

HEURISTICS FOR ENERGY EFFICIENT VEHICLE ROUTING PROBLEM

A Thesis by

Majid Talaei

Bachelor of Science, Azad University, 2009

Submitted to the Department of Industrial and Manufacturing Engineering
and the faculty of the graduate school of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

December 2012

© Copyright 2012 by Majid Talaei

All Rights Reserved

HEURISTICS FOR ENERGY EFFICIENT VEHICLE ROUTING PROBLEM

The following faculty members have examined the final copy of this thesis for form and content, and recommended that it be accepted in partial fulfillment of the requirements for degree of Master of Science with a major in Industrial Engineering.

Krishna K. Krishnan, Committee Chair

Mehmet Bayram Yildirim, Committee Co-Chair

Ramazan Asmatulu, Committee Member

DEDICATION

To my loving parents, brother and family members

ACKNOWLEDGEMENTS

Foremost I would like to thank my advisor Dr. Krishna Krishnan for all his support, patience and immense knowledge throughout my master's study at Wichita State University. This thesis would not have been complete without his continued support, encouragement, kindness and sound advice. I could have not imagined having a better advisor than him for my master study. I would like to thank Dr. Bayram Yildirim for his inputs, directions and insightful comments. My sincere thanks also go to Dr. Ramazan Asmatulu for his precious time and valuable comments.

I would like to thank all my dear friends for their support specially Dr. Shokoufeh Mirzaei who always helped me with my questions and provided her own work and MATLAB code to me generously.

Finally, I would like to thank my parents and family for their perfect support and love, as well as their encouragement. Their kindness to me will be remembered forever.

ABSTRACT

US logistics cost of 1.397 trillion dollars in 2007, which stands for more than 10 percent of the total GDP of the country justifies any attempt of reducing it. Transportation followed by inventory-carrying and logistics administration has the greatest cost share in logistics. A tool which is very critical in transportation planning and can contribute to huge savings if used properly is Vehicle Routing Problem. Near optimum vehicle routes which are designed by outstanding heuristics and experts could contribute significantly to cost saving.

Another important issue which directly affects logistics and transportation is energy consumption. Energy consumption and energy saving plans are hot topics everywhere nowadays. Issues such as green house effect, global warming effect and oil resources termination are great global concerns.

This research tries to modify vehicle routing problem heuristics and make them sensitive to the issue of energy consumption. Traditional VRP heuristics and solution methods have tried to minimize total distance traveled of vehicles as the main objective function, while energy consumption minimization is the objective function of energy efficient VRP heuristics in this research. Two heuristics are modified in an “Energy Efficient” manner, nearest neighbor algorithm and saving algorithm. The proposed heuristics are examined with several benchmark problems from literature and are found to be efficient and effective both in terms of total distance travelled and energy consumption.

TABLE OF CONTENTS

Chapter	Page
1. INTRODUCTION	1
1.1 Motivation and Background	1
1.2 Overview of the Vehicle Routing Problem.....	3
1.3 Energy Consumption Consideration in VRP	6
1.4 Research Focus and Objectives	8
2. LITERATURE REVIEW	9
2.1 Introduction.....	9
2.2 Classical heuristics.....	9
2.3 Modern heuristics.....	15
2.4 Energy and VRP	17
3. METHOD	19
3.1 Introduction.....	19
3.2 Nearest neighbor algorithm.....	20
3.3 Energy efficient nearest neighbor (EENN) algorithm for CVRP	22
3.4 Saving Algorithm.....	26
3.5 Energy Efficient Clarke Wright Method for VRP (ECW) (Mirzaei and Krishnan, (2011)).....	26
3.6 Improved Energy Efficient Clarke Wright Method for VRP (Improved ECW)	28
4. Computational Results.....	30
4.1 Computational results of EENN for CVRP	30
4.2 Computational Results for Improved ECW	36
5. Conclusion and Future Work.....	40
5.1 Comparison of 5 Algorithms	40
5.2 Conclusion and future work.....	43
REFERENCES	45
APPENDICES	49

TABLE OF CONTENTS (CONT.)

Chapter	Page
A. MATLAB CODE FOR NN ALGORITHM	50
B. MATLAB CODE FOR EENN ALGORITHM.....	52
C. MATLAB CODE FOR IMPROVED ECW	55

LIST OF FIGURES

Figure	Page
1. U.S. business logistics cost of year 2010.....	2
2. A typical VRP solution with 4 vehicle routes.....	4
3. U.S. fuel price trends between years 2000 and 2011	7
4. two potential pints, p_1 and p_2 , for route extension (Gaskell, 1967)	10
5. the effect of demand consideration in selection of customers with equal saving values (Altinel and Oncan, 2005)	13
6. GA parents for VRP (Baker and Ayechew, 2003).....	15
7. GA offspring for VRP (Baker and Ayechew, 2003)	16
8. nearest neighbor algorithm node selection criteria	22
9. EENN node selection criteria	23
10. ECW algorithm (Mirzaei and Krishnan (2011)).....	28
11. EENN algorithm improvement over NN algorithm	34
12. Distance and energy trends of Problem 1, Vehicle 1	35
13. Distance and energy trends of Problem 5, Vehicle 2.....	36
14. Problem 25 solved with Improved ECW algorithm for different alpha values	37
15. Improvement percentages of Improved ECW over ECW	39
16. Comparison of five algorithms in terms of energy consumption	42
17. Comparison of 5 algorithms in terms of total distance travelled.....	43

CHAPTER 1

INTRODUCTION

1.1 Motivation and Background

Managing the relationship between suppliers, manufacturers, customers and generally supply chains have been always a great concern for many companies. Supply chain management (SCM) are built to manage this relationship in the most effective and efficient way. As defined by the council of supply chain management professionals (CSCMP), “Supply chain management encompasses the planning and the management of all activities involved in sourcing and procurement, conversion, and all logistics management activities”. They further argue that SCM tries to build a coordination and collaboration among channel partners, which can be suppliers, intermediaries, third-party service providers, and customers. SCM is capable of creating a synergy by intra- and intercompany integration and management (Lambert & Cooper, 2000). These definitions imply a distinction between logistics and SCM; if one considers the logistics as managing the intercompany affairs, SCM can be viewed as the extension of logistics outside the company to include suppliers and customers. However Cooper, Lambert, and Pagh (1997) discuss that the concept of SCM as the integration of logistics across the supply chain has been slightly revised lately to the integration and management of key business processes across the supply chain. On the other hand, Lambert, Cooper and Pagh (1998) argue that supply chain management can be classified into two main categories: planning and execution. Supply chain planning (SCP) includes activities such as forecasting materials required for production, production and distribution planning and so on, while supply chain execution (SCE) embraces the activities related to the real implementation of supply chain plan, like production and stuck

control, warehouse management, transportation and delivery (as cited in Giaglis, Minis, Tatarakis, & Zeimpekis, 2004, p. 749). SCP has been more interesting to researchers due to its critical role on cost effectiveness, strategic planning and competitiveness opportunities that it can bring to the company (Giaglis et al. 2004). However, there are still many opportunities in the area of SCE to improve, especially in the area of logistics and distribution management. The costs that logistics imposed to a company can be divided into three main categories: logistics administration, inventory-carrying costs and transportation costs. Based on the “CSCMP’s 22nd annual state of logistics report”¹, transportation accounts for 63% of the U.S. business logistics cost for the year 2010. Figure 1 shows the share of other logistics cost’s components.

