(A) Results obtained using LP based approach (B) Results obtained using Ant Colony based approach

Table 9: Comparison of Ant Colony based and LP based technique to solve min-max MDVRP for Scenario 1

Depot No.	Distance Travelled by Each Vehicle*(L_opt_vehic)* (Ant Colony based method)	Distance Travelled by Each Vehicle (LP based method)
1	Vehic 1 = 111.60	Vehic 1 = 114.05
	Vehic 2 = 129.20*	Vehic 2 = 110.96
2	Vehic 1 = 94.21	Vehic 1 = 108.48
	Vehic 2 = 100.39	Vehic 2 = 131.47*
3	Vehic 1 = 125.35	Vehic 1 = 104.38
	Vehic 2 = 114.59	Vehic 2 = 92.37
4	Vehic 1 = 118.97	Vehic 1 = 102
	Vehic 2 = 97.71	Vehic 2 = 124.65

* denotes the maximum distance travelled in a set of vehicle routes

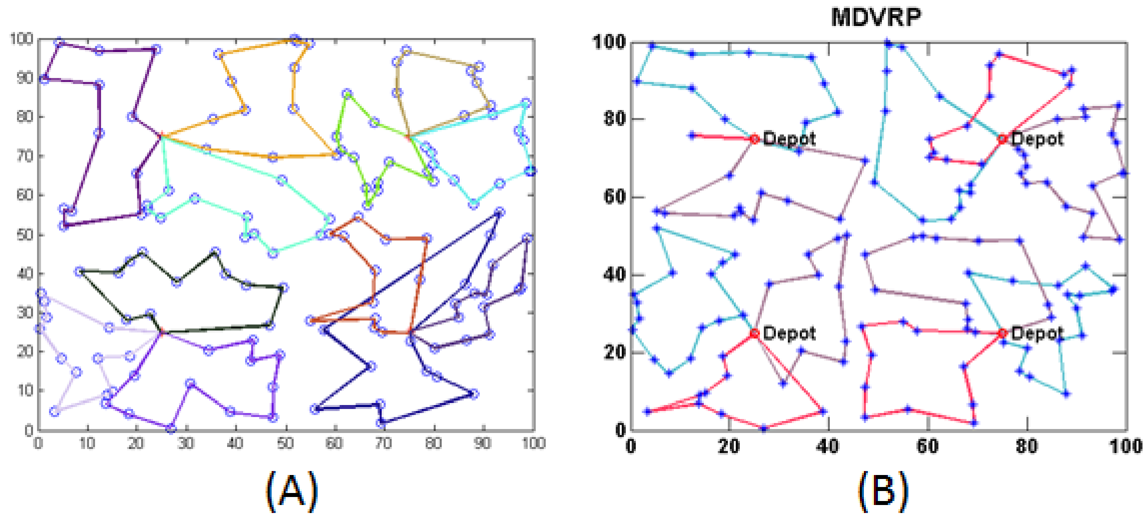


Figure 4-11: For the case of Scenario 2 (140 customers and 4 depots).

(A) Results obtained using LP based approach (B) Results obtained using Ant Colony based approach

Table 10: Comparison of Ant Colony based and LP based technique to solve min-max MDVRP for Scenario 2

Depot No.	Distance Travelled by Each Vehicle*(L_opt_vehic)* (Ant Colony based method)	Distance Travelled by Each Vehicle (LP based method)
1	Vehic 1 = 110.97	Vehic 1 = 97.91
	Vehic 2 = 114	Vehic 2 = 111.65
	Vehic 3 = 25.24	Vehic 3 = 114.39
2	Vehic 1 = 114.57	Vehic 1 = 171.40*
	Vehic 2 = 103.83	Vehic 2 = 105.09
	Vehic 3 = 95.11	Vehic 3 = 85.85
3	Vehic 1 = 115.94	Vehic 1 = 120.27
	Vehic 2 = 105.66	Vehic 2 = 142.12
	Vehic 3 = 102.22	Vehic 3 = 113.29
4	Vehic 1 = 116.28*	Vehic 1 = 82.57
	Vehic 2 = 112.56	Vehic 2 = 79.89
	Vehic 3 = 91.21	Vehic 3 = 67.70

* denotes the maximum distance travelled in a set of vehicle routes

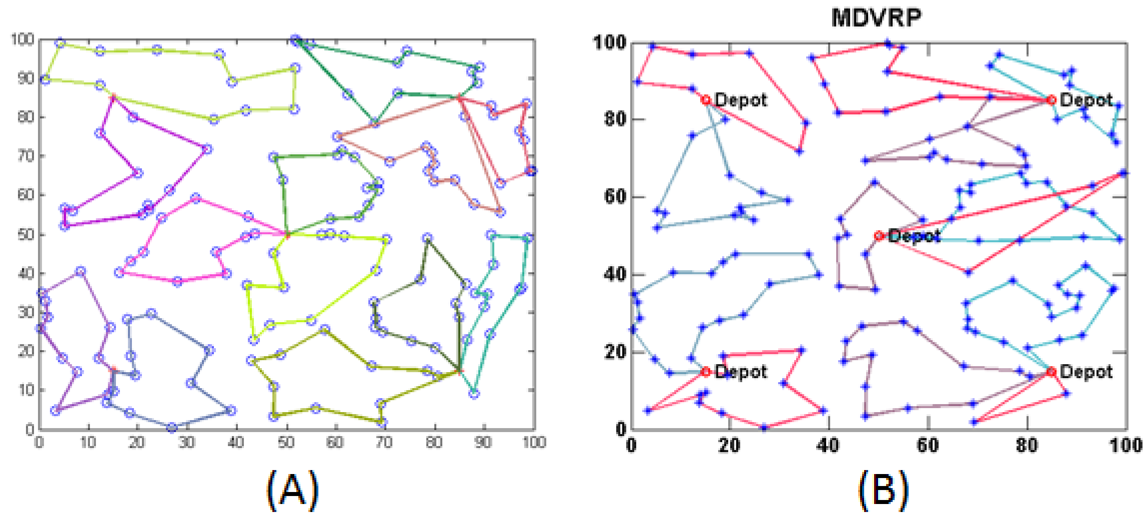


Figure 4-12: For the case of Scenario 3 (140 customers and 5 depots).

(A) Results obtained using LP based approach (B) Results obtained using Ant Colony based approach

Table 11: Comparison of Ant Colony based and LP based technique to solve min-max MDVRP for Scenario 3

Depot No.	Distance Travelled by Each Vehicle*(L_opt_vehic)* (Ant Colony based method)	Distance Travelled by Each Vehicle (LP based method)
1	Vehic 1 = 108.15	Vehic 1 = 89
	Vehic 2 = 96.66	Vehic 2 = 94.48
2	Vehic 1 = 117.17	Vehic 1 = 89.41
	Vehic 2 = 104.08	Vehic 2 = 96.95
3	Vehic 1 = 114.02	Vehic 1 = 73.46
	Vehic 2 = 116.20	Vehic 2 = 94.47
	Vehic 3 = 46.99	Vehic 3 = 110.20
4	Vehic 1 = 118.03	Vehic 1 = 85.84
	Vehic 2 = 76.18	Vehic 2 = 126.05*
	Vehic 3 = 113.65	Vehic 3 = 106.80
5	Vehic 1 = 81.90	Vehic 1 = 99.15
	Vehic 2 = 112.65	Vehic 2 = 104.73
	Vehic 3 = 118.96*	Vehic 3 = 65.25

* denotes the maximum distance travelled in a set of vehicle routes

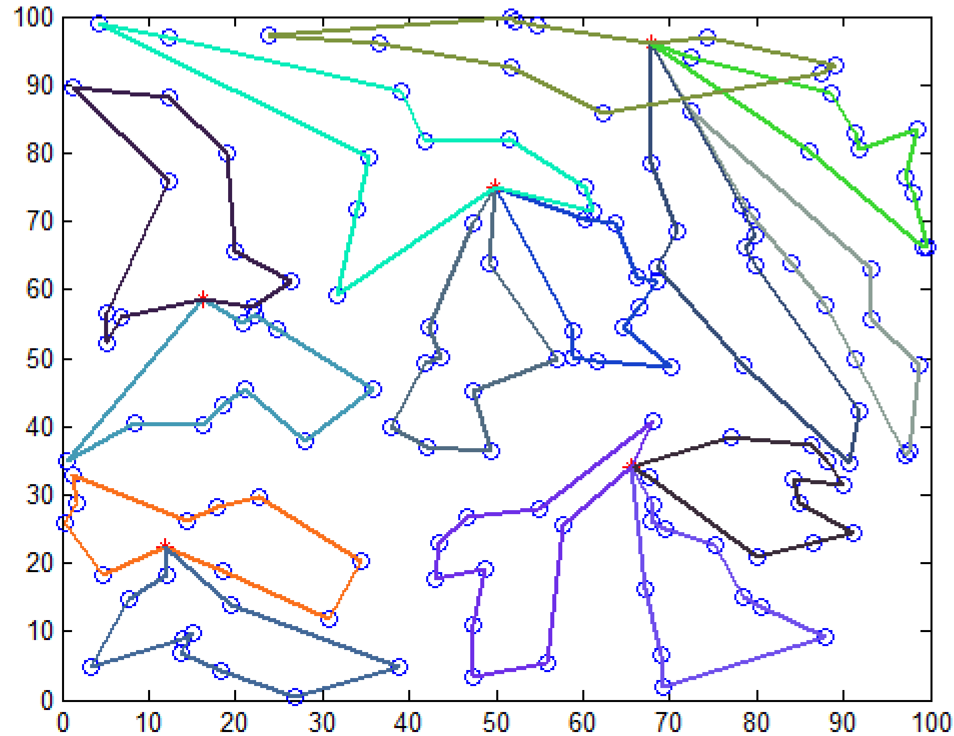


Figure 4-13: Result Obtained 5 Depot 16-Vehicle 140 City Problem using LP Based Method

Similar to SDVRP, an extensive simulation exercise has been carried out to validate the repeatability of the solution obtained through ant colony optimization technique. A sample 5 depot - 16 vehicles - 140 city problem was chosen as a test bed to compare the results. A sample result obtained using Carlsson LP based technique is shown in Figure 4-13. The Ant Colony based approach was used for 40 simulation runs. A sample result is also shown in Figure 4-14. A comparison was made in terms of the maximum distance travelled by any vehicle in a set of vehicle tours and was plotted in Figure 4-15. Continuous blue line represents the longest tour obtained using the LP method, while the red dashed line represents the longest tour of a vehicle using the ant colony based method. From the figure, it is deduced that ant colony based technique fairs well consistently in comparison with Carlssson LP based technique in terms of optimality of the solution. Finally, a plot showing 300 simulation runs is shown in Figure 4-16. The difference between Figure 4-15 and Figure 4-16 being the former is a plot obtained for a constant set of inputs while the latter is generated by different set of inputs. Inputs here is the depot and customer locations given by x,y coordinates.

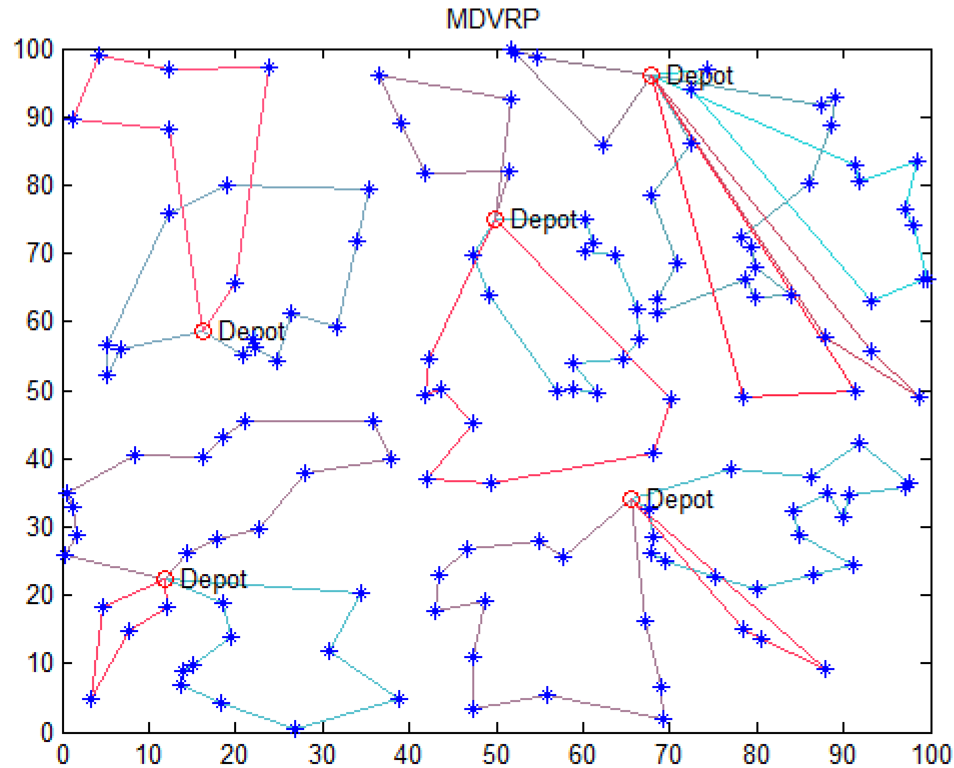


Figure 4-14: A Sample Result* obtained for the same 5 depot 16-vehicle 140 city scenario using Ant Colony Based approach during the Simulation Exercise

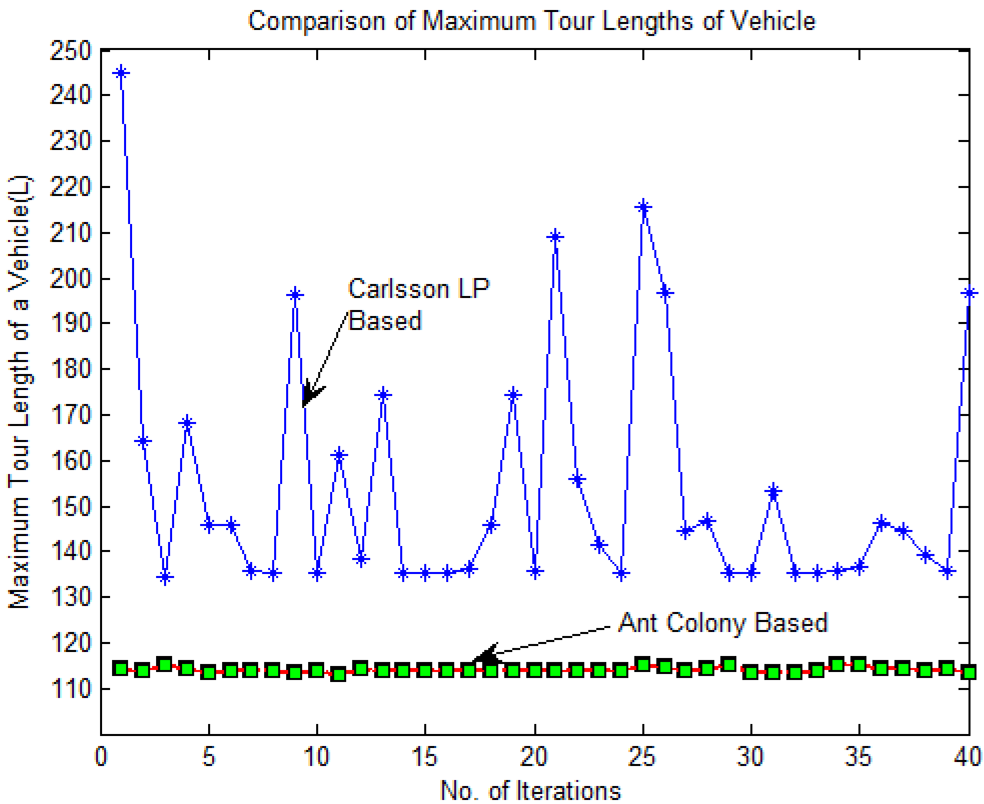


Figure 4-15: Comparison of Maximum Tour Lengths of Vehicle for a 5 Depot-16 vehicle-140 City Problem

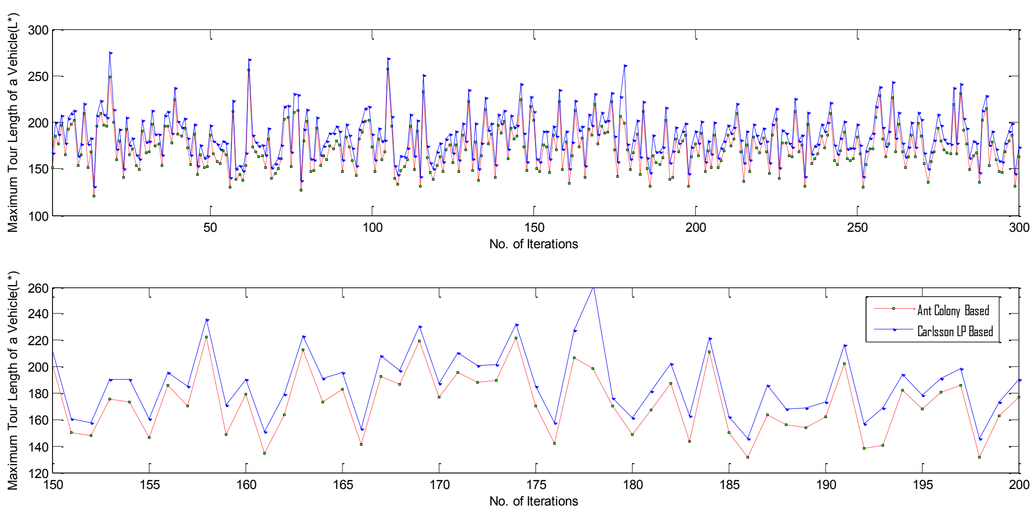


Figure 4-16: Comparison of Maximum Tour Lengths of Vehicle for a 140 City Problem

Some of the observations that can be made from the results are as follows. It is observed that the vehicle lengths in the ant colony approach are almost uniform (few exceptions) validating the claim made using Lemma 2 in section

3.2 that equal partitioning would lead to equitable distributions of cities and hence the lengths of the tours of cities. Since the area has been partitioned equally, these lengths are also distributed equally. Getting a guaranteed optimal solution to this problem is known to be NP hard, however approximate methods, such as proposed here, can improve significantly on the time required to calculate a solution. It may be noted that the proposed approach makes an assumption regarding the uniform distribution of cities. In practice, the cities may not be distributed uniformly, in which case the results may not be as optimal. However, these sub-optimal solutions can be used with other optimization methods to obtain a near-optimal solution. Moreover, this approximation will get better as the total number of cities increases. Thus, this is a good initial guess for the solution, but for smaller sets of cities there will probably be a need to exchange cities between tours after this initial partitioning is done. A drawback of this method is the calculation time required to achieve the solution. For the cases from Figure 4-10 - Figure 4-12, the approximate time taken to achieve the solution is 200 – 250 seconds. On the other hand, Carlsson method was able to solve in few seconds. This is because the method used by Carlsson et al. utilizes the well-known Concorde TSP solver [58] which can get to near-exact TSP solution within a few seconds. One way to improve the running time is to use the inherent distributed nature of the problem once the area is partitioned, and to perform the calculation of the each SDVRP on a separate processor.

Chapter 5. Conclusion and future work

The paper presents an ant colony optimization based algorithm to solve an interesting class of problem called min-max MDVRP. Unlike a traditional MDVRP, which minimizes the total distance travelled, the min-max MDVRP minimizes the maximal distance travelled by a vehicle. This version of problem holds immense applications for time-critical problems. The proposed approach makes use of the distance constraint in traditional MDVRP to find its optimal value. This optimal distance constraint, when used in a traditional MDVRP, minimizes the maximal distance travelled by a vehicle. Verification of the proposed technique has been carried with the help of extensive simulations, and results have been compared with the results obtained using a known LP based algorithm. It has been established that the results obtained using the proposed ant colony based approach provides more optimal results although takes more calculation time as compared to the LP based algorithm. A brief introduction to computational complexity has been provided in 2.7 and the idea here is to help the practitioners to achieve gain insights in the complex behaviors of such algorithms which will enable in developing better algorithms in future compared to existing ones. Thus the future scope of this thesis could be to understand the complexity of the problem and bring out a compare it with existing methods. Implementation of some of the tour improvement heuristics like 2-opt or 3-opt with the candidate list method have been carried out in order to expedite convergence towards the solution. The min-max MDVRP problem was subdivided into many min-max SDVRPs by partitioning equally the region consisting of depots and cities. The partition of the polygon was carried out with an existing method developed by Carlsson. Some of the established convex partitioning techniques have also been listed and future work here is to explore better ways of achieving optimal convex partition. The technique has been validated with theory and with extensive simulations, and results have been compared with the results obtained using an existing method developed by Carlsson. Future work includes using parallel processing programming techniques to expedite the convergence towards the solution and integrating this method with other optimization techniques to achieve more optimal solutions in cases where we have non-uniform distribution or smaller number of cities.

References

[1] Cesar, H., Oliviera B., Vasconcelos, G.C., "A hybrid search method for the vehicle routing problem with time windows", *Annals of Operations Research*, 2008.

[2] Toth,P., Vigo, D., "The Vehicle Routing Problem", *Monographs on Discrete Mathematics and Applications*, SIAM, Philadelphia, 2001.

[3] US Department of Labor, "Occupational Outlook Handbook", *Bureau of Labor Statistics*, 2010-11.

[4] Dantzig, G.B., Ramser J.H., "The truck dispatching problem", *Management Science*, vol. 6:60, 1959.

[5] Karp, R.M., "Reducibility among combinatorial problems", *Complexity of Computer Computations*, *Advances in Computing research*, pp. 85-103, Plenum Press, 1972.

[6] Goss, S., Aron, S., Deneubourg, J.L., Pasteels, J.M., "Self-organized shortcuts in the Argentine ant", *Naturwissenschauften*, pp. 76:579-581, 1989.

[7] Chinneck, J.W., "Feasibility and Infeasibility in Optimization: Algorithms and Computational Methods", *Chapter 1*, pp. 1, Springer, 2008.

[8] Lee, J., "A First Course in Combinatorial Optimization", *Chapter 1: Introduction*, pp. 1, Cambridge University Press, 2004.

[9] Schrijver, A., "On the History of Combinatorial Optimization (till 1960)", pp. 41, CWI (Centre for Mathematics and Computer Science), Amsterdam, Netherlands, 1990.

[10] http://www.tsp.gatech.edu/history/milestone.html

[11] Diaby, M., "The Travelling Salesman Problem: A Linear Programming Formulation", *WSEAS Transactions on Mathematics*, issue 6, vol. 6, pp. 745-754, 2007.

[12] Ralphs, T.K., Kopman, L., Pulleyblank, W.R., Trotter Jr., L.E., "On the Capacitated Vehicle Routing Problem", *Mathematical Programming, Series B 94*, pp. 343, 2003.

[13] Gillett, B.E., Johnson, J.G., "Multi-Terminal Vehicle-Dispatch Algorithm", *Omega*, vol.4, pp. 771-718, 1976.

[14] Tillman, F., "The multiple terminal delivery problem with probabilistic demands", *Transportation Science*, vol. 3, pp. 192-204, 1969.

[15] Chao, I.M., Golden, B., Wasil, E., "A new heuristic for the multidepot vehicle routing problem that improves upon best-known solutions", *American Journal of Mathematical and Management Sciences*, vol. 13, no. 3, pp. 371-406, 1993.

[16] Laporte, G., Nobert, Y., Taillefer, S., "Solving a family of multi-depot vehicle routing and location-routing problems", *Transportation Science*, vol. 22, p.172, 1988.

[17] Lim, A., Wang, F., "Multiple-Depot Vehicle Routing Problem: A One-Stage Approach", *IEEE Transactions on Automation Science and Engineering*, vol. 2, no. 4, pp. 397- 402, 2005.

[18] Carlsson, J., Ge, D., Subramaniam, A., Wu, A., Ye, Y., "Solving Min-Max Multi-Depot Vehicle Routing Problem", *Fields Book on Global Optimization*, 2007.

[19] Johnson, D.S., "Approximation Algorithms for Combinatorial Problems", *STOC '73 Proceedings of the fifth annual ACM symposium on Theory of computing*, 1973.

[20] Laporte, G., "The Traveling Salesman Problem: An overview of exact and approximate algorithms", *European Journal of Operational Research 59/2*, pp. 231-248, 1992.

[21] Laporte, G., "The Vehicle Routing Problem: An overview of exact and approximate algorithms", *European Journal of Operational Research 59*, pp. 345-358, 1992.

[22] Kara, I., Bektas, T., "Integer Linear Programming Formulation of the Generalized Vehicle Routing Problem", *Proceedings of the 5th EURO/INFORMS Joint International Meeting*, 2003

[23] Dantzig, G.B., Fulkerson, D.R., Johnson, S.M., "Solution of a large-scale travelling salesman problem", *Operations Research 2*, pp. 393-410, 1954.

[24] Carpaneto,G., Toth, P., "Some new branching and bounding criteria for the asymmetric travelling salesman problem", *Management Science 26*, pp. 736-743, 1980.

[25] Held, M., Karp, R.M., "The travelling salesman problem and minimum spanning trees", *Operations Research 18*, pp. 1138-1162, 1970.

[26] Held, M., Karp, R.M., "The travelling salesman problem and minimum spanning trees: Part II", *Mathematical Programming 1*, pp. 6-25, 1971.

[27] Miliotis, P., "Integer programming approaches to the travelling salesman problem", *Mathematical Programming 10*, pp. 367-378, 1976.

[28] Miliotis, P., "Using cutting planes to solve the symmetric salesman problem", *Mathematical Programming 15*, pp. 177-188, 1978.

[29] Christofides, N., Mingozzi, A., Toth, P., "Exact algorithms for the vehicle routing problem, based on the spanning tree and shortest path relaxations", *Mathematical Programming 20*, pp. 255-282, 1981.

[30] Fisher, M.L., Jaikumar, R., "A generalized assignment heuristic for vehicle routing", *Networks 11*, pp. 109-124, 1981.

[31] Rosenkrantz, D.J., Stearns, R.E., Lewis, II, P.M., "An analysis of several heuristics for travelling salesman problem", *SIAM Journal on Computing 6*, pp. 563-581, 1977.

[32] Karp, R.M., "A patching algorithm for the nonsymmetric travelling-salesman problem", *SIAM Journal on Computing 8*, pp. 561-563, 1979.

[33] Lin, S., "Computer solutions of the traveling salesman problem", *Bell System Computer Journal 44*, pp. 2245-2269, 1965.

[34] Kirkpatrick, S., Gelatt, Jr., C.D., Vecchi, M.P., "Optimization by Simulated Annealing", *Science 220*, pp. 671-680, 1983.

[35] Glover, F., "Tabu Search", *Report 88-3*, Centre for Applied Artificial Intelligence(CAAI), Graduate School of Business, University of Colorado, 1988.

[36] Clarke, G., Wright, J.W., "Scheduling of vehicles from a central depot to a number of delivery points", *Operations Research 12*, pp. 568-581, 1964.

[37] Gillett, B., Miller, L., "A heuristic algorithm for the vehicle dispatch problem", *Operations Research 22*, pp. 340-349, 1974.

[38] Christofides, N., Mingozzi, A., Toth, P., "The vehicle routing problem", *Combinatorial Optimization*, Wiley, Chichester, pp. 315-338, 1979.

[39] Beni, G., Wang, J., "Swarm Intelligence in Cellular Robotic Systems", *NATO ASI Series*, vol. 102, pp. 703-712, 1993.

[40] Bonabeau, E., Meyer, C., "Swarm Intelligence: A whole new way to think about business", *Harvard Business Review*, R0105G, pp. 105-114, May 2001.

[41] Bonabeau, E., Dorigo, M., Theraulaz, G., "Swarm Intelligence: From Natural to Artificial System", *Oxford University Press*, New York, 1999.

[42] Dorigo, M., Colorni, A., "The Ant System: Optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man and Cybernetics –Part B*, vol. 26, no. 1, pp. 1-13, 1996.

[43] Dorigo, M., Gambardella, L.M., "Ant Colonies for the Travelling Salesman Problem", *BioSystems 43*, pp 73-81, 1997.

[44] Dorigo, M., Gambardella, L.M., "Ant Colony System: A Cooperative Learning Approach to the Travelling Salesman Problem", *IEEE Transactions on Evolutionary Computation*, vol.1, no.1, 1997.

[45] Stutzle, T., Dorigo, M., "ACO Algorithms for the Travelling Salesman Problem", *Periauz(eds)*, Evolutionary Algorithms in Engineering and Computer Science: Recent Advances in Genetic Algorithms, Evolution Strategies, Evolutionary Programming, Genetic Programming and Industrial Applications, Wiley, 1999.

[46] Bullnheimer, B., Hartl, R.F., Strauss, C., "Applying the Ant System to the Vehicle Routing Problem", *Proceedings of 2nd International Conference on Metaheuristics*, 1997.

[47] Bell, John E., McMullen, Patrick R., "Ant Colony Optimization Techniques for the Vehicle Routing Problem", *Advanced Engineering Informatics 18*, pp 41 – 48, 2004.

[48] Rizzoli,A.E., Oliverio F., Montemanni R., Gambardella L.M., "Ant Colony Optimization for vehicle routing problems: from theory to applications", *Technical Report IDSIA-15-04*, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale, September 2004.

[49] Paessens, H., "The savings algorithm for the vehicle routing problem", *European Journal of Operations Research 34(3)*, pp. 336–344, 1988.

[50] Beardwood, J., Halton, J.H., Hammersley, J.M., "The shortest path through many points", *Proceedings of Cambridge Philosophical Society*, vol. 55, pp. 299-327, 1959.

[51] Carlsson, J., Armbruster, B., Ye, Y., “Finding equitable convex partitions of points in a polygon efficiently”, *ACM Transactions on Algorithms*, vol. 6, issue 4, article no. 72, 2010.

[52] Stone, A.H., Tukey, J.W., “Generalized sandwich theorems”, *Duke Mathematical Journal*, vol. 9, pp. 356-359, 1942.

[53] Zivaljevic, R.T., “Topological Methods”, *Handbook of Discrete and Computational Geometry*, chapter 11, pp. 209-224, 1997.

[54] Bespamyatnikh, S., Kirkpatrick, D.G., Snoeyink, J., “Generalizing Ham Sandwich Cuts to Equitable Subdivisions”, *Symposium on Computational Geometry (SoCG)*, pp. 49-58, 1999.

[55] Bullnheimer, B., Hartl, R.F., Strauss, C., “An improved ant system algorithm for the vehicle routing problem”, *Annals of Operations Research*, vol. 89, pp. 319-328, 1999.

[56] Lin, S., Kernighan, B.W., “An effective heuristic algorithm for the TSP”, *Operations Research*, vol. 21, pp. 498-516, 1973.

[57] http://www.stanford.edu/class/msande310/handout.shtml

[58] http://www.tsp.gatech.edu/concorde.html

[59] Neumann, F., Sudholt, D., Witt, C., “Computational Complexity of Ant Colony Optimization and Its Hybridization with Local Search”, *Innovations in Swarm Intelligence*, vol. 248, pp. 91-120, 2009.

[60] O’Rourke, J., “Computational Geometry in C (Second Edition)”, Chapter 2, 1998