CAPACITATED VEHICLE ROUTING PROBLEM BY APPLYING ANT COLONY SYSTEM WITH SAVING HEURISTIC

MAY CHOON HTIKE

M.C.Sc. (THESIS) AUGUST, 2016

CAPACITATED VEHICLE ROUTING PROBLEM BY APPLYING ANT COLONY SYSTEM WITH SAVING HEURISTIC

MAY CHOON HTIKE

M.C.Sc. (THESIS) **AUGUST, 2016**

CAPACITATED VEHICLE ROUTING PROBLEM BY APPLYING ANT COLONY SYSTEM WITH SAVING HEURISTIC

BY

MAY CHOON HTIKE

B.C.Sc. (Hons:)

Dissertation submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science (M.C.Sc.)

of

COMPUTER UNIVERSITY (MANDALAY)

AUGUST, 2016

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude and sincere thanks to all persons who contributed directly or indirectly towards the success of this thesis.

I owe my respectful thanks to **Dr. Win Aye**, the Rector, Computer University (Mandalay), for her kind permission to make this thesis in this university.

I am deeply thankful to **Dr. Thi Thi Soe**, Associate Professor, Department of Software Technology, Computer University (Mandalay), for her continuous, sincere, and kind guidance during the period of completing this thesis.

I am greatly indebted to my supervisor, Dr. **Aye Aye Chaw**, Associate Professor, Head of Department of Software Technology, Computer University (Mandalay), for her valuable supervision, good advice, detailed guidance, and helpful suggestions throughout the terms of finishing the thesis.

I would like to extend my deepest gratitude to **U Thaung Kyaw**, Associate Professor, Head of Department of English at Computer University (Mandalay), for editing my thesis.

Finally, I also thank all the members of Computer University (Mandalay), who attended the seminars on this thesis, for their support, valuable suggestions, helpful hints, and fair criticisms.

Moreover, I also thank all my **mentors and friends**, for their invaluable suggestions and helpful contributions to success of this thesis.

Especially, my deepest thanks go to **my parents** for giving me moral and material support, sympathy and empathy, care and kindness throughout my studies and my efforts to complete this thesis without any trouble.

ABSTRACT

Ant Colony Optimization (ACO) is a meta-heuristic algorithm for solving the hard combinational optimization problems. The ACO heuristics is a distributed and cooperative search method that imitates the behavior of real ants in its the search for food. The Vehicle Routing Problem (VRP) concerns with the transport of items between depots and customers by means of a fleet of vehicles. The Capacitated Vehicle Routing Problem (CVRP) is the basic version of the VRP. The CVRP concerns the design of a set of minimum cost routes, starting and ending at a single depot, for a fleet of vehicles to service a number of customers with known demands. In this system, we apply the Ant Colony System (ACS) metaheuristics which is the computational method for CVRP, combined with Saving heuristic algorithm. In this approach, we calculate the minimum distance routes between the depot and the customers for the two Benchmark datasets, in which, the vehicles construct the routes by successively choosing customer to visit, continuing until all customer has been reached. This system implement by using C# programming language.

CONTENT

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

CHAPTER 1

INTRODUCTION

Ant Colony Optimization (ACO) algorithms are part of swarm intelligence algorithms that are made up of simple individuals that cooperate through self-organization without any form of central control over the swarm members. ACO is inspired in the foraging behavior of real ants. When searching for food, ants initially explore the area surrounding their nest in a random manner. While moving, ants leave a chemical pheromone trail on the ground. Ants communicate using pheromone trail. Ants deposit pheromones along the path that other ants can follow. ACO uses the pheromone trail as communication medium. In this system, vehicles are represents the ants, food are represent the customers and the nest represents the central depot.

Vehicle Routing Problem (VRP) is concerned with transport of items between depots and customers by means of a fleet of vehicle. In VRP, goods are delivered from a single depot and return to it. The VRP finds the best route to service all customers by a fleet of vehicle and minimize the transportation cost.

Capacitated Vehicle Routing Problem (CVRP) is one of the Vehicle Routing Problems. It concerns the design of a set of minimum cost routes, starting and ending at a single depot, floor a fleet of vehicles to service a number of customers with known demands. In ACS, computational methods for CVRP are exact methods, heuristic and metaheuristic methods. In this system we apply the ACS metaheuristics.

So, we intended to apply the Ant Colony System (ACS) algorithm with Saving Heuristic to solve the CVRP problem. In this system, vehicles concurrently and asynchronously move through adjacent states of the problem by building the routes. They move by applying pheromone trail and heuristic information. Once the system has built an solution, the system will evaluates the partial solution and deposits pheromone trails on the components or connections it used. This pheromone information will direct the search of the future vehicles. While moving, the system will incrementally build solutions to the optimization problem.

In this system, initially, m vehicles are positioned on n customers randomly and initial pheromone trail levels are applied to arcs. In order to solve the CVRP, the vehicles construct solutions by successively choosing a customer to visit, continuing until each customer has been visited. When constructing routes if all remaining choices would result in an infeasible solution due to vehicle capacity being exceeded then the depot is chosen and a new route is started. Vehicles choose the next city to visit using a combination of heuristic and pheromone information. During the construction of a route, the vehicle modifies the amount of pheromone on the chosen arc by applying a local updating rule. Once all vehicles have constructed their tours, then the amount of pheromone on arcs belonging to the best solution, as well as the global best solution, are updated according to the global updating rule. Finally, the system gives the minimum numbers of routes, vehicle used and distances to travel.

1.1 Objective of the Thesis

Objectives of the Thesis are

- To study the Ant Colony Optimization and ACS with Saving algorithm
- To understand the theories of VRP and CVRP in detail
- To calculate the transportation cost of CVRP by applying ACS with an algorithm of Saving

of the problem by building the routes. They move by applying pheromone trail and heuristic information. Once the system has built an solution, the system will evaluates the partial solution and deposits pheromone trails on the components or connections it used. This pheromone information will direct the search of the future vehicles. While moving, the system will incrementally build solutions to the optimization problem.

In this system, initially, m vehicles are positioned on n customers randomly and initial pheromone trail levels are applied to arcs. In order to solve the CVRP, the vehicles construct solutions by successively choosing a customer to visit, continuing until each customer has been visited. When constructing routes if all remaining choices would result in an infeasible solution due to vehicle capacity being exceeded then the depot is chosen and a new route is started. Vehicles choose the next city to visit using a combination of heuristic and pheromone information. During the construction of a route, the vehicle modifies the amount of pheromone on the chosen arc by applying a local updating rule. Once all vehicles have constructed their tours, then the amount of pheromone on arcs belonging to the best solution, as well as the global best solution, are updated according to the global updating rule. Finally, the system gives the minimum numbers of routes, vehicle used and distances to travel.

1.1 Objective of the Thesis

Objectives of the Thesis are

- To study the Ant Colony Optimization and ACS with Saving algorithm
- To understand the theories of VRP and CVRP in detail
- To calculate the transportation cost of CVRP by applying ACS with an algorithm of Saving

- To search the optimal minimum cost route between depot and customers

1.2 Organization of the Thesis

The thesis is organized as follows. Chapter 1 is the introduction and objective of the system. Chapter 2 is the theory background giving a brief introduction to Swarm Intelligence, artificial ant, foraging behavior of Ant and Optimization, ACO Metaheuristic, application area of ACO and ACO Algorithms. Chapter 3 presents the methodology: ACS with Heuristic for CVRP. Chapter 4 shows the design and implementation of the system on the methodology as explained in Chapter 3. Chapter 5 gives the conclusion.

CHAPTER 2

THEORY BACKGROUNG

This chapter gives a short literature review on Swarm Intelligence [12], Ant Colony Optimization [7] and its applications including the routing problems, and Vehicle Routing Problem [10]. This chapter also briefly describes the foraging behavior of Ants, Ant Colony Optimization Metaheuristic and Application of ACO.

2.1 Swarm Intelligence

Scientists discovered the variety of the interesting insect or animal behaviors in the nature. Many animal groups clearly display structural order, with the behavior of the organisms so integrated that even though they may change shape and direction, they appear to move as a single coherent entity. This kind of aggregate motion is called collective behavior or swarm behavior. This collective behavior can be seen in bird flocks, fish schools as well as in insects like midges and mosquitoes. For example, for a bird to participate in a flock, it only adjusts its movements to coordinate with the movements of its flock mates, typically its neighbors that are close to it in the flock. A bird in a flock simply tries to stay close to its neighbors, but avoid collisions with them. Each bird does not take commands from any leader bird since there is no lead bird. Any bird can fly in the front, center and back of the swarm. Swarm behavior helps birds take advantage of several things including protection from predators – especially for birds in the middle of the flock – and searching for food – essentially each bird is exploiting the eyes of every other bird.

The main principles of the swarm behavior are as follows:

Collision Avoidance	: Avoid with nearby flock mates.
Flock Centering	: Attempt to match velocity and to stay close with nearby flock mates.
Homogeneity	: Every bird in flock has the same behavior model. The flock moves without a leader, even though temporary leaders seem to appear.
Locality	: The motion of each bird is only influenced by its nearest flock mates. Vision is considered to be the most important senses for flock organization.

In swarm behavior, all the time individuals attempt to maintain a minimum distance between themselves and others. This rule has the highest priority and corresponds to an often times observed behavior of animals in nature. This swarm behavior is identified with four collective dynamical behaviors. These are swarm, torus, dynamic parallel group, and highly parallel group. In the first case, one is an aggregate with cohesion, but a low level of polarization among members. In the second case, individuals perpetually rotate around an empty core in random direction. In the third case, individuals are polarized and move as a coherent group, but individuals can move throughout the group and density and group form can fluctuate. In the last case, one is much more static in terms of exchange of spatial positions within the group than the dynamic parallel group and the variation in density and form is minimal.

From high-level view of the first one, a swarm can be viewed as a group of agents cooperating to achieve some purposeful behavior and achieve some goal. This apparent collective intelligence seems to emerge from what are often large groups of relatively simple agents. The agents use simple local rules to govern their actions and via the interactions of the entire group, the swarm achieves its objectives. A type of self-organization emerges from the collection of actions of the group. An autonomous agent is a subsystem that interacts with its environment, which probably consists of other agents, but acts relatively and independently from all other agents. The autonomous agent does not follow commands from a leader, or some global plan. The emergent collective intelligence of groups of simple autonomous agents is called Swarm Intelligence. Thus, it is a learning and optimization technique in which individuals are associated with act to create a potential solution.

Several collective behavior inspired algorithms have been proposed since 1990. Among these, Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) [1] are two most notable algorithms in this domain.

PSO which incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior, from which the idea is emerged, is a population-based search algorithm which is initialized with a population of random solutions, called particles. Unlike in the other evolutionary computation techniques, each particle in PSO is also associated with a velocity. Particles fly through the search space with velocities which are dynamically adjusted according to their historical behaviors. Therefore, the particles have the tendency to fly towards the better and better search area over the course of search process. The PSO was first designed to simulate birds seeking food which is defined as a cornfield vector.

ACO deals with artificial systems that are inspired from the foraging behavior of real ants, which are used to solve discrete optimization problems. A detailed description of ACO is described in next following section.

2.2 From Real to Artificial Ants

Ant colonies, and more generally social insect societies, are distributed systems that, in spite of the simplicity of their individuals, presented a highly structured social organization. As a result, ant colonies can accomplish complex tasks that in some cases far exceed the individual capabilities of a single ant.

The field of ''ant algorithms'' studies models derived from the observation of real ants' behavior, and uses these models as a source of inspiration for the design of novel algorithms for the solution of optimization and distributed control problems.

The main idea is that the self-organizing principles which allow the highly coordinated behavior of real ants can be exploited to coordinate populations of artificial agents that collaborate to solve computational problems. Several different aspects of the behavior of ant colonies have inspired different kinds of ant algorithms. Examples are foraging, division of labor, brood sorting, and cooperative transport. In all these examples, ants coordinate their activities via stigmergy, a form of indirect communication mediated by modifications of the environment. For example, a foraging ant deposits a chemical on the ground which increases the probability that other ants will follow the same path. Biologists have shown that many colony-level behaviors observed in social insects can be explained via rather simple models in which only stigmergic communication is present. In other words, biologists have shown that it is often sufficient to consider stigmergic, indirect communication to explain how social insects can achieve self-organization. The idea behind ant algorithms is then to use a form of artificial stigmergy to coordinate societies of artificial agents.

One of the most successful examples of ant algorithms is known as ''Ant Colony Optimization,'' or ACO. ACO is inspired by the foraging behavior of ant colonies, and targets discrete optimization problems. This introductory chapter describes how real ants have inspired the definition of artificial ants that can solve discrete optimization problems.

2.3 Ants' Foraging Behavior and Optimization

The visual perceptive faculty of many ant species is only rudimentarily developed and there are ant species that are completely blind. In fact, an important insight of early research on ants' behavior was that most of the communication among individuals, or between individuals and the environment, is based on the use of chemicals produced by the ants. These chemicals are called pheromones. This is different from, for example, what happens in humans and in other higher species, whose most important senses are visual or acoustic. Particularly important for the social life of some ant species is the trail pheromone. Trail pheromone is a specific type of pheromone that some ant species, such as Lasius niger or the Argentine ant Iridomyrmex humilis (Goss, Aron, Deneubourg, & Pasteels, 1989) [7], use for marking paths on the ground, for example, paths from food sources to the nest. By sensing pheromone trails foragers can follow the path to food discovered by other ants. This collective trail-laying and trail-following behavior whereby an ant is influenced by a chemical trail left by other ants is the inspiring source of ACO.

Pheromone continuously evaporates. Once the pheromone level falls down under a threshold, ants are not influenced to follow the trail. Therefore, if a trail is not used for a time, it tends to disappear. This autocatalytic phenomenon of positive feedback is the key point for establishing the shortest path from nest to food source (Beckers et al., 1992) [7].

The first and foremost ACO algorithm was Ant System (AS) which was developed in 1992 by Marco Dorigo [8] as his PhD thesis AS became very popular after its publication by Dorigo and colleagues in 1996. Many researchers have since developed improvements to the original algorithm, and applied them to a range of different problems.

The first member ACO algorithm AS, a translation of observed ant behavior into working optimization algorithm, was innovated by Dorigo and colleagues in 1991 [7], and was more fully described by Dorigo in next year. AS used exploits three different pheromone updating and is designed as a set of algorithms: ant-density, ant-quantity, ant-cycle. In the first two algorithms, ant deposit pheromone when building a solution while in the last one ants deposit after they have built a tour. From the experiment on benchmark application, it was shown that ant-cycle performance is much better than the other two.

After the introduction of AS, abroad mutants and optimization of AS were later developed by inspiring the most performing ant-cycle, and eventually it leads to formalize an ACO-metaheuristic approach called Ant Colony Optimization (ACO) by Dorigo and DiCaro in 1999 and Dorigo and colleagues in 2000 [7]. The inspiring source of ACO is the pheromone trail lying and adopting the behavior of real ants which use pheromone as a communication medium. ACO is based on the Stigmergic Communication of a colony of simple agents, i,e. artificial ant or ant for short, mediated by artificial pheromone trails which serve as a distributed information. Artificial ants used in ACO are stochastic solution construction procedures that probabilistically build a solution by iteratively adding solution components to partial solutions by taking into account heuristic information on the problem instance being solved, if available, and artificial pheromone trails which change dynamically at run-time to reflect the agents' acquired search experience. In ACO algorithm [9], the artificial ants act the following behavior. A colony of the ants concurrently and asynchronously moves through the next state of problem by building paths on constructed graph. They move by applying a stochastic local decision policy that makes use of pheromone trails and heuristic information. By moving, ants incrementally build solutions to the optimization problem. Once an ant has built a solution, or while the solution is being built, the ant evaluates the partial solution and deposits pheromone trails on the components of connections it used. This pheromone information will direct the search of the future ants.

Besides ants' activity an ACO algorithm includes two more procedures: pheromone trail evaporation and daemon action (optional). Pheromone evaporation is the process by means of which the pheromone trail intensity on the components decreases over time. From a practical point of view, pheromone evaporation is needed to avoid a too rapid convergence of the algorithm towards a sub-optimal region. It implements a useful form of forgetting, favoring the exploration of new areas of the search space. Daemon actions can be used to implement centralized actions which cannot be performed by single ants.

Examples are the activation of a local optimization procedure, or the collection of global information that can be used to decide whether it is useful or not to deposit additional pheromone to bias the search process from a non-local perspective. As a practical example, the daemon can observe the path found by each ant in the colony and choose to deposit extra pheromone on the components used by the ant that built the best solution. Pheromone updates performed by the daemon are called off-line pheromone updates.

2.5 Metaheuristic

Combinatorial optimization problems are intriguing because they are often easy to state but very difficult to solve. Mandy of the problems arising in applications are NP-hard, that is, it is strongly believed that they cannot be solved to optimality within polynomially bounded computation time, Hence, to practically solve large instances one often has to use a approximate methods which return near-optimal solutions in a relatively short time. Algorithms of this type are loosely called heuristics. They often use some problem-specific knowledge to either build or improve solutions.

A metaheuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems. The use of metaheuristics has significantly increased the ability of finding very high quality solutions to hard, practically relevant combinatorial optimization problems in a reasonable time.

Metaheuristics are typically high-level strategies which guide an underlying, more problem specific heuristic, to increase their performance. The main goal is to avoid the disadvantages of iterative improvement and, in particular, multiple descent by allowing the local search to escape from local optima. This is achieved by either allowing worsening moves or generating new starting solutions for the local search in a more "intelligent" way than just providing random initial solutions. Many of the methods can be interpreted as introducing a bias such that high quality solutions are produced quickly. This bias can be of various forms and can be cast as descent bias (based on the objective function), memory bias (based on previously made decisions) or experience bias (based on prior performance). Many of the metaheuristic approaches rely on probabilistic decisions made during the search. But, the main difference to pure random search is that in metaheuristic algorithms randomness is not used blindly but in an intelligent, biased form.

Some variants of metaheuristics are

- Simulated Annaling
- Tabu Search
- Genetic Algorithm
- Ant Colony Optimization (ACO)
- Variable Neighbourhood Search

2.5.1 The ACO Metaheuristic

A particularly successful metaheuristic is inspired by the behavior of real ants. Starting with Ant System, a number of algorithmic approaches based on the very same ideas were developed and applied with consideravle success to a variety of combinatorial optimization problems form academic as well as from real-world applications.

Ant colony optimization is a metaheuristic in which a colony of artificial ants cooperates in finding good solutions to difficult discrete optimization problems. Cooperation is a key design component of ACO algorithms: The choice is to allocate the computational resources to a set of relatively simple agents (artificial ants) that communicate indirectly by stigmergy, that is, by indirect communication mediated by the environment.

ACO algorithms can be used to solve both static and dynamic combinatorial optimization problems. Static problems are those in which the characteristics of the problem are given once and for all when the problem is defined, and do not change while the problem is being solved.

A paradigmatic example of such problems is the TSP (Johnson & McGeoch, 1997; Lawler, Lenstra, Rinnooy Kan, & Shmoys, 1985; Reinelt, 1994), in which city locations and their relative distances are part of the problem definition and do not change at run time. On the contrary, dynamic problems are defined as a function of some quantities whose value is set by the dynamics of an underlying system. The problem instance changes therefore at run time and the optimization algorithm must be capable of adapting online to the changing environment.

```
Procedure of ACO Metaheuristic
        Procedure ACO_MetaHeuristic
        ScheduleActivities
        Construct Ants Solutions
        Update Pheromones
        Daemon Actions
        End-ScheduleActivities
End-procedure
```

Figure 2.1 Procedure of ACO Metaheuristic

In Figure 2.1, the ACO metaheuristic behavior is described in pseudo-code. The main procedure of the ACO metaheuristic manages the scheduling of the three above-discussed components of ACO algorithmns via the Schedule Activities construct:

(i) management of the ants' activity

(ii) pheromone updating, and

(iii) daemon actions.

The ScheduleActivities construct does not specify how these three activities are scheduled and synchronized. In other words, it does not say whether they should be executed in a completely parallel and independent way, or if some kinds of synchronization among them are necessary. The designer is therefore free to specify the way these three procedures should interact.

2.6 Applications of ACO

This section provides an overview of the most notable applications of ACO to different combinatorial optimization and other problems. In this literature review, the applications are classified and presented into subcategories based on the problems nature [7].

2.6.1 Assignment Problems

Assignment Problems involve the allocation of resources to items to a number of constraints. The distinguishing characteristics of these problems are that they contain two distinct types of entity "items and resources", and that while all items must be assigned a resource, the number of items assigned to one resource can vary from zero to many depending on the problem.

For the practical application of the ACO metaheuristic to assignment problem, two pheromones are used in their system; one is used to select the next item to be assigned and another is used to determine which resource to assign to which item. ACO algorithm for the Generalized Assignment Problem (GAP) has taken this second one approach, and associate pheromone with the assignments made, the addition of a local search procedure improved results for the algorithm significantly. In the generalized assignment problem, a set of tasks has to be assigned to a set of agents in such a way that a cost function is minimized.

The Quadratic Assignment Problem (QAP) is a facility layout problem, with applications in building and office layout, keyboard design and scheduling. It consists of assigning n facilities (i,e., items) to n locations (i,e., resources), with exactly one facility assigned to each location. The aim is to minimize the total cost of flows between facilities (determined by a mixture of the distance between locations and the amount of flow required between facilities). Many ACO algorithms have been developed for this problem. Many of these ACO algorithms produced results competitive with alterative solution approaches.

2.6.2 Scheduling Problems

A scheduling problem is a problem for solving the optimal schedule under various objectives, different environment, and characteristic of the various jobs. There is a myriad (various) of terms related to this problem.

Scheduling is concerned with the allocation of scarce resources to tasks over time. Scheduling problems are central to production and manufacturing industries, but also arise in a variety of other settings. In shop scheduling problems, where jobs have to be processed on one or several machines such that some objective function is optimized.

In 1994, an ACO algorithm for the Job Shop Problem (JSP) was developed by Colorni, Dorigo, aniezzo and Trubian [7]. The construction graph they define is made up of the operations to be scheduled plus an additional node representing the empty sequence from which ants start. This extra node is required as pheromone is associated with the edges of the graph and, unlike the TSP, it is important to know which operation appears first in the schedule, Again in 1999, Vander Zwaan and Marques describe an ACO algorithm similar to above Colorni and colleagues' [7]; with only minor changes to the way pheromone is updated.

The majority of scheduling ACO algorithms associates pheromone with the absolute position of an operation in the permutation. This innovate removes the need for an artificial start node and appears suited to describing permutations than associating pheromone with pairs of adjacent components. This approach has been used in ACO algorithms of the Flow Shop Problem (FSP), and Single Machine Total Tardiness Problem (SMTTP).

2.6.3 Subset Problem

Subset selection problems involve finding an optimal feasible subset of an initial set of objects with respect to an objective function and/or some constraints. Many well-known combinatorial problems are members of this class, e.g., Maximum Clique Problems, Knapsack Problems, Boolean Satisfiability Problems, Constraint Satisfaction Problems, and Graph Matching Problems. Among these problem, in 1999, the algorithm for Multiple Knapsack Problem was developed by Leguizamon and Smith.

In subset problems, a solution to the problem under consideration is represented as a subset of the set of available items (components) subject to problem-specific constraints. For example, in the TSP a solution may be seen as consisting of a subset of the set of available arcs. However, these problems are often represented, more conveniently, using other representations, such as a permutation of the graph nodes, cities, in the TSP case.

For Sub Set problems in ACO, solutions constructed by ants are subsets of objects, and the order in which objects are selected is not significant. Hence, a first pheromone strategy consists in laying pheromone on objects of the best constructed solutions. When compared to the applications other problem, there are two main particularities involved with ACO applications to subset problems. First, in subset problems one is not particularly interested in an ordering of the components. Therefore, the most recently incorporated item need not be necessarily considered when selecting the next item to be added to the solution under construction. As a result, in subset problems pheromone trails are typically associated with components and not with connections. Second, the number of components in the solutions built by different ants may be different, so that the solution construction phase ends only when all the ants have completed their solutions.

2.6.4 Application to Routing Problems

The Travelling Saleman Problems (TSP) is the first problem that the ant algorithms were applied due to the high degree of similarity between ants finding the shortest path to a food source and artificial ants finding the shortest Hamiltonian cycle in a graph. Consequently, each major ACO algorithm has been applied to either the symmetric TSP or asymmetric TSP (ATSP).

Each of these applications follows the approach of the first AS for the TSP. ACS, MMAS and ASrank algorithms for TSP produced improved results compared the early ACO algorithms, and the best results have been obtained when local search heuristics were used to improve the solutions produced at each iteration.

A candidate set is a subset of the available components, chosen to reduce the number of components that must be considered and often consisting of components that appear to be good to include.

In ACS for the TSP, statically generated candidate sets (i.e. a subset of the available components, chosen to reduce the number of components that must be considered and often consisting of components that appear to be good to include) have been used where a set of the closest cities is maintained for each city. In effect, these separate sets of components represent a single set of candidate edges, but it is useful for the TSP to consider them separately. When constructing solutions, ants consider first only those components in the appropriate candidate set and, only if that is exhausted, do they examine the remaining components. The use of static candidate sets has allowed the application of ACS to TSP instances of more than 1000 cities. The use of dynamically generated candidate sets, that take pheromone information into account, has also improved algorithm speed as well as achieving improvements in solution quality.

Another problem like TSP is the Vehicle Routing Problem (VRP), which consists of delivering commodities of different weights to a number of customers/cities using vehicles with specific capacities: minimize the number of vehicles used, the total distance traveled [5]. An added complication is that vehicles must start at and return to a depot/warehouse. Each vehicle also has maximum distance/time for its route. The classical vehicle routing problem (VRP) aims to find a set of routes at a minimal cost (finding the shortest path, minimizing the number of vehicles, etc) beginning and ending the route at the depot, so that the known demand of all nodes are fulfilled. Each node is visited only once, by only one vehicle, and each vehicle has a limited capacity. Some formulations also present constraints on the maximum traveling time.

Bullnheimer and colleagues [7] developed a version of their ASrank algorithm for this problem. They note that once customers have been assigned to vehicles, the VRP reduces to a number of TSPS. Hence, their algorithm is applied to the VRP in the same way as ACO algorithms for the TSP, with the modification. When the current route makes it impossible to select another customer without violating a vehicle's capacity, or would exceed the vehicle's maximum route length, the next component chosen is the component representing the depot.

Using a more complex heuristic measure than the inverse of the distance between customers can lead to improved results. This observation was found by Bullnheimer and colleagues and Doerner and colleagues [7]. Both use static candidate sets in their respective ACO algorithms. Doerner and colleagues use sophisticated heuristic information in their ACO algorithm and they showed that theirs is performed better than a number of alternative metaheuristics. Bullnheimer and colleagues showed that their algorithm was able to find good solutions to a number of benchmark instances, but these solutions cannot be competitive with those of the best performing algorithms for these problems.

Another one is the Sequential Ordering Problem (SOP) for production planning. SOP finds a minimum Hamiltoniam path in directed graph with weight on the edges and on the nodes subject to the precedence constraints among nodes. With some additional precedence constraints, SOP can be modeled as an ATSP. By applying this, Hybrid Ant Systems for SOP (HAS-SOP) was published by Gambardella and Dorigo [7]. HAS-SOP uses a local search heuristic, a relatively novel feature at the time of their proposed, to improve the solutions.

2.7 Variations of ACO Algorithms

The main mechanism at work in ACO algorithms that triggers the discovery of good tours is the positive feedback given through the pheromone update by the ants: the shorter the ant's tour, the higher the amount of pheromone the ant deposits on the arcs of its tour. This in turn leads to the fact that these arcs have a higher probability of being selected in the subsequent iterations of the algorithm. This section explains the variations of the ACO algorithms.

2.7.1 Max-Min Ant System (MMAS)

MAX-MIN ant system (MMAS) is another improvement, proposed by Stutzle and Hoos (2000), over the original ant system idea. MMAS differs from AS in that (i) only the best ant adds pheromones trails, and (ii) the minimum and maximum values of the pheromone are explicitly limited (AS and ACS) these values are limited implicitly, that is the value of the limits is a result of the algorithm working rather than a value set explicitly by the algorithm designer). Added Maximum and Minimum pheromone amounts [τ_{max},τ_{min}] only global best or iteration best tour deposited pheromone all edges are initialized to τ_{max} and reinitialized to τ_{max} when nearing stagnation.

2.7.2 Rank-Based Ant System (ASrank)

Another improvement over Ant System (AS) is the rank-based version of AS (ASrank), proposed by Bullnheimer et al. (1999c). In ASrank each ant deposits an amount of pheromone that decreases with its rank. Additionally, the best-so-far ant always deposits the largest amount of pheromone in each iterations. In ASrank, all solutions are ranked according to their fitness. The amount of pheromone deposited is then weighted for each solution, such that the solutions with better fitness deposit more pheromone than the solutions with worse fitness.

2.7.3 Continuous Orthogonal Ant Colony (COAC)

The pheromone deposit mechanism of COAC is to enable ants to search for solutions collaboratively and effectively. By using an orthogonal design method, ants in the feasible domain can explore their chosen regions rapidly and efficiently, with enhanced global search capability and accuracy.

The orthogonal design method and the adaptive radius adjustment method can also be extended to other optimization algorithms for delivering wider advantages in solving practical problems.

CHAPTER 3

METHODOLOGY: ACS WITH HEURISTIC FOR CVRP

3.1 Vehicle Routing Problem

The Vehicle Routing Problem (VRP) is a class of well-known NP-hard combinatorial optimization problems [10]. The VRP is concerned with the design of the optimal routes, used by a fleet of identical vehicles stationed at a central depot to serve a set of customers with known demands. In the basic version of the problem, known as a Capacitated VRP (CVRP), only capacity restriction for vehicles are considered and the objective is to minimize the total cost (or length) of routes.

The study of the VRP is very important. The VRP contributes directly to a real opportunity to reduce costs in the important area of logistics. Logistics can be roughly described as the delivery of goods from one place (supplier) to others (consumers).Transportation management, and more specifically vehicle routing, has a considerable economical impact on all logistic systems.

Due to the nature of the problem, it is not viable to use exact methods for large instances of the VRP. Therefore, most approaches rely on heuristics that provide approximate solutions. Some specific methods have been developed to this problem. Another option is to apply standard optimization techniques, such as tabu search, simulated annealing, constraint programming, genetic algorithms and ant systems. Our main interest is about the metaheuristics used to solve the VRP and more particularly about the ant colony system.

The first algorithm based on the ant colony system, applied to the CVRP, was proposed by [Bullnheimer & al.1999] [7] known as << Ant System>> (AS), applied first for the TSP in [Sorigo & ak.1996] [2]. The pheromone and the nearest neighbor heuristic are used to build the routes of the vehicles.

The same authors [Bullnheimer & al. 1999] [7] proposed an improved AS which consists of replacing the nearest neighbor heuristic, in the transition rule of the basic algorithm AS, is applied for the pheromone updating rule.

3.1.1 Various Types of Vehicle Routing Problem

- Capacitated vehicle routing problem (CVRP)
- Multi-depot vehicle routing problem (MDVRP)
- Vehicle routing problem with time windows (VRPTW)
- Stochastic vehicle routing problem (SVRP)
- Site-dependent vehicle routing problem (SDVRP)
- Open Vehicle Routing Problem (OVRP)
- Period vehicle routing problem (PVRP)
- Vehicle routing problem with pick-up and delivery (VRPPD)

In the CVRP one has to deliver goods to a set of customers with known demands on minimum-cost vehicle routes originating and terminating at a depot. The vehicles are assumed to be homogeneous and having a certain capacity. In some versions of the CVRP one also has to obey a route duration constraint that limits the lengths of the feasible routes. The VRPTW extends the CVRP by associating time windows with the customers. The time window defines an interval during which the customer must be visited. The objective of the VRPTW is to serve a number of customers within predefined time windows at minimum cost (in terms of distance travelled), without violating the capacity and total trip time constraints for each vehicle. The OVRP is closely related to the CVRP, but contrary to the CVRP a route ends as soon as the last customer has been served as the vehicles do not need to return to the depot. The MDVRP extends the CVRP by allowing multiple depots. The SDVRP is another generalization of the CVRP in which one can specify that certain customers only can be served by a subset of the vehicles. Furthermore, vehicles do not need to have the same capacity in the SDVRP. In the CVRP, MDVRP and SDVRP one seeks to minimize the total traveled distance, whereas in the OVRP and VRPTW, the first priority is to minimize the number of vehicles and minimizing the traveled distance is the second priority. The choice of objective is not an intrinsic feature of the problems, but just the tradition in the metaheuristic literature. Most exact methods and some metaheuristics for the VRPTW minimize total traveled distance instead of minimizing number of vehicles used.

3.2 Capacitated Vehicle Routing Problem

In the CVRP one has to deliver goods to a set of customers with known demands on minimum-cost vehicle routes originating and terminating at a depot. The vehicles are assumed to be homogeneous and having a certain capacity. In some versions of the CVRP one also has to obey a route duration constraint that limits the lengths of the feasible routes. The objective is to find the minimum cost route to serve all the customers by satisfying the following constraints.

- Each customer is visited exactly once by exactly one vehicle.
- All vehicle routes start and end at the depot.
- For each vehicle route, the total demand does not exceed the vehicle capacity Q.
- For each vehicle route, the total route length that can travel is restricted.

3.2.1 CVPR Formulation

The CVRP can be represented as a weighted directed graph $G = (V, A)$ where $V = \{ v_0, v_1, v_2, ..., v_n \}$ represents the set of the vertices and $A = \{ (v_i, v_j) : i \neq j \}$ represents the set of arcs. The vertex v_0 represents the depot and the others represent the clients. To each arc (v_i, v_j) a non-negative value d_{ij} is associated. This value corresponds to the distance between the vertex v_i and the vertex v_j in terms of cost or time between the two vertices. A demand q_i and time service δ_i (q_0=0, δ_0=0) are associated with each client (vertex) v_i. In this case, the objective is to minimize the total cost of routing and at the same time respect the following constraints:

(i) Every client is visited exactly once by exactly one vehicle,

(ii) All the vehicles paths/routes start and end at the depot,

(iii) The total demand of clients of each path/route should not exceed the capacity of each vehicle.

(iv) The number of vehicles is supposed to be unlimited; it is calculated during the construction of the routes of vehicles.

3.2.2 Mathematical Formulation of CVRP

The Capacitated Vehicle Routing Problem (CVRP) concerns the design of a set of minimum cost routes, starting and ending at a single depot, for a fleet of vehicles to service a number of customers with known demands. Mathematically, it can be represented by a weighted graph:

- $G = (V, A)$
- $V = \{v_0, v_1, v_2, ..., v_n\}$ where v_0 represents the depot and $v_1, ..., v_n$ all the clients.
- q_i the demand of the client i, $i \in V$
- d_{ij} the distance (cost) between the vertices (clients) v_i and v_j
- $K = \{k_1, k_2, ..., k_m\}$represents the vehicles fleet
- Q the capacity of each vehicle, $k_i \in K$ (the fleet is homogeneous)

In order to find the clients visit order, define the decision variables as follows:

$$x_{ij}^{k} = \begin{cases} 1 & \text{if the vehicle } k \text{ visits the client } j \text{ directly after client } i \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

$$x_{i}^{k} = \begin{cases} 1 & \text{if the client } i \text{ is served by the vehicle } k \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

The objective function is

$$\text{Min} \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} d_{ij} x_{i,j}^{k} \tag{3.3}$$

with the constraints

$$\sum_{k \in V} \sum_{j \in N} x_{i,j}^{k} = 1, \forall i \in V \tag{3.4}$$

$$\sum_{j \in V} x_{i,j}^{k} - \sum_{j \in V} x_{j,i}^{k} = 0, \forall \text{ i} \in \text{V}, \text{k} \in K \tag{3.5}$$

$$\sum_{j \in V} x_{0,j}^{k} = 1, \forall \text{k} \in \text{K} \tag{3.6}$$

$$\sum_{j \in V} x_{j,n+1}^{k} = 1, \forall \text{k} \in \text{K} \tag{3.7}$$

$$x_{i,j}^{k} = 1 \Rightarrow \text{y}_{\text{i}} - q_j = y_j, \forall i, j \in V, k \in K \tag{3.8}$$

$$y_0 = Q, 0 \le y_i, i \in V \tag{3.9}$$

$$x_{i,j}^k \in \{0,1\}, \forall i, j \in V, k \in K \tag{3.10}$$

The function of the Euclidean cost solution $X = (x_{i,j}^k)$, $\forall i, j \in V, k \in K$ is defined by:

$$\mathrm{Cost}(X) = \sum_{k \in K} \sum_{i \in V} \sum_{j \in V} d_{i,j} x_{i,j}^k \tag{3.11}$$

The number of vehicles used by the solution X is defined by:

$$\text{NB Vehicles}(X) = \sum_{k \in K} \sum_{i \in V} x_{0,j}^k \tag{3.12}$$

3.3 ACO Heuristic for Capacitated Vehicle Routing Problem

Using ACO, an individual ant simulates a vehicle, and its route is constructed by incrementally selecting customers until all customers have been visited. Initially, each ant starts at the depot and the set of customers included in its tour is empty. The ant selects the next customer to visit from the list of feasible locations and the storage capacity of the vehicle is updated before another customer is selected. The ant returns to the depot when the capacity constraint of the vehicle is met or when all customers are visited. The total distance L is computed as the objective function value for the complete route of the artificial ant. The ACO algorithm constructs a complete tour for the first ant prior to the second ant starting its tour. This continues until a predetermined number of ants m each construct a feasible route. Using ACO, each ant must construct a vehicle route that visits each customer. To select the next customer j, pseudo random probability equation is used.

3.3.1 The Savings Heuristic

This heuristic was proposed by [Clarke & Wright 1964] and improved by [Paessens 1988] [4]. It is the basis of most of the commercial software used to solve the vehicle routing problems in the industrial applications. The objective is to determine whether it is better to combine the clients v_i and v_j in the same route (when the value of γ_{ij} is big) or to put them in two different routes. The Savings value of the clients v_i and v_j is calculated as follows:

$$\gamma_{ij} = d_{i0} + d_{0j} - g.d_{ij} + f.\left|d_{i0} - d_{0j}\right| \qquad (3.13)$$

where

- d_{ij} represents the distance between the customer i and the customer j
- The index 0 corresponds to the depot
- f and g represent 2 parameters of the heuristic.

ACS with Saving Algorithm for CVRP		
1	Start	
2	Initialize	
3	For I ^{ma} iteration do:	
	(a)	For all ants generate a new solution
	(b)	Update the pheromone trails using Local updating
	(c)	Update the pheromone trails using Global updating
4	End	

Figure 3.1 ACS with Saving Algorithm for CVRP

3.3.2 Nearest Neighbor Heuristic

Nearest Neighbor Heuristic termed as Greedy Heuristic. Firstly, it needs to sort all distance between cities in ascending order. It construct a tour by repeatedly selecting the shortest distance and adding this route to the tour if it doesn't create a cycle with less than number of cities and will not allow to add the same route twice. Finally, it needs go to the start city.

3.4 Description of the Algorithm

In this section, construction of the vehicle routes and updating of the pheromone values are presented.

3.4.1 Construction of the Vehicle Routes

Initially in the ACS hybrid, m ants are positioned on all the graph vertices and a quantity of initial pheromone is applied on the arcs. Each ant takes its departure from the depot to visit the clients. Each client is visited once and only once by an ant, however, the depot can be visited several times. If the load stored by the ant exceeds the vehicle constraint capacity, the ant must return to the depot. We then get a complete route for a vehicle. When an ant goes back to the depot, it starts from scratch again. It initializes another route to visit other new clients. This operation is repeated over and over again until all clients are visited. This means that a solution to the Capacitated Vehicle Routing Problem (CVRP) has been found. During the process of building a route, the ant modifies the quantity of pheromone on the chosen arc by applying a local updating rule. Once all the ants are done with the building of their routing, the quantity of pheromone on the arcs belonging to the best routing found is updated according to the global updating rule.

The rule used for the construction of the routes is described hereafter. An ant k, positioned on a customer i, chooses the next customer j to visit by applying the following probabilistic rules.

$$p_{ij} = \arg \max_{u \in F_k(i)} \{(\tau_{ij})^{\alpha}.(\eta_{ij})^{\beta}.(\gamma_{ij})^{\lambda}\} \quad , \textit{if} \;\; q \le q_0 \tag{3.14}$$

$$p_{ij} = \frac{(\tau_{ij})^{\alpha}.(\eta_{ij})^{\beta}.(\gamma_{ij})^{\lambda}}{\sum_{u \in F_k(i)} (\tau_{ij})^{\alpha}.(\eta_{ij})^{\beta}.(\gamma_{ij})^{\lambda}} \quad ,\text{if} \;\; q \succ q_0 \tag{3.15}$$

Where,

- F_k (i) is the list of the customer not yet visited by the ant k positioned at the customer i,
- q is a random variable that follows a uniform distribution on [0,1],
- q0 is a parameter $(0 \le q \le 1)$ that determines the relative importance of the exploitation versus the exploration. Before an ant visits the next vertex, q is generated randomly. If q ≤ q0, the exploitation is then encouraged, otherwise the process of exploration is encouraged,
- τ ij is the quantity of pheromone associated to the arc (i , j),
- η ij is the heuristic of visibility that is the inverse of the distance between the customer i and j ,
- $\gamma_{ij} = d_{i0} + d_{0j} - g.d_{ij} + f.\left|d_{i0} - d_{0j}\right|$ is the heuristic of Savings where f and g are two parameters.

α, β and λ are three parameters that determine the relative importance of the pheromone, the distance and Savings respectively.

The Distances Between Customers are calculated by the Euclidean Distance Formula,

$$d_{ij} = \sqrt{(i_x - j_x)^2 + (i_y - j_y)^2} \tag{3.16}$$

Where,

- d_{ij} is the distance between customer i and j.

3.4.2 Updating Pheromone

Updating of the pheromone consists of local and global pheromone update.

3.4.2.1 Local Updating Rule

After all the artificial ants have improved the solutions through the heuristics, the pheromone trails will be updated. This is the main feature of an ACO algorithm which assists at improving future solutions since the updated pheromone trails would reflect the ants' performance and the quality of their solutions found.

While an ant is building its solution, the pheromone level on each arc (i, j) that is visited is updated according to the given local updating rule.

$$\tau_{ij}^{new} = (1-\rho) + \rho \,.\, \Delta\tau_{ij} \qquad (3.17)$$

3.4.2.2 Global Updating Rule

Once all ants have built their tours then the global updating rule is applied. In the ACS method only the globally best ant is allowed to deposit pheromone in an attempt to guide the search .

$$\tau_{ij}^{new} = (1-\rho)\tau_{ij}^{old} + \rho \,.\, \Delta\tau_{ij} \qquad (3.18)$$

In above local and global updating rule, ρ is the parameter of evaporation of the pheromone (0 <ρ<1). If the arc (i, j) is used by an ant whose solution is accepted, the quantity of pheromone is then increased on this arc by $\Delta \tau_{ij}$ that is equal to $1/L^*$ with L^* is the length of tours found by this ant.

CHAPTER 4

DESING AND IMPLEMENTATION OF THE SYSTEM

This chapter presents the design and implementation of the system conducted in this thesis.

4.1 Design of the System

In this system, Capacitated Vehicle Routing Problem is solved by using Ant Colony System and Saving Heuristic on Benchmark datasets.

The system calculates the processing steps: distance between customers using Euclidean Distance, the initial pheromone value by using Nearest Neighbor Heuristic method, the heuristic and saving value with the ACS algorithm parameters. After calculating the preprocessing steps, the system starts calculating the route construction step using probabilistic equation. For all the problems tested, the maximum iteration is set to 1000. While evaluating these steps, the system updates the local pheromone value. After all the customers have been served; finally, the system updates the global pheromone and generates result of the Capacitated Vehicle Routing Problem. The system flow is shown in Figure 4.1.

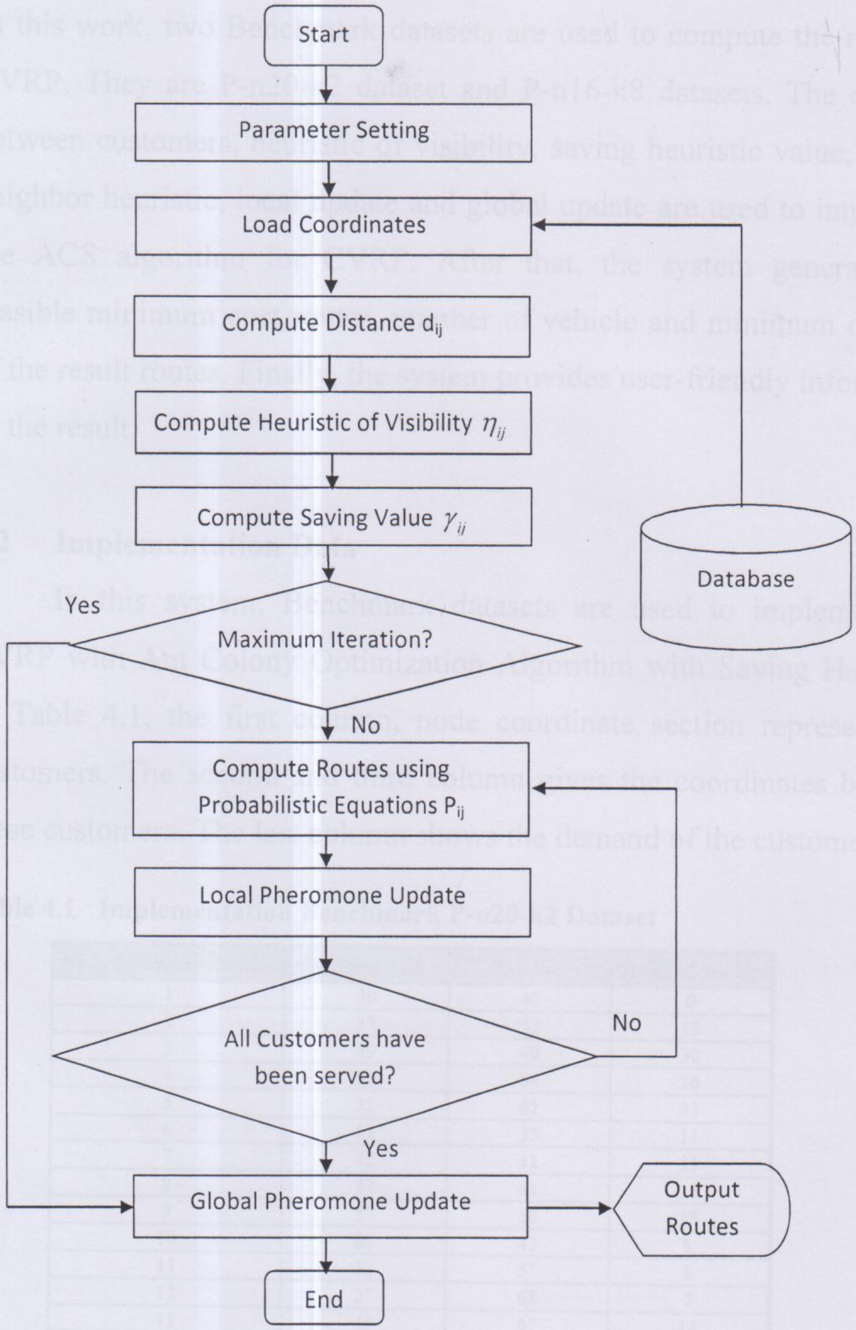

Figure 4.1 System Flow

In this work, two Benchmark datasets are used to compute the result of CVRP. They are P-n20-k2 dataset and P-n16-k8 datasets. The distance between customers, heuristic of visibility, saving heuristic value, nearest neighbor heuristic, local update and global update are used to implement the ACS algorithm for CVRP. After that, the system generates the feasible minimum cost routes, number of vehicle and minimum distance of the result routes. Finally, the system provides user-friendly information as the result.

4.2 Implementation Data

In this system, Benchmark datasets are used to implement the CVRP with Ant Colony Optimization Algorithm with Saving Heuristic. In Table 4.1, the first column, node coordinate section represents the customers. The second and third column gives the coordinates between these customers. The last column shows the demand of the customers.

Table 4.1 Implementation Benchmark P-n20-k2 Dataset

Node Coordinate Section	X Coordinate	Y Coordinagte	Demand Section
1	30	40	0
2	37	52	19
3	49	49	30
4	52	64	16
5	31	62	23
6	52	33	11
7	42	41	31
8	52	41	15
9	57	58	28
10	62	42	8
11	42	57	8
12	27	68	7
13	43	67	14
14	58	48	6
15	58	27	19
16	37	69	11
17	61	33	26
18	62	63	17
19	63	69	6
20	45	35	15

Table 4.2 Details Relative to the Instances

Instance Name	Number of Customer	Capacity of a Vehicle
P-n20-k2	20	160

4.2.1 Parameter Setting

According to the good value of ACS algorithm parameters of reference [1], the following parameter settings are set to get a good compromise to solution quality for the system.

- The number of ants m = n-1 initially placed at random on all the summits,
- $\alpha = 5$, $\beta = 5$, $\lambda=5$ with α, β and λ are three parameters that determine respectively the relative importance of the pheromone, of the distance and of Savings,
- $\rho = 0.1$ is the parameter of evaporation of the pheromone used in updating rules of the pheromones,
- f = g = 2 with f and g the two parameters of the heuristic of Savings,
- $q_0 = 0.9$ is the parameter that determines the relative importance of the exploitation versus the exploration. Before that an ant visits the next summit, q is generated randomly. If $q \le q_0$ the exploitation is then encouraged, otherwise it is the exploration process that is encouraged.

4.3 Construction of Vehicle Routes

The preprocessing steps of the system are calculation of distance d_{ij} between cities by using Euclidean distance formula, the initial pheromone value τ_0 by using Nearest Neighbor Heuristic method, the number of vehicle and the ACS algorithm parameters setting.

Table 4.3 The Pre-Processing Steps

Customer (i)	Customer (j)	Distance Value	Heuristic Value	Saving Value
1	2	13.8924	0.0719	13.8924
1	3	21.0238	0.0476	21.0238
1	4	32.5576	0.0307	32.5576
1	5	22.0227	0.0454	22.0227
…	…	…	…	…
20	19	38.4707	0.0259	39.0423

Table 4.3 shows the distance value d_{ij}, the heuristic value η_{ij} and saving value γ_{ij} between customers. The Euclidean distance formula (Equation 3.16) is used to calculate the distance between the customers. For calculating the distance, the system uses the coordinates between the customers in Table 4.1. In this step, the distance d_{12}, d_{13}, d_{14},…, $d_{20,18}$, $d_{20,19}$ are calculated as follows.

$$d_{ij} = \sqrt{(i_x - j_x)^2 + (i_y - j_y)^2}$$

$$d_{12} = \sqrt{(30-37)^2 + (40-52)^2} = 13.8924$$

$$d_{13} = \sqrt{(30-49)^2 + (40-49)^2} = 21.0238$$

…

…

…

$$d_{24} = \sqrt{(37-52)^2 + (52-64)^2} = 19.2093$$

Figure 4.2 Calculating Distance between Customers

After calculating the distances, the heuristic value η_{ij} is calculated by using these distances for indicating how promising the choice of the next customer j is from current customer i. In this step, the system calculates the heuristic values $\eta_{12}, \eta_{13}, \eta_{14}, \ldots, \eta_{20,19}$.

$$\eta_{ij} = 1/d_{ij}$$

$$\eta_{12} = 1/d_{12} = 1/13.892 = 0.0719$$
$$\eta_{13} = 1/d_{13} = 1/21.024 = 0.0475$$
…
…
…
$$\eta_{45} = 1/d_{45} = 1/21.095 = 0.0474$$

Figure 4.3 Calculating Heuristic Values

By using the distances in (Figure 4.2), saving value γ_{ij} is calculated by inserting these distance values to the Equation (3.13). In this step, we can yield the result of saving value $\gamma_{12}, \gamma_{13}, \gamma_{14}, .., \gamma_{20,18}$.

$$\gamma_{ij} = d_{i0} + d_{0j} - g.d_{ij} + f.|d_{i0} - d_{0j}|$$

$$\gamma_{12} = d_{10} + d_{02} - g.d_{12} + f|d_{10} - d_{02}|$$
$$= d_{11} + d_{12} - 2.d_{12} + 2|d_{11} - d_{12}|$$
$$= 0 + 13.892 - 2(13.892) + 2|0 - 13.892| = 13.892$$
…
…
…
$$\gamma_{45} = d_{40} + d_{05} - g.d_{45} + f|d_{40} - d_{05}|$$
$$= d_{41} + d_{15} - 2.d_{45} + 2|d_{41} - d_{15}|$$
$$= 32.5576 + 22.0227 - 2(21.0950) + 2|32.5576 - 22.0227|$$
$$= 33.4601$$

Figure 4.4 Calculating Saving Value

4.3.2 Implementation Result For P-n20-K2 Dataset

The implementation of the system results are shown in this section.

Table 4.4 Implementation Result for CVRP Problem

Instance Name	No. of Vehicle	Minimum Distance	No. of Route
P-n20-k2	2	219.587	2

The table 4.4 shows the CVRP results depends on the number of customer, the demand need by these customer and the vehicle's capacity that travels. In the first instance P-n20-k2, the vehicles with capacity Q=160 are served to (20) customers with 2 vehicles and the minimum distance is (219.587). In the second instance P-n16-k8, the vehicles with capacity Q=35 are served to 16 customers with (8) vehicles and the minimum distance is (450.418).

4.4 Implementation

When the program is started to run, the main window form of the system appears as shown in Figure 4.5.

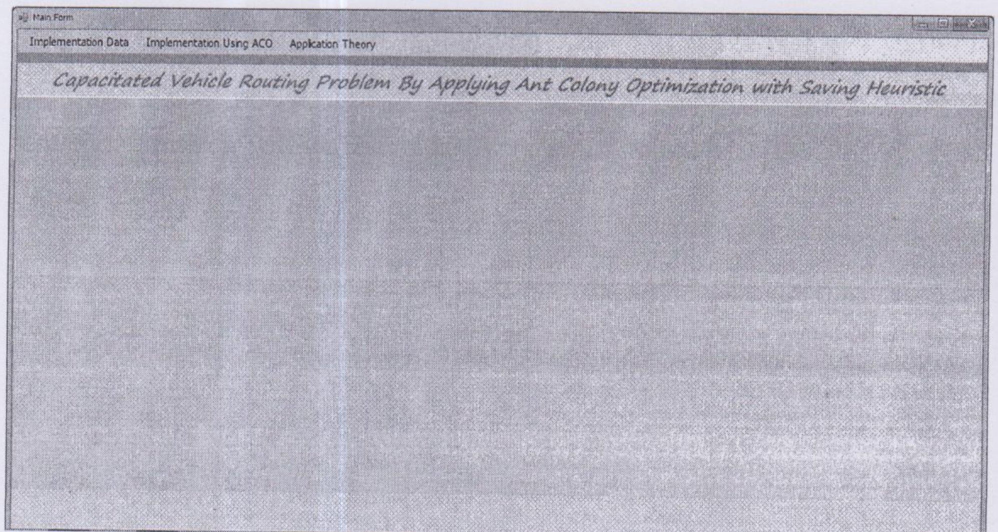

Figure 4.5 Main Window Form of the System

Firstly, user can start by clicking "Implementation Data" menu in Figure 4.5 to view the datasets information and the parameter settings. If user ckicks the "Implementation Data", the system shows the Implementation Data Information as shown in Figure 4.6.

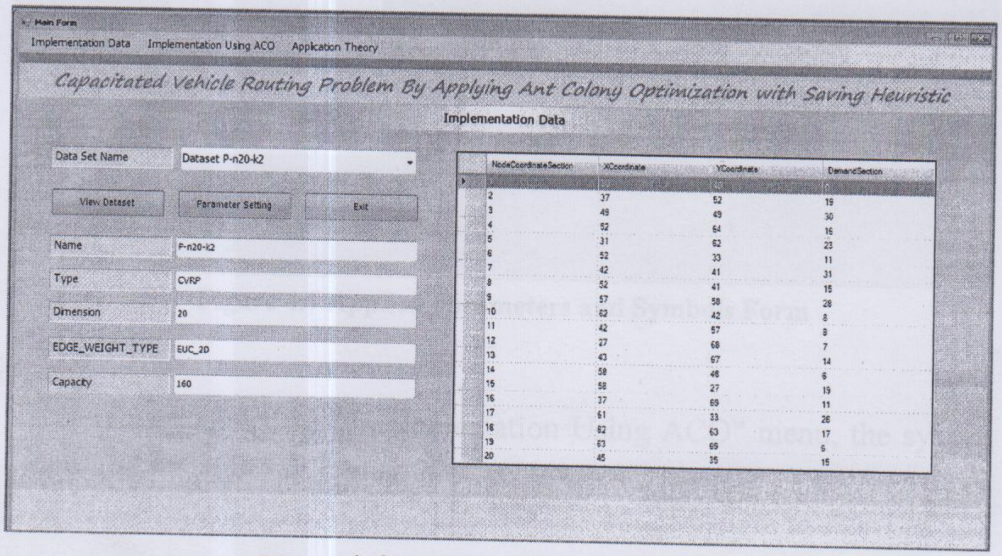

Figure 4.6 Implementation Data Form

If user clicks the "View Dataset" button, the system shows the dataset information in the grid view. And if user clicks the "Parameter Setting" button, the system shows the Applied Parameters and Symbols Form as shown in Figure 4.7. If user clicks the "Exit" button, the system leaves the current active form.

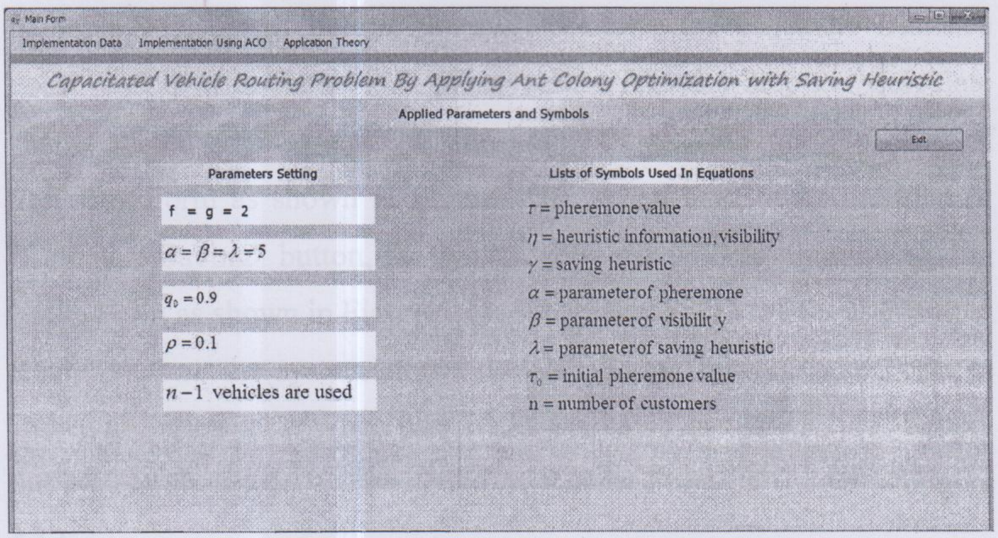

Figure 4.7 Applied Parameters and Symbols Form

If user clicks the "Implementation Using ACO" menu, the system shows the Implementation Steps Form as shown in Figure 4.8.

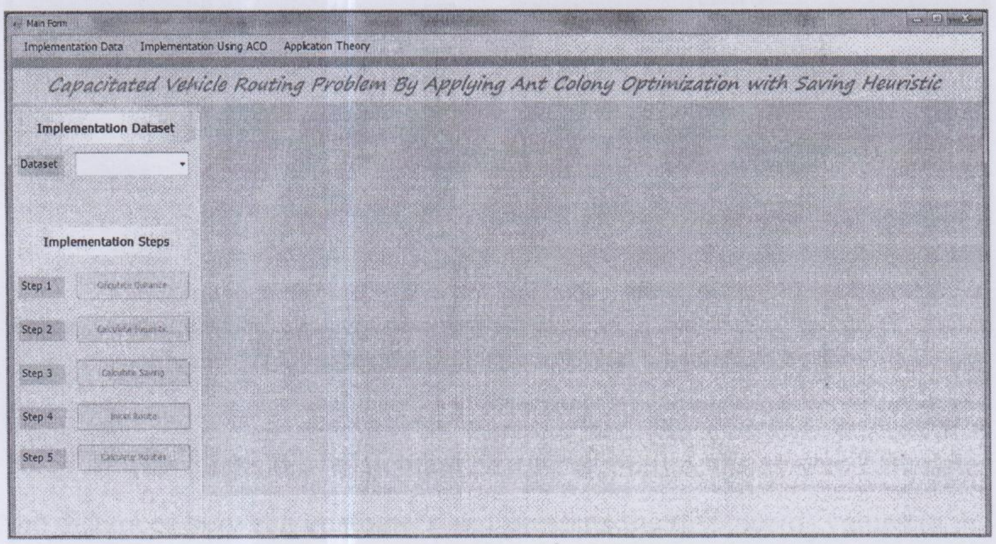

Figure 4.8 Implementation Steps Form

To start implementing the system, user must choose one dataset in Figure 4.8. After choosing the dataset, user can start calculating the implementation steps from Step 1 to Step 5. When user clicks the Step 1:

"Calculate Distance" button, the system shows the Calculate Distance Form as shown in Figure 4.9. When user clicks the Step 2: "Calculate Heuristic" button, the system shows the Computing the Heuristic of Visibility Form as shown in Figure 4.10. When user clicks the Step 3: "Calculate Saving" button, the system shows the Computing the Saving Value Form as shown in Figure 4.11. When user clicks the Step 4: "Initial Route" button, the system shows the Calculating Initial Route Form as shown in Figure 4.12. When user clicks Step 5: "Calculate Route" button, the system shows the Construction of Routes Form as shown in Figure 4.13.

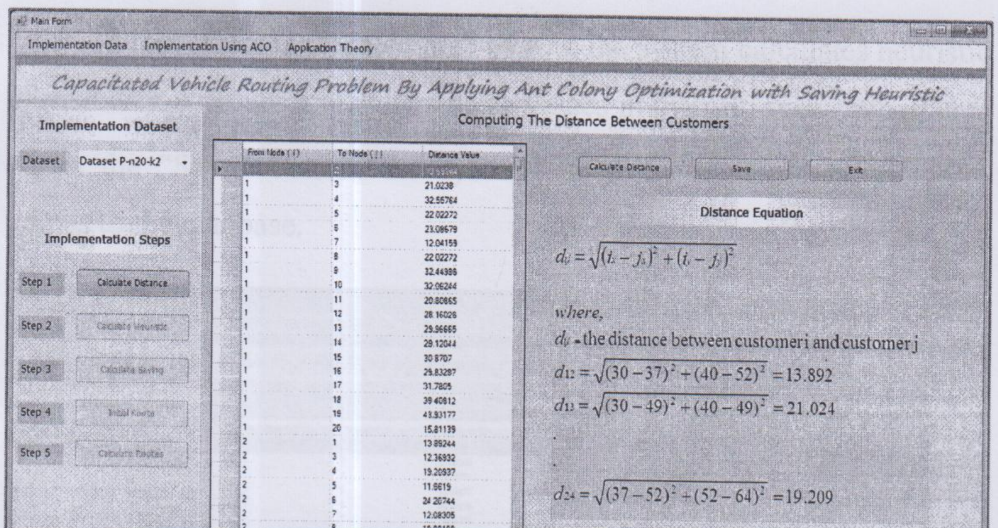

Figure 4.9 Computing Distance between Customers Form

If user clicks the "Calculate Distance" button, the system calculates the distances between customers and presents the distance d_{ij} values in the grid view as shown in Figure 4.10. If user clicks the "Save" button, the system saves the distance values to the database.

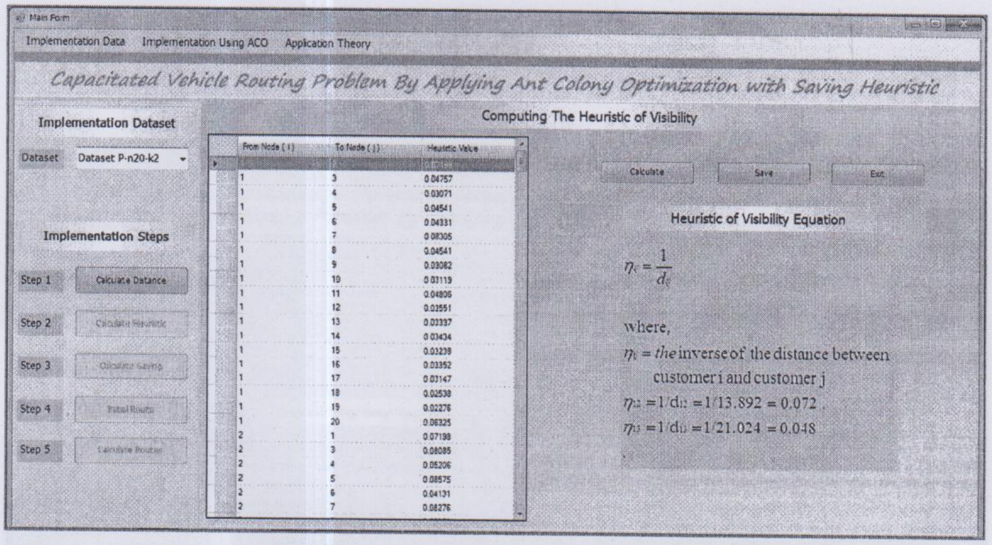

Figure 4.10 Computing the Heuristic of Visibility Form

If user clicks the “Calculate” button, the system calculates heuristic of visibility η_{ij} values and shows these values in the grid view. When user clicks the “Save” button, the system saves the heuristic of visibility values to the database.

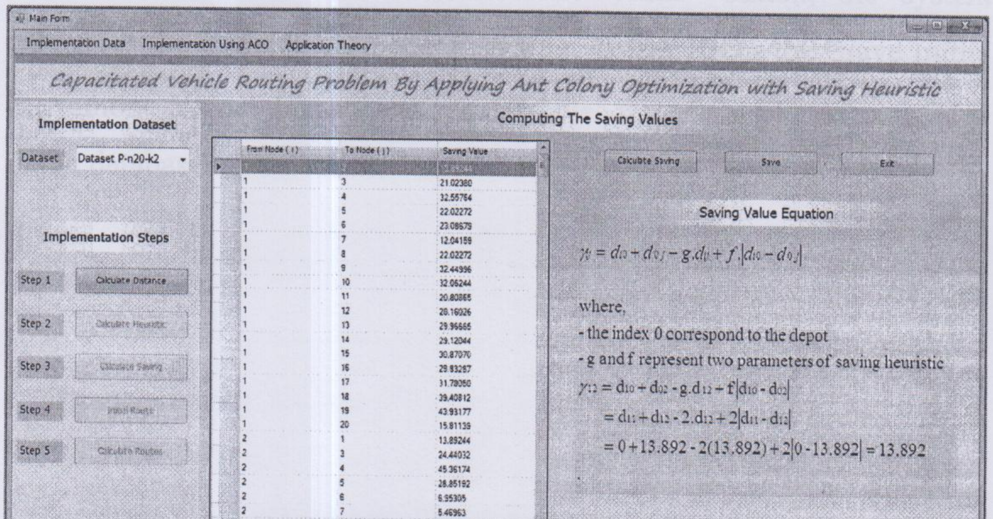

Figure 4.11 Computing the Saving Values

If user clicks the “Calculate Saving” button, the system calculates saving γ_{ij} values and shows these values in the grid view. When user clicks the "Save" button, the system saves the saving values to the database.

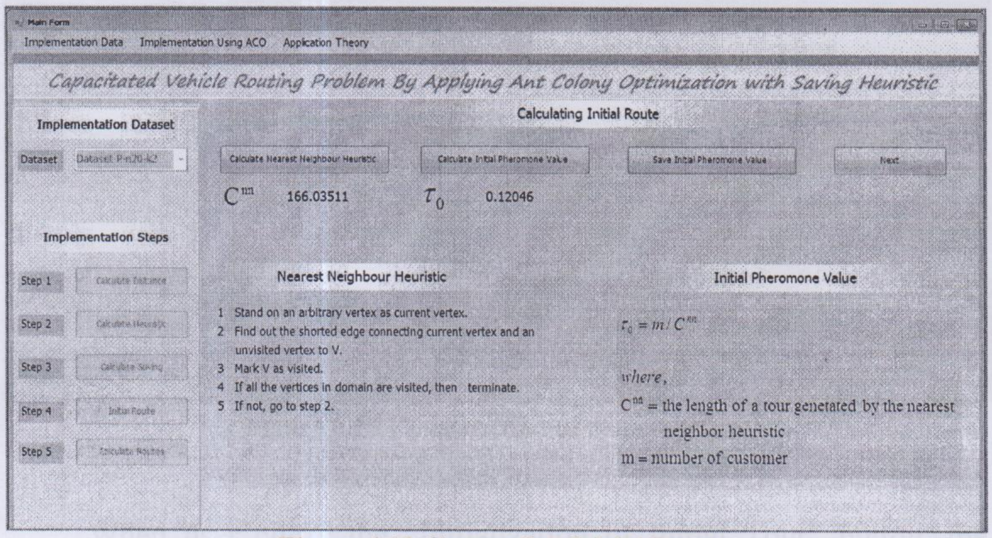

Figure 4.12 Computing Initial Route

If user clicks the "Calculate Nearest Neighbor Heuristic" button, the system calculates the nearest neighbor heuristic C^{nn} value. And, if user clicks the "Calculate Initial Pheromone Value" button, the system calculates the initial pheromone τ_0 value. If user clicks the "Save Initial Pheromone Value" button, the system saves the initial pheromone value to the database.

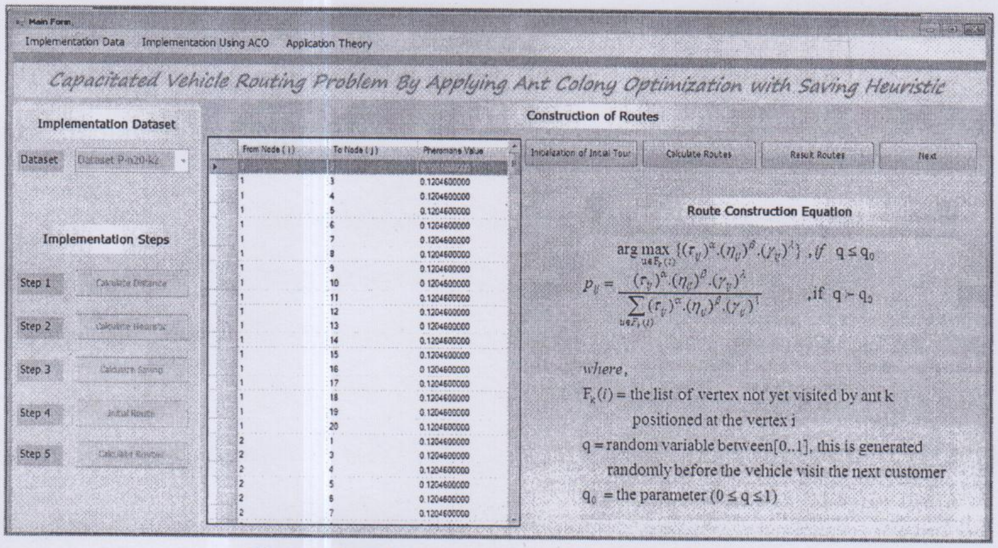

Figure 4.13 Construction of Routes Form

When user clicks the "Initialization of Initial Tour" button, the system initialize the initial tour to all routes and shows these data in the grid view. When user click the "Calculate Route" button, the system evaluates the routes. After that, when user clicks the "Result Routes" button, the system show the Result Route From as shown in Figure 4.14.

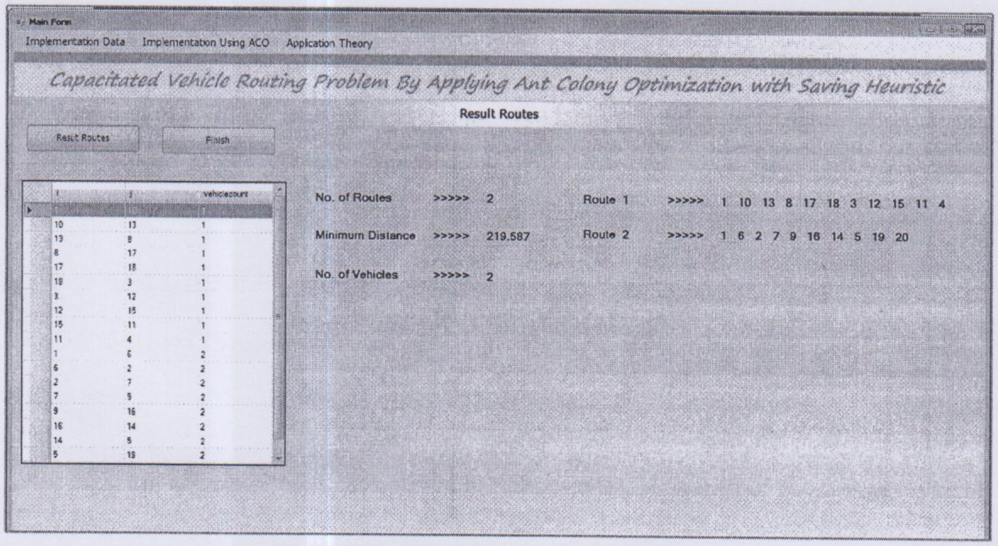

Figure 4.14 Showing Result Routes Form

In Figure 4.14, the system shows the result of the system. In result, the number of routes, minimum distance, number of vehicle used and routes are display as a result.

CHAPTER 5

CONCLUSION

CVRP is the process of distribution of goods from depot to geographically scatter customers by a fleet of vehicles. The goal of the thesis is to find the minimum cost routes on Capacitated Vehicle Routing Problem by applying Ant Colony Optimization. To achieve this goal, ACO is implemented under C# programming language on Microsoft Visual Studio using two Benchmark instance datasets.

The Capacitated Vehicle Routing Problem (CVRP) is a well-known combinational optimization problem, which is concerned with the distribution of goods between the depot and customers. This system has presented solving the CVRP problem that a fleet of capacitated vehicle has to deliver goods to geographically distributed customers with variable demand. The system is applied to two Benchmark (P-n20-k2 and P-n16-k8) instance of CVRP and the results obtained the minimum cost routes and minimum number of vehicles. By applying the ACS algorithm for CVRP, it is found that it avoids long convergence time by directly concentrating on the search in a neighborhood of the best algorithm. So, ACS with Saving heuristic has the capability to tackle the CVRP with satisfactory solution quality and run time.

5.1 Limitations

The Limitations of the thesis are as follow:

- This system has considered only on two Benchmark Datasets.
- This system provides the only static vehicle routing problem with known demands.

5.2 Further Extension

Future work will be conducted to improve the system. Alternative formulas will be examined to enhance the method of update and pseudo-random decision of ants for selecting the next node in solution construction. It is possible to extend the static vehicle routing problem to dynamic routing problem with run time demand. And additional improvement might lie on the combination of this system with the other metaheuristics like genetic algorithm.

REFERENCES

[1] A.E. Rizzoli, F. Oliverio, R. Montemanni, L.M. Gambardella1,

Ant Colony Optimization for Vehicle Routing Problems: From Theory to Applications

Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA) Galleria 2, CH-6928 Manno, Switzerland{andrea, roberto, luca}@idsia.ch AntOptima, via Fusoni 4, CH-6900 Lugano, Switzerland fabrizio.oliverio@antoptima.ch

[2] A.H.E. Hassani, L. Bouhafs and A. Koukam

A Hybrid Ant Colony System Approach for the Capacitated Vehicle Routing Problem and the Capacitated Vehicle Routing Problem with Time Window

Systems and Transportation Laboratory University of Technology of Belfort-Montbliard, France

[3] A. Ezzatneshan

An algorithm for the Vehicle Problem

International Journal of Advanced Robotic Systems, Vol. 7, No. 2 (2010) ISSN 1729-8806, pp. 125-132

[4] H.S. Lopes, V.L.D. Molle, C.R.E. Lima

An Ant Colony Optimization System for the Capacitated Vehicle Routing Problem

CILAMCE 2005 – ABMEC & AMC, Guarapari, Esp´ ırito Santo, Brazil, 19th – 21st October 200

[5] J.E. Bell, P.R. McMullen

Ant colony optimization techniques for the vehicle routing problem

J.E. Bell, P.R. McMullen/Advanced Engineering Informatics 18 (2004) 41-48

[6] L. Bouhafs, A. Hajjam, A. Koukam

A Hybrid Heuristic Approach to Solve the Capacitated Vehicle Routing Problem

Journal of Artificial Intelligence: Theory and Application (Vol.1-2010/Iss.1) Bouhafs et al. / A Hybrid Heuristic Approach to Solve the Capacitated ... / pp. 31-34

[7] M. Dorigo, T. Stutzle

Ant Colony Optimization

The MIT Press, 2004

Cambridge, Massachusetts,London, England

[8] M. Dorigo

Ant Colony Optimization for the Vehicle Routing Problem

[9] L.S. Lee, Z.A. Majid and H.V. Seow

Ant Colony Optimization for Capacitated Vehicle Routing Problem

Journal of Computer Science 8 (6): 846-852, 2012

[10] T. Caric and H. Gold

Vehicle Routing Problem

ISBN 978-953-7619-09-1 ISBN 978-953-7619-09-1, pp. 142, September 2008, I-Tech, Vienna, Austria

[11] V. Maniezzo, L.M. Gambardella, F. Luigi

Ant Colony Optimization

Future & Emerging Technologies unit of the European Commission through Project BISON (IST-2001-38923)

[12] E. Bonabeau, M. Dorigo and G.Theraulaz

Swarm Intelligence: From Natural to Artifical Systems

New York, NY: Oxford University Press, 1999.