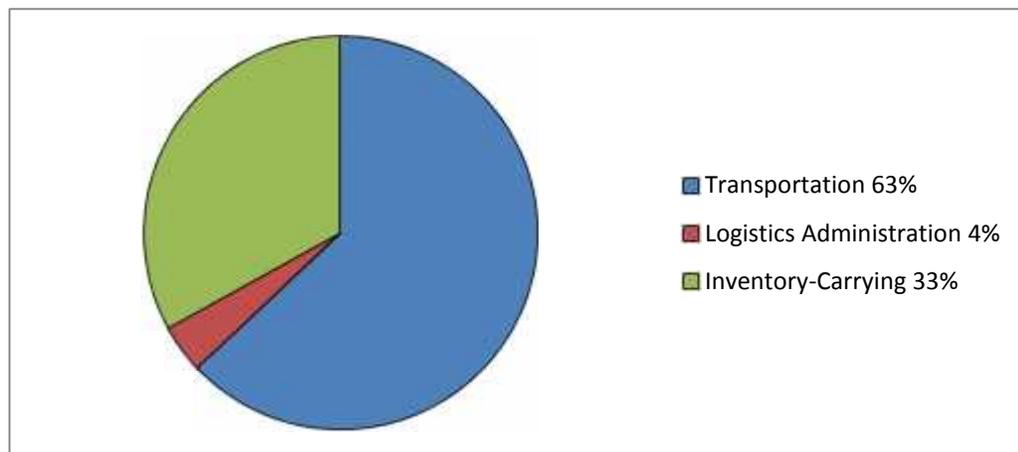


Figure 1: U.S. business logistics cost of year 2010

In this report, the total U.S. business logistics cost is reported \$1.2 trillion for 2010, a share of \$760 billion for transportation costs. This sheds light on the fact that even very small percentages of improvement of logistics can save millions of dollars for the country. Potential areas of logistics that have the capability of cost improvement should be targeted.

¹ www.cscmp.org

1.2 Overview of the Vehicle Routing Problem

Vehicle Routing Problem (VRP) is the backbone in distribution management and physical distribution (Laporte, 1992a; Ghiani, Guerriero, Laporte, & Musmanno, 2003). It can be described as the “problem of designing optimal delivery or collection routes from one or several depots to a number of geographically scattered cities or customers, subject to side constraints” (Laporte, 1992a). VRPs are combinatorial optimization problems (NP-hard). Optimization problems can be divided into two categories: problems with continuous variables and problems with discrete variables, which are called combinatorial. In combinatorial problems, the goal is to find the best solution among a set of finite solutions (Papadimitriou & Steiglitz, 1998). Some examples of combinatorial optimization problems are: integer programming, vehicle routing problem, traveling salesman problem, etc. Mathematical definition of VRP is as follows:

Let $G = (V, A)$ be a graph, where $V = \{1, \dots, n\}$ is a set of vertices (nodes) representing cities, where *depot* is located at node 1, and A is the set of arcs (edges). With every arc (i, j) $i \neq j$ there is a corresponding non-negative distance matrix $C = (c_{ij})$. In some cases, c_{ij} can be interpreted as travel cost or travel time between nodes i and j . When C is symmetrical (travel cost of node i to j is equal to travel cost of node j to i), it is convenient to consider the set of arcs as a set E of undirected arcs. Furthermore, assume there are m vehicles available at depot to service the nodes (customers), where $m_L \leq m \leq m_U$. When $m_L = m_U$, number of vehicles (m) is said to be fixed. When $m_L = 1$ and $m_U = n-1$, m is said to be free. When m is not fixed, it is logical to consider a fixed cost associated with use of a vehicle but usually for the sake of simplicity, this cost is ignored. All vehicles are considered to be identical and have the fixed capacity D . The VRP is to design vehicle routes with least cost in such a way that:

(i) each city in $V \setminus \{1\}$ is visited only once and once by exactly one vehicle;

- (ii) depot is the origin and the destination of all routes;
- (iii) some side constraints are satisfied.(Laporte, 1992a)

A typical VRP solution is showed in Figure 2. As illustrated in this figure, nodes (cities or customers) are scattered around depot and 4 vehicle routes starting and ending at depot are designed to serve all the customers.

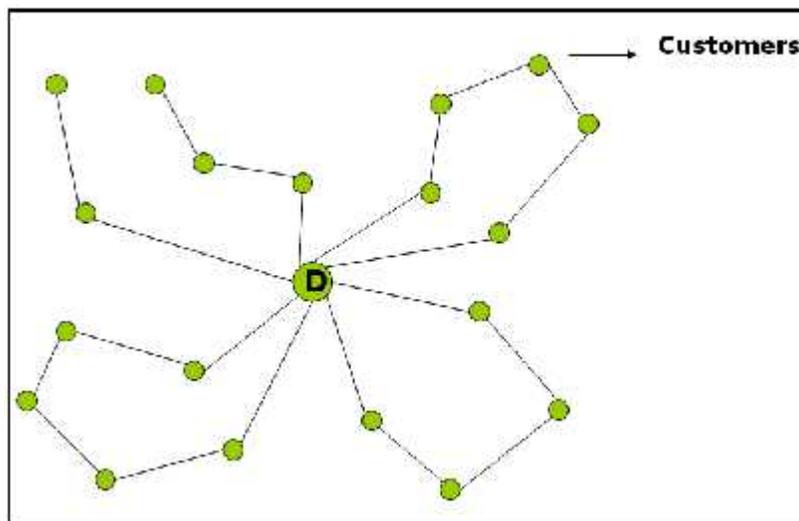


Figure 2: A typical VRP solution with 4 vehicle routes

Different classes of VRPs have been developed to model the problems faced in of real world. Each of VRP categories answers to the specific needs of customers, logistics and distribution departments or both of them. The categories of research in VRP are given below:

- Capacitated VRPs (CVRPs): in this category, there are a number of identical vehicles all of which have a limited capacity. In this, the side constraint is that the sum of demands of all the cities (customers) in a route must not exceed the vehicle capacity.
- Distance-constrained VRPs (DVRPs): In this category, total time or distance of a vehicle route must not exceed a predetermined bound.

- VRPs with Time Windows (VRPTW): In this category, which is an extension of CVRPs, in addition to the capacity constraint for vehicles, each customer i should be served within a time interval $[a_i, b_i]$ and in case of early arrive of the vehicle, waiting times are allowed in city i up to time a_i .
- VRPs with Backhauls (VRPBs): This is an extension of CVRP in which the customer set $V \setminus \{1\}$ is divided into two subsets. The first subset of customers, called *Linehaul customers*, need to be supplied with a specific number of deliveries, d_i , while the second subset of customers, called *backhaul customers*, need vehicles to pick up some inbound products. In all the routes in VRPBs, if both linehaul and backhaul customers are present, linehaul customer must be served before backhaul customers (Toth & Viegó, 2002).
- VRPs with Pick up and Deliveries (VRPPD): Each customer i has some quantity of homogeneous products both to be delivered and picked up which is represented as d_i and p_i respectively. For each customer i , there are nodes as origin of delivery demand and nodes as destination of pick up demand represented with Q_i and D_i respectively (Toth & Viegó, 2002).
- Heterogeneous Fleet Vehicle Routing Problem (HVRP) (Gendreau, Laporte, Musaraganyi, Taillard, 1999): Unlike the classical vehicle routing problem in which all vehicles are identical, customers in HVRP are served by a fleet of heterogeneous vehicles. Each type of vehicles has their own fixed cost, capacity, and variable cost. The goal therefore is to determine the best fleet composition that yields the least overall cost. However Golden, Raghavan, and Wasil (2008) argue that basic assumptions in HVRP are modified in the literature as follows:
 - (i) There are unlimited numbers of vehicles for each vehicle type

- (ii) Fixed costs of all vehicle types are ignored
- (iii) The routing costs are *vehicle – independent*, i.e. for any arc ij , c_{ij} is the same for all vehicle types.

They further categorize HVRP to several subgroups. One of these subgroups which the route costs are assumed to be vehicle dependent (removing the third assumption) is called “Heterogeneous VRP with Vehicle Dependent Routing Costs (HVRPD)”.

1.3 Energy Consumption Consideration in VRP

As energy costs have risen, energy savings are currently being researched extensively. Estimated dates for the termination of petroleum resources have made many governments to initiate energy efficient plans and programs for alternative resources of energy. The European Union has set itself a 20% energy saving objective by the year 2020². On the other hand rather than just energy saving, the green house effect is another global concern which is very sensitive to the consumption of a big class of energy, fossil fuel burning. Vehicle fuel consumption, specifically related to this research, seems to be a good candidate for both energy efficiency plans and emission reduction plans. Currently, the US Department of Energy funds R&D to develop energy efficient and environmentally friendly vehicle technologies³. Furthermore, recent rising and volatile trends of fuel prices as shown in Figure 3 have made companies more enthusiastic about energy efficient transportation plans. Gusikhin, Macneille, and Cohn, (2010) argue that currently most of the commercial packages and methods for VRP minimize traveled distance as the main objective of the problem. They mention that although there is a correlation between the distance traveled and fuel consumption, it is not a perfect correlation and fuel efficiency along with emissions are related also to driving conditions like highway or city driving.

² http://ec.europa.eu/research/energy/eu/research/efficiency/index_en.htm

³ <http://www.eere.energy.gov/topics/vehicles.html>

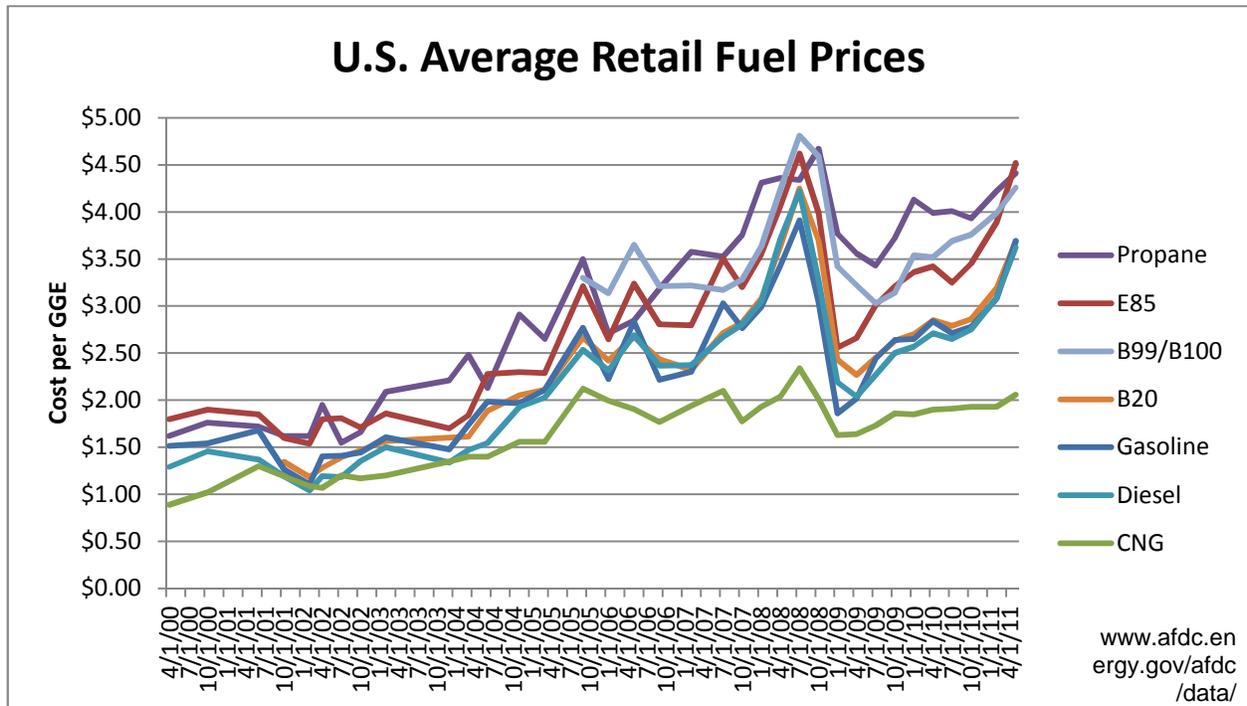


Figure 3:U.S. fuel price trends between years 2000 and 2011

They further argue that since fuel efficiency and emissions vary significantly among different types of vehicles, it is necessary to focus on a variant of Heterogeneous VRP with Vehicle Dependent Routing Costs (HVRPD) which is previously introduced in section 1.2, and call it “The Vehicle Routing Problem to Minimize Mixed-Fleet Fuel Consumption and Environmental Impact (VRPMF)”. So having different types of cars with different highway and city MPGs, the vehicle routes and delivery tasks assigned to the vehicle types are designed in a way to minimize the total fuel consumption. In another research conducted by I. Kara, B. Kara, and Yetis (2007) they define a new cost objective for VRP which is “load based”, i.e. besides the distance between two cities in VRP, they also consider the load of the vehicle and call this “Energy Minimizing Vehicle Routing Problem (EMVRP)”. They argue that the work a vehicle does along an arc (i, j) can be calculated as:

$$\text{Work} = \text{weight of vehicle over link (i, j)} * \text{distance of link (i, j)} \quad (1)$$

Weight of a vehicle over link (i, j) is obviously constituted from tare weight of the vehicle plus loads weight of the vehicle over link (i, j). “Since work is energy, minimizing the total work done is equivalent to minimizing total energy used.” (I. Kara, B. Kara, and Yetis, 2007). The authors claim that this is the first time in the literature “vehicle load” is considered in an objective function of VRP. They use an exact method (integer programming) to solve this “load based” VRP. However, the NP-hard nature of VRP necessitates the presence of heuristic methods for EMVRP, just like the rich literature on classical VRP.

1.4 Research Focus and Objectives

The main objective of this research is to focus on generating energy efficient capacitated vehicle routing problem heuristics. This research takes inspiration from the work done by I. Kara, B. Kara, and Yetis (2007) to consider “load based” cost objective function for vehicle routing problem. Two heuristics are modified to develop “energy efficient” heuristics for the VRP problem. The first heuristic that is modified is the nearest neighbor method which is very efficient in terms of computational time and simplicity while producing an average quality solution. The second heuristic is the saving algorithm which generates a very high quality solution while having an average computational time. The energy efficient saving algorithm is recently published in a paper by Mirzaei and Krishnan (2011). However the proposed method is an improvement of the heuristic.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The main focus of this chapter is to review different heuristic methods for VRP. Heuristics for VRP are classified into two main groups. The first group is classical heuristics which are through the initial methods that were generated to solve VRP and some of them are still being used widely due to their good quality solution and computational time. The saving algorithm is probably the most famous algorithm which belongs to this group. The second group is modern heuristics or metaheuristics. Most famous Metaheuristics which are used to solve VRP are tabu search and simulated annealing. These two former classes of heuristics are created specifically for VRP. However as it is mentioned by Laporte (1992a), TSP procedures such as nearest neighbor algorithm, insertion algorithms and tour improvement procedures could be all the time used for specially Capacitated VRPs and distance-constrained VRPs without any modifications. For a review over main TSP algorithms refer to Laporte (1992b).

The last part of this chapter will review the literature relevant to issue of energy in the VRP. For a review of exact and approximate methods of VRP refer to Laporte (1992a). For a taxonomic review refer to Eksioglu, VolkanVural & Reisman (2009).

2.2 Classical heuristics

2.2.1 Saving algorithm

Clarke and Wright saving algorithm published in Clarke and Wright (1964) is one of the earliest methods to solve CVRP, but still is widely being used in commercial packages. This method is based on calculating saving values for all possible pairs of customers:

$$S_{ij} = c_{i1} + c_{1j} - c_{ij}, \tag{2}$$

Where c_{i1} is the distance of customer i from depot, c_{1j} is the distance of customer j from depot and c_{ij} is the distance of customers i and j from each other. Initially it is assumed that one vehicle is assigned to each customer and there is a simple depot-single customer-depot route for all of the customers. These routes are merged together based on saving values. This merge could be done in either sequential or parallel way. In parallel way which is also known as *best feasible merge*, at each iteration the best feasible merge of two routes that leads to largest saving is done, while in sequential way the first saving which can be merged feasibly to the current route is considered.

Gaskell (1967) enhances the Clarke and Wright method by introducing new saving calculation procedures. As shown in Figure 4, if the end of the route is on point A, then based on Clarke and Wright (1964) method, saving value of points p_1 and A will be greater than saving value of P_2 and A. So p_1 will have priority over p_2 to be added to the route while many planners prefer to add p_2 first to the route.

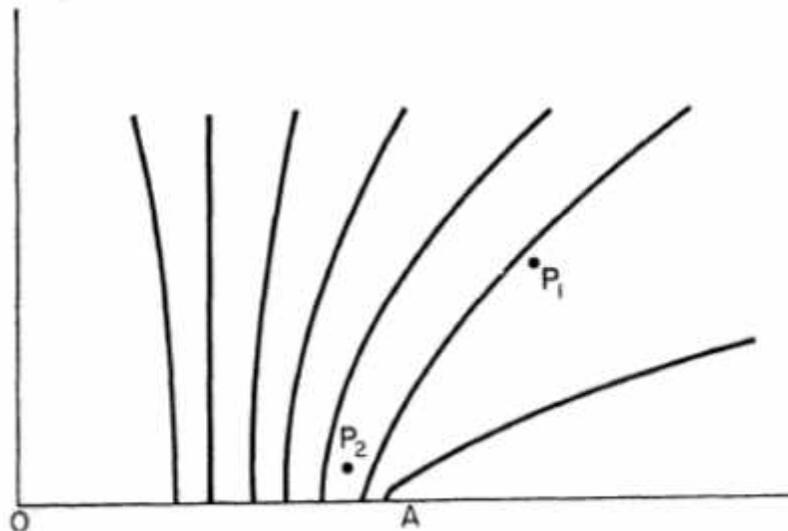


Figure 4: two potential points, p_1 and p_2 , for route extension (Gaskell, 1967)

To address this issue two new saving formulas are introduced in his work which gives the

priority to point P_2 before point p_1 .

Yellow (1970) says that classical Clarke and Wright method of saving calculation requires a large matrix or a computer file of saving values. All possible saving methods should be calculated and sorted. Hence to introduce a rapid and effective search method of the link with the highest saving, saving formula is redefined in a polar- coordinated way for selection of a point to extend the current route. Based on this new saving formula, a locus of points with equal saving values is defined. Then the search for the link with the highest saving value could be carried out on a list of polar-coordinated delivery points (customer points) that are arranged based on their angular coordinate. This procedure eliminates the need for creating a saving list.

Holmes and Parker (1976) define symmetric and nonsymmetrical algorithms based on saving method that exploits a perturbation procedure to improve the solutions. In their method, surpassing of some links and excluding them from solutions seems to generally improve the quality solutions.

Mole and Jameson (1976) propose a sequential method of route construction based on Clarke and Wright saving criterion. Their method consists of a set of applicable features: each vehicle could have its exclusive constraints such as capacity constraints and customer priority allocations, total distance traveled and delivery territories. Maintenance programs are also capable of being considered in serviceability of the vehicles. They further claim that that this method is fast in terms of sensitivity analysis with respect to fleet requirements.

By reviewing the literature on VRP heuristics, Buxey (1979) concludes that Monte Carlo simulation could be a good alternative method to schedule the vehicle routes since all previous VRP solution methods rely greatly on “trial and error” evaluation procedures and good solutions are all the time obtained by constructing many alternatives. But for large sized problems

schedule population should be constrained artificially so an appropriate sample could be taken and the selection procedure could be conducted more easily. These requirements are satisfied by selecting the saving criteria. This selection has some benefits. Saving criteria of selecting candidates to extend the route allows for tour crossing, what that always happens when there are unequal loading requirements. In fact some perturbations exist in the tour geometry because of these unequal loading requirements and there should be an allowance for a limited number of these perturbations (crossings). The author mentions that selection of a good starting solution could help the Monte Carlo simulation significantly.

Paessens (1988) conducted a literature review on the vehicle routing problem. He claims that some of the time the parametric saving formula doesn't perform well and therefore adds another parametric parameter to the saving formula:

$$S_{ij} = C_{i0} + C_{0j} - \} C_{ij} + \sim |C_{0i} - C_{j0}| \quad (3)$$

However adding more parameters to the saving formula worsen the computational times significantly. The author recommends using of Traveling Salesman Problem to improve each route after VRP solution is done.

Altinel and Oncan (2005) add another term to the equation. They mention that the vehicle routing problem is as a manner of fact a combination of multiple traveling salesman problem and bin packing problem. The new term they add to saving equation is originally a method to solve the bin packing problem. In filling of the packages with limited capacity put the larger items first.

$$S_{ij} = C_{i0} + C_{0j} - \} C_{ij} + \sim |C_{0i} - C_{j0}| + v \frac{d_i + d_j}{d} \quad (4)$$

The same logic is considered for capacitated vehicle. In allocating the vehicle capacity to the

customers' demand prioritize larger demands. The last term in Equation takes the most advantage of the vehicle capacity. As shown in Figure, there is a simple network of eight customers on the circle and at the center the depot is located. The distances of any two adjacent customers are the same as well as distance of each customer from the depot. The vehicle capacity is considered to be 100. If the saving equation proposed by Paessens (1988) is used (ignoring the customers' demand shown beside the nodes), there is no difference in between saving values of any two adjacent neighbors. But with considering the customers' demand and exploiting Formula, customers with the demand 80 and 20 will have the priority to be added to the route.

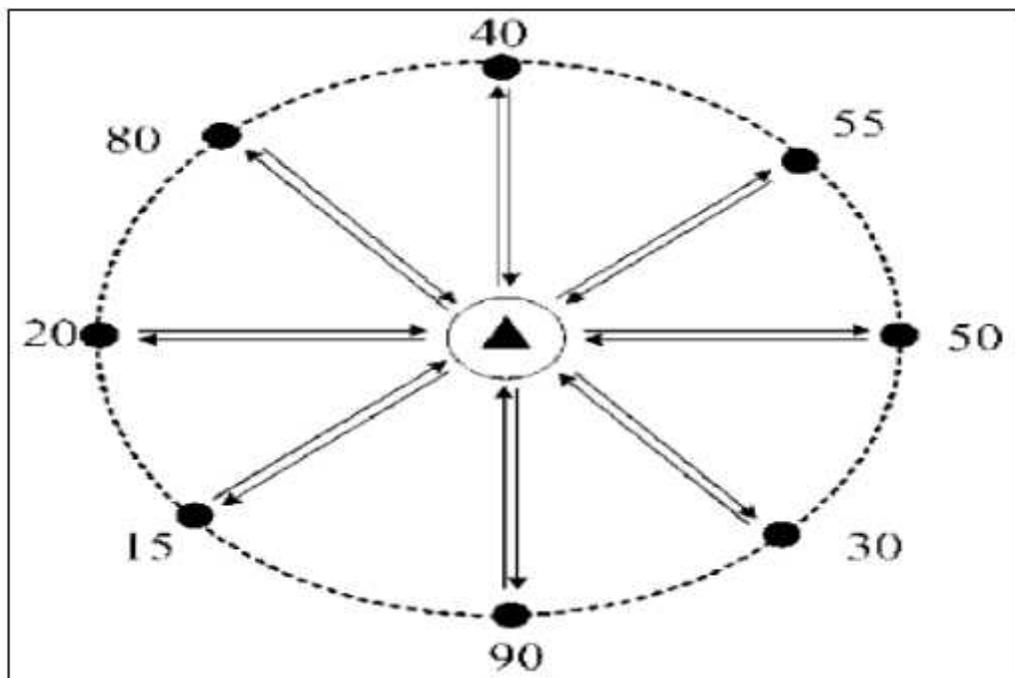


Figure 5: the effect of demand consideration in selection of customers with equal saving values (Altinel and Oncan, 2005)

2.2.2 Sweep algorithm

According to Laporte (1992a), “It seems that the origins of the sweep algorithm can be traced back to the work of Wren (1971) and Wren and Holliday (1972) for CVRPs with one or several depots and vertices in the Euclidian plane” (p. 355). Laporte (1992a) further adds that this method got its name (Sweep) from the work of Gillett and Miler (1974).

Gillett and Miler (1974) claim that Sweep algorithm outperforms Gaskell’s saving algorithm and also slightly outperforms Christofides and Eilon’s results, but it is not as computationally efficient as they are. Sweep algorithm basically works based on the polar coordination of the network nodes. It could be summarized as follows (Gillett and Miller, 1974 and Laporte, 1992a):

- (1) sort all the customers based on their polar angel
- (2) choose an unused vehicle and start assigning the unrouted customers having smallest angle (*forward-sweep algorithm*) or biggest angle (*backward-sweep algorithm*) to the vehicle as far as its capacity allows, if any customers remain unrouted, repeat step 2 for new vehicles.
- (3) perform TSP for each route to optimize it. Exchange customers between adjacent routes, i.e. start exchanging from nearest locations to each route. This could improve the results.
Re-optimize and stop.

Gillett and Miler (1974) mention that the computational time for Sweep algorithm increases linearly with growing the problem size if the average number of locations per route remains constant. But in the case of increasing average number of customers per route assuming the number of total locations remains constant, computational time grows quadratically.

2.3 Modern Heuristics

Tabu search is probably the best heuristic method which produces quality solutions for VRP. Gendreau, Hertz and Laporte (1994) proposed a tabu search method called TABUROUTE, for capacitated and length-constrained VRP. Among the adjacent solutions, they exploit a process of omitting and reinserting vertices, called GENI, to improve the solutions. They mention two important features of their method that makes it successful. First, infeasible solutions are allowed with the mechanism of penalty cost, so the risk of local optimum is minimized and second, the GENI procedure which causes perturbations in the solution periodically and this again reduces the risk of local optimum. However as claimed by the authors, “tabu search is a metaheuristic that must be tailored to the shape of the particular problem at hand” (p. 1283).

Baker and Ayechew (2003) use a hybrid genetic algorithm method to solve VRP. The initial population of solutions is generated randomly, in a structured manner or a mixture of random and structured. In structured solutions, customers are sorted and then are allocated to vehicles based on some criteria. After generating the initial population solutions and their members (parents), a two-point crossover method is exploited to produce the offspring. Parent solutions are selected from initial population solutions and then two points are selected randomly from the chromosome.

	1	2	3	4	5	6		7	8	9	10	11	12	13	14	15	16	17	18	19	20
P1:	1	1	1	1	2	2		2	3	3	2	3	3	3	3	4	4	4	4	4	1
P2:	1	1	2	2	1	2		2	2	2	3	3	3	3	3	4	4	4	4	4	4

Figure 6: GA parents for VRP (Baker and Ayechew, 2003)

The first offspring takes the gens from the left side of the first point on first parent, along with the gens between two points on second parent and the gens from the right side of the second point on first parent. Figure 6 shows the two parents and sorted customers on the first line along

with dashes indicating the two randomly selected points on the chromosome. Figure 7 shows the offspring generated based on the procedure explained above. The quality of the solutions is evaluated based on a fitness function, which is the total distance traveled. TSP could be used to explicit the route of each vehicle. Mutations and neighborhood search are also applied on this GA. Authors claim their GA method, in case of coming along with neighborhood search could compete with tabu search and simulated annealing.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
O1:	1	1	1	1	2	2	2	2	2	3	3	3	3	3	4	4	4	4	4	1
O2:	1	1	2	2	1	2	2	3	3	2	3	3	3	3	4	4	4	4	1	4

Figure 7: GA offspring for VRP (Baker and Ayechev, 2003)

Bell and McMullen (2004) propose a single and a multiple ant colony optimization for VRP. In this method, each vehicle is simulated as an ant and ants construct the vehicle routes by walking on the network. Ants select the next location to be added to the route based on maximizing a function of pheromone value of potential links and inverse distance of links. Each ant has a memory that records the locations she that she has met to prevent visit them again. Each ant gets back to the depot when the capacity is filled up. Before initialization of the algorithm there is an initial value of pheromone on all the links. There are local pheromone updates after each ant is done with its own solution. Global updates are done after a predetermined number of solutions are achieved. They further add to method to improve the quality of the solutions. First they use the 2-opt method, which all possible pairwise exchanges of customers for any individual ant route is examined to see if there is any improvement. The second neighborhood search is a candidate list which allows the ant to examine only a restricted set of customers which are close together. Authors further conduct an experimental design for some cause problems and define

single or multiple ant colony as one experimental factor and candidate list size as another one. Authors claim that solutions are within 1% of the known optimal solutions.

2.4 Energy and VRP

Vehicle energy consumption is a very critical factor both for auto manufacturers and consumers. Auto manufacturers conduct lots of research to figure out accurate relationships of energy consumption with other parameters. Delucchi, Burke, Lipman and Miller (2000) present a model for vehicle energy consumption. In this model energy consumption is a function of trip parameters such as vehicle speed, road grade, and trip duration and vehicle parameters such as vehicle weight and engine efficiency. Their model gets these inputs and calculates the actual energy consumption of the vehicle based on physics laws and empirical approximations.

Simpson (2005) proposes a parametric analytical model for vehicle energy consumption. This model as a dynamic vehicle simulator is capable of considering stochastic driving cycles and is featured to perform sensitivity analysis and model uncertainty. Vehicle energy consumption is predicted through parametric analytical expressions, based on road load equations.

Kara et al. (2007) for the first time define a new cost objective for VRP which is energy based. They not only consider the distance of the links of the network, but also consider the weight of the vehicle over the links and define a “load based” cost objective. The energy consumption over a link is defined the product of the distance of the link and the weight of the vehicle over the link. Obviously the aim is to minimize the energy consumption of the vehicle.

Gusikhin et al. (2010) argues that most of VRP commercial packages focus on total distance travelled as main objective to minimize it. It may be claimed that there is a correlation between the distance traveled and energy consumption, so minimizing the distance automatically leads to minimize the energy consumption. However, it should be noted that the correlation between the

distance and energy consumption is not a perfect correlation and energy consumption depends also on highway driving or city driving. For a mixed-fleet of vehicle with different city and highway MPGs, they propose models to design the routes and vehicle assignments to the customers in a way to minimize energy consumption.

Charoenroop, Satayopas and Eungwanichayapant (2010) propose a city bus routing model to minimize energy consumption. They claim that public transportation designs are usually based on energy minimization considerations. Their model is an extension of CVRP that aims to minimize fuel consumption of the bus. Fuel consumption is considered to be a function of velocity, weight and power per weight ratio, i.e. the travel cost of link ij is based on the velocity of the bus on link ij , gross weight of the bus (Passengers plus vehicle) over link ij and a rate called power per rate. Branch and bound is exploited to solve the model to minimize the fuel consumption.

Mirzaei and Krishnan (2011) defined a new saving calculation that is energy based. The cost of traveling over link ij is the product of distance of link ij and the vehicle load weight over link ij and the constant gravity, “ g ”, which is missing in the work of Kara et al. (2007). So instead of distance saving values, energy saving values are calculated and then routes are constructed based on energy saving list. Detailed explanation of their method is presented in Chapter 3.

CHAPTER III METHOD

3.1 Introduction

As mentioned earlier, this research focuses on generating heuristics for CVRP capable of saving on fuel consumption and consequently lesser emissions of poisonous fumes. The idea is based on the fact that the MPG of transportation vehicles is dependent on the total load weight they are carrying. This is the same logic I. Kara, B. Kara, and Yetis, 2007 consider in defining the work a vehicle does over arc ij as “weight of vehicle over link (i, j) * distance of link (i, j) ”. Recall from the classic mechanics that the work is equal to force multiplied by displacement. Therefore the work a vehicle does is equal to the forces resist against vehicle movement multiplied by the distance vehicle travels:

$$W = \vec{F} \cdot \vec{d} \tag{5}$$

The forces that act against vehicle movement are friction force and aerodynamic drag. In this research the aerodynamic drag will be ignored and the focus will be only on friction force. So by ignoring the aerodynamic drag, the force term in equation 5 will stand only for friction force:

$$W = \vec{F}_{RR} \cdot \vec{d} \tag{6}$$

And the friction force is:

$$\vec{F}_{RR} = C^{RR} m \vec{g}^G \tag{7}$$

Where, C^{RR} is the rolling resistance and $m \vec{g}^G$ is the vehicle weight. C^{RR} depends on road surface and tire type and could be considered constant along the whole tour of the vehicle and therefore will be eliminated from calculations (Mirzaei & Krishnan, 2011). Therefore the “load based” cost objective for VRP can be updated as:

$$\text{Cost of (i, j)} = [\text{Load of the vehicle over link (i, j)} + \text{tare}] * g * \text{distance of (i, j)} \quad (8)$$

Equation 8 is equal to the work the vehicle does over link (i, j) and can be said is equal to the energy vehicle consumes since work is energy. Equation 8 is the baseline concept in the modification of nearest neighbor algorithm and saving algorithm to make them “energy efficient”. The energy efficient heuristics are first described followed by the original heuristics.

3.2 Nearest neighbor algorithm

The notations which will be used in this algorithm are presented below. It is tried to include all the notations that Mirzaei and Krishnan (2011) used in their work since their work will be later compared with the methods being presented here.

V	Set of nodes (customers)
A	Set of arcs
m	Number of available vehicles
N	Fleet size of the VRP solution ($1 \leq N \leq m$)
R_v	Set of customers assigned to the vehicle's route v ($1 \leq v \leq N$)
w_0	Vehicle initial weight
w_i	Weight of customer i 's demand
w_t	Vehicle tare weight
L_0	Vehicle initial load weight
C_{ij}	Distance between node i and j
D_{ip}	Demand of node i for product p
u_p	Unit product weight for product p
P	Set of products
VC	Vehicle capacity in terms of number of the products

VC_w Vehicle capacity in terms of total load weight

Nearest neighbor method is originally a traveling salesman problem (TSP) heuristic. TSP could be defined as the following:

Let $G = (V, A)$ be a graph which V is a set of nodes and A is a set of arcs. A matrix $C = (c_{ij})$ represents the distances (costs) of going from node i to node j . The problem is to determine the shortest path which goes through all the nodes only once and once.

TSP like VRP is a NP-hard combinatorial problem and there is a rich literature on its heuristics. TSP heuristics could be divided into two main categories (Laporte, 1992b): (i) tour construction procedures and (ii) tour improvement procedures. Nearest neighbor belongs to the tour construction heuristics and tries to get the maximum benefit from going one step to the next one. Hence these kind of heuristics are sometimes called “greedy heuristics”. Nearest neighbor algorithm steps according to the Laporte (1992b) are: (Algorithm 1)

- (i) Select an arbitrary point as starting point
- (ii) Determine the closest node to the last one already considered and add it to the tour.
Repeat step (ii) if any nodes are not included in the tour.
- (iii) Link the last node of the tour to the start point

These steps are designed for TSP. However Laporte (1992a) argues that TSP algorithms can often be used for solving VRPs. He adds that nearest neighbor method can be used to solve CVRP almost without modification. Hence, steps needed to solve a CVRP with the nearest neighbor algorithm could be defined as follows: (Algorithm 2)

- (i) Dispatch a vehicle from the depot to the closest node to the depot.
- (ii) Determine the closest node to the last one already considered and add it to the vehicle tour. Repeat step (ii) as long as vehicle capacity allows.

(iii) Repeat steps (i) and (ii) for new vehicles if any nodes are not visited yet.

In the next part the nearest neighbor algorithm (for TSP) will be modified to solve CVRPs in an “energy efficient” manner.

3.3 Energy efficient nearest neighbor (EENN) algorithm for CVRP

As mentioned before, nearest neighbor algorithm is originally a TSP heuristic. However after solving a VRP, any vehicle route can be considered as a TSP problem. Hence, nearest neighbor algorithm will be modified in an energy efficient manner to improve the energy consumption of each vehicle assigned to a route. Figure 8 shows that if the nearest neighbor algorithm for CVRP is being used and the vehicle is already at the node q and both nodes p and s are the closest nodes to the q, it doesn't make any difference to select either of them.

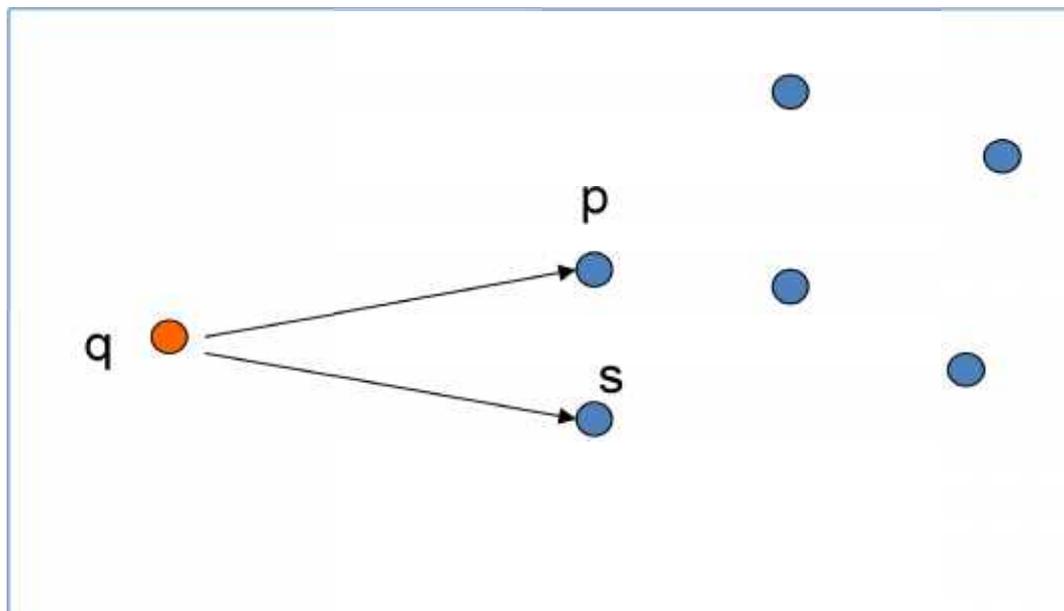


Figure 8: nearest neighbor algorithm node selection criteria

However, as illustrated in Figure 8, if customer demands be considered (assuming all the

demands are identical) p and q will not have the same priority to be selected as the next node and it is more energy efficient to serve the node with greater demand (node s) first . The logic comes from the idea that vehicle should get rid out of the heavier demands as soon as possible to carry them less in the whole vehicle tour. This will cause the vehicle to have an average lighter load weight over the vehicle tour and consequently the MPG of the vehicle will be improved. This also can be explained by using equation 5, an average smaller value of “weight of vehicle over link (i, j)” will cause to do less work and subsequently less energy consumption. The EENN algorithm could be defined in different ways, one is: (Algorithm 3)

- (i) Solve the problem with nearest neighbor algorithm for CVRP (Algorithm 2)
- (ii) Start with an arbitrary vehicle route and perform:

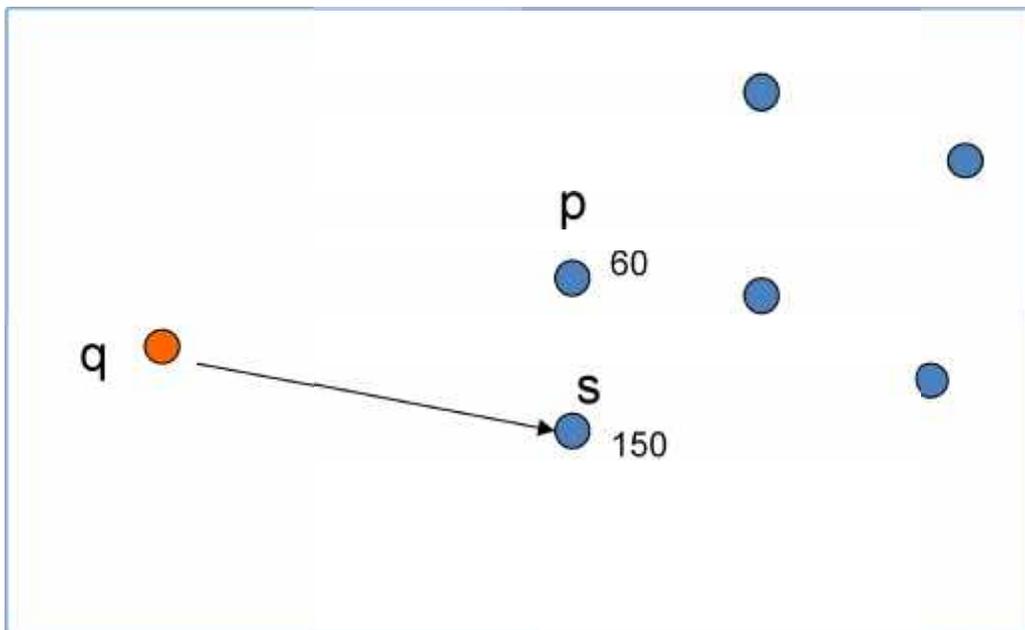


Figure 9: EENN node selection criteria

1. Start from any arbitrary node in the vehicle tour
2. Being currently on node i (i=1 for depot), go to the next node j having the least

$$\text{value of: } (1 - \tau)c_{ij} + \tau \frac{1}{w_j} \quad (8)$$

3. Repeat step 2 if any nodes are not visited in the vehicle tour

(iii) Repeat step (ii) for all other vehicle routes

Formula 8 tries to create a trade-off between two criteria for selecting the next node to be added to the current route. These two criteria are the nearest neighbor and the heaviest neighbor to the current route. Appropriate alpha values that minimize the objective function (energy consumption) of the algorithm should be investigated for any vehicle tour. However, in most of the parametric problems it is important to investigate the appropriate range of the parameter to vary and in this case since the ratio of c_{ij} and w_j varies from case to case, c_{ij} and $1/w_j$ will not have the appropriate sensitivity to $(1 - \tau)$ and τ respectively all the times. Hence c_{ij} and w_j could be modified in the following way:

$$\left\{ \begin{array}{l} c'_{ij} = \frac{c_{ij}}{\text{Min}_{i,j \in V} c_{ij}} \quad \text{For } \forall i, j \in V \\ w'_j = \frac{\text{Max}_{j \in V} w_j}{w_j} \quad \text{For } \forall j \in V \end{array} \right. \quad (9)$$

The following example helps to clarify the problem:

Suppose there is a vehicle route consisting of five nodes. The vehicle is currently at node 1 ($i = 1$) and the other 4 nodes are not yet included in the route. Table 1 shows the distances of these 4 nodes from node 1 and the demands weight of each. If the closest node to the node 1 is supposed to be selected, node 5 will be selected. However if the node with heaviest demand is supposed to be selected, node 4 will be selected.

Table 1: An Example of Input Data for Algorithm 3

Nodes	2	3	4	5
Distances from node 1	23	25	21	19
Demands weight	50	75	90	60

If formula 9 is calculated for Table 1, i.e. divide all the nodes distances by minimum distance (node 5) and divide the heaviest demand (node 4) by all nodes weights for each node, the same decision would be taken regarded to closest node criteria and heaviest demand criteria (Table 2).

Table 2: An Example of Input data for Algorithm 4

Nodes	2	3	4	5
Distances from node 1	23/19=1.21	25/19=1.31	21/19=1.1	19/19=1
Demands weight	90/50=1.8	90/75=1.2	90/90=1	90/60=1.5

Using Formula 6 to calculate new values of c_{ij} and, Formula 6 can be updated as:

$$(1-r)c'_{ij} + rw'_j \tag{10}$$

As mentioned before, using Formula 9 and getting the formula 10 consequently, will relax the necessity of determining appropriate ranges of alpha for different problems. By using Formula 10 in the EENN for CVRP, Algorithm 3 will be updated as following: (Algorithm 4)

- (i) Solve the CVRP with nearest neighbor algorithm for CVRP (Algorithm 2)
- (ii) Start with an arbitrary vehicle route and perform:
 1. Start from any arbitrary node in the vehicle tour
 2. Being currently on node i ($i=1$ for depot), go to the next node j having the least

value of $(1-\Gamma)c'_{ij} + \Gamma w'_j$ (Formula 10)

3. Repeat step 2 if any nodes are not visited in the vehicle tour

(iii) Repeat step (ii) for all other vehicle routes

3.4 Saving Algorithm

Saving algorithm is one of the most popular heuristic methods both in literature research and commercial VRP packages. It was originally developed by Clarke and Wright, (1986). According to Christofides and Eilon (1969) and Laporte (1992a), Clarke and Wright saving method can be described as following:

- (i) Assume that each customer has a unique vehicle that serves it and backs to depot afterwards. Hence $n-1$ vehicle routs $(i, 1, i)$ are created ($i = 2, \dots, n$)
- (ii) By merging customers i and j , one vehicle is eliminated and the saving in mileage achieved by this merge is given by:

$$S_{ij} = c_{i1} + c_{1j} - c_{ij}, \quad (11)$$

Compute Formula 11 for $i, j = 2, \dots, n$ and $i \neq j$.

- (iii) Order the savings list in a non-increasing fashion.
- (iv) Consider two vehicle routs containing arcs $(i, 1)$ and $(1, i)$. If $s_{ij} > 0$, introduce arc (i, j) by tentatively merging routs $(i, 1)$ and $(1, i)$ and implement the merge if it is feasible. Keep performing this step until no further improvement is possible, stop.

Mirzaei and Krishnan (2011) have modified this method in an “energy efficient” manner which will be explained in the next part.

3.5 Energy Efficient Clarke Wright Method for VRP (ECW) (Mirzaei and Krishnan, (2011))

This method is similar to the original Clarke and Wright method with this difference that

instead of calculating saving values of distances, saving values of energy consumption are calculated. To calculate the energy consumption, they assume that the vehicle will be always dispatched from the depot with full capacity. The notations that are used here are the same notations that are used in Nearest Neighbor Algorithm. According to Mirzaei and Krishnan (2011), the new saving value is calculated as:

$$\begin{aligned}
S_{ij}^E &= (w_0 c_{0i} + (w_0 - w_i) c_{i0} + w_0 c_{0i} + (w_0 - w_j) c_{j0}) - (w_0 c_{0i} + (w_0 - w_i) c_{ij} + (w_0 - w_i - w_j) c_{j0}) \\
S_{ij}^E &= (w_0 - w_i) c_{i0} + w_0 c_{0j} - (w_0 - w_i) c_{ij} + w_i c_{j0} \\
\forall i, j \in V \setminus \{1\}, i \neq j
\end{aligned} \tag{12}$$

Vehicle initial load weight is assumed to be equal to vehicle capacity:

$$L_0 = VC_w \tag{13}$$

They argue that the unlike Clarke Wright method, Energy Efficient Clarke Wright (ECW) is sensitive to the direction of the routes, i.e. saving value of the route 1-i-j-1 could be different of route 1-j-i-1. The ECW algorithm is described in Figure 10.

- 0- Start
- 1- Calculate Equation 12 for each pair of nodes.
- 2- Sort the obtained value in descending order and create a saving list
- 3- Select the highest saving in the list; if no constraint will be violated, for any S_{ij}
- 4- If neither i nor j is already assigned to a route then start a new route with these nodes; otherwise go to 5,
- 5- If i is already assigned and is the last node in its route, then insert j after i in the same route; otherwise go to 6
- 6- If j is already assigned and is the first node on its route, then insert i before j in the same route; otherwise go to 7
- 7- If both are already assigned to different routes and i is the last node in one route and j is the first node in the other route, then the two routes will be merged; otherwise go to 8
- 8- Eliminate S_{ij} form the saving list and if the list is not exhausted yet go to 3; otherwise stop. The set of routes formed during step 4-7 are the answer. Make a list of customers which are not assigned and put them in a new route and add the route to the set of previously found routes.

Figure 10: ECW algorithm (Mirzaei and Krishnan (2011))

3.6 Improved Energy Efficient Clarke Wright Method for VRP (Improved ECW)

To improve ECW, we take the inspiration from work of Altinel and Oncan (2005). The main concept in modifying the nearest neighbor algorithm for CVRP to be “Energy Efficient” was to get rid out of heavier demands as soon as possible. This is the same idea Martello and Toth (1990) use (as cited in Altinel and Oncan, 2005, p.955) to solve the bin packing problem: in filling the bin with limited capacity, put the larger items first. It is believed that if the *first fit decrease* idea be put together with saving method which is proposed by Mirzaei and Krishnan (2011), more energy consumption savings could be achieved since not only the routes are

constructed in a way to minimize energy consumption but also the vehicle capacity will be used more efficiently. On the other hand, by taking the first fit decrease idea into consideration while constructing the routes, an order of meeting the customers could be generated in a way that demands' weight are decreasing and this will also contribute to energy consumption. The saving method proposed by Mirzaei and Krishnan (2011) is modified into the following saving formula:

$$S_{ij}^E = (w_0 - w_i)c_{i0} + w_0c_{0j} - (w_0 - w_i)c_{ij} + w_ic_{j0} + \Gamma \frac{w_i + w_j}{L_0} \quad (14)$$

The last term in Formula 14 takes into the account the importance of heavier demands while constructing the routes. Although in calculation of energy consumption in Mirzaei and Krishnan (2011) work L_0 is equal to vehicle capacity (Formula 13), but in Formula 14 it is defined as total customers' weight:

$$L_0 = \sum w_i \quad \text{For } \forall i \in V / \{1\} \quad (15)$$

CHAPTER 4

Computational Results

4.1 Computational results of EENN for CVRP

A set of 27 VRP benchmark problems from Augerat et al. (1995) will be solved with nearest neighbor algorithm for CVRP (Algorithm 2) and EENN for CVRP (Algorithm 4) and then the results will be compared together. To calculate the total energy consumption for each CVRP problem a number of issues and assumptions should be clarified first:

- All demands are assumed to be identical ($u_p = u$ and $D_{ip} = D_i$).
- Weight of customer i 's demand can be calculated as:

$$W_i = D_i * u \quad \text{For } \forall i \in V \quad (16)$$

- Vehicle initial load weight can be calculated as:

$$L_0 = \sum w_i \quad \text{For } \forall i \in R_v \quad (17)$$

- Vehicle initial weight is constituted from vehicle tare weight and vehicle initial weight:

$$w_0 = w_t + L_0 \quad (18)$$

The first problem, A-n32-k5, is a set of 32 customers with their given demands in terms of number of products. The unit product weight of $u = 50$ kg, vehicle capacity of $vc = 10000$ kg, and vehicle tare weight of $w_t = 10000$ kg are assumed for all the 27 problems. The solution of first problem which is generated by nearest neighbor algorithm for CVRP (same as section (i) of Algorithm 4) contains 3 vehicles. Each vehicle tour starts from depot (1), serves a number of customers as its capacity allows and backs again to the depot:

- Vehicle 1: 1-31-27-17-13-2-8-14-22-32-20-18-3-4-1
- Vehicle 2: 25-15-28-21-6-30-16-11-26-23-10-19-9-29-5-12-1

- Vehicle 3: 1-7-24-1

The total distance traveled (sum of three vehicles) is 770.22 m and based on the Equation 5 the total energy consumption could be calculated as following:

- Energy consumption for vehicle 1 = $[(w_0) c_{1,31} + (w_0 - w_{31}) c_{31,27} + (w_0 - w_{31} - w_{27}) c_{27,17} + \dots + (w_0 - w_{31} - \dots - w_4) c_{4,1}] * g$

The last term in the above equation is equal to $w_t * c_{4,1}$ since the vehicle has dropped all the demands when head backs to the depot. The energy could be calculated for other 2 vehicles in the same way. The total energy consumption (sum of three vehicles) is 100427583 N.m. The EENN for CVRP (Algorithm 4) will try to improve each of the three vehicle routs generated by nearest neighbor algorithm (Starting point is depot). New routes for vehicles are:

- Vehicle 1: 1-20-13-3-2-18-17-14-8-31-22-32-4-27-1
- Vehicle 2: 26-25-16-28-5-10-29-12-21-11-6-9-23-15-30-19
- Vehicle 3: 1-7-24-1

The first vehicle rout is generated by an $r = 4\%$, the second one has also an $r = 4\%$ and the third vehicle route is not changed ($r = 0$) compared with nearest neighbor algorithm results. The total distance traveled is reduce to 734.52 m, showing a 4.6% improvement and the total energy consumption is also reduce to 94066256 N.m., showing a 6.3% improvement. Table 3 shows the 27 problem solutions. The case studies were run on an Intel Core Quad CPU, Q8400 @2.6GHz, 2.66 GHz, 4GB RAM, and 32-bit Operating System. The results show that EENN outperforms the nearest neighbor in 25 cases out of 27 in terms of energy consumption. The performances of both algorithms are evaluated based on the total distance traveled which is the traditional objective function of VRP and total energy consumption. The energy columns in table 1 are calculated based on total energy consumption of all vehicles in each problem solution. The

improvements achieved by applying the EENN algorithm in terms of percentages of improvement of energy consumption and total distance traveled are also presented in separate columns.

Table 3: Comparison of NN and EENN Algorithms (problems from Augerat et al., 1995)

Problem	NN		EENN		Saved Energy by applying EENN	Distance reduction by applying EENN (%)	NN computational time (s)	EENN computational time (s)	
	Distance	Energy	Distance	Energy					
1	A-n32-k5	770.22	100427583	734.52	94066256	6.3%	4.6%	0.0005	0.020
2	A-n33-k5	599.36	78877737	591.44	78084188	1.0%	1.3%	0.0005	0.018
3	A-n33-k6	669.16	92664115	669.46	91375864	1.4%	0.0%	0.0005	0.019
4	A-n34-k5	680.62	92404218	675.91	91854779	0.6%	0.7%	0.0006	0.020
5	A-n36-k5	754.07	102852105	765.07	99532332	3.2%	-1.5%	0.0006	0.022
6	A-n37-k5	802.36	102457077	790.71	99936402	2.5%	1.5%	0.0006	0.025
7	A-n37-k6	841.44	111300901	831.63	109712699	1.4%	1.2%	0.0007	0.021
8	A-n38-k5	705.11	91417310	705.11	91417310	0.0%	0.0%	0.0006	0.022
9	A-n39-k5	841.36	114082883	841.36	114082883	0.0%	0.0%	0.0006	0.023
10	A-n39-k6	751.28	97005046	762.75	96822034	0.2%	-1.5%	0.0006	0.023
11	A-n44-k7	927.12	121759520	901.00	118148798	3.0%	2.8%	0.0008	0.027
12	A-n45-k6	813.00	111011673	803.33	109226791	1.6%	1.2%	0.0008	0.028
13	A-n45-k7	1078.92	145099394	1023.56	136010323	6.3%	5.1%	0.0008	0.026
14	A-n46-k7	890.00	124976923	890.64	119771849	4.2%	-0.1%	0.0008	0.030
15	A-n48-k7	950.33	129993352	947.97	129023737	0.7%	0.2%	0.0009	0.029
16	A-n53-k7	1004.55	137306715	1009.71	135324944	1.4%	-0.5%	0.0010	0.031
17	A-n54-k7	1105.15	150867423	1076.23	140685282	6.7%	2.6%	0.0010	0.033
18	A-n55-k9	964.36	132478597	948.79	129332060	2.4%	1.6%	0.0011	0.032
19	A-n60-k9	1156.87	157598964	1140.77	154573294	1.9%	1.4%	0.0013	0.036
20	A-n61-k9	971.85	133882760	951.98	126263159	5.7%	2.0%	0.0012	0.036
21	A-n62-k8	1101.80	153469661	1056.93	148041704	3.5%	4.1%	0.0012	0.041
22	A-n63-k9	1299.17	178190095	1268.23	174127352	2.3%	2.4%	0.0013	0.037
23	A-n63-k10	1216.65	160970367	1215.55	160779073	0.1%	0.1%	0.0013	0.036
24	A-n64-k9	1216.15	158831180	1193.07	154905121	2.5%	1.9%	0.0014	0.036
25	A-n65-k9	1006.63	137084146	992.43	135113068	1.4%	1.4%	0.0015	0.037
26	A-n69-k9	1082.08	144779848	1077.46	143973329	0.6%	0.4%	0.0017	0.043
27	A-n80-k10	1343.76	183278631	1339.76	182709575	0.3%	0.3%	0.0021	0.052

The greatest energy improvement is 6.7 % and the greatest distance reduction is 5.1 %. In four of the cases the distances of EENN are outperforming negatively, but in all of them energy is still doing better than nearest neighbor.

Figure 3 shows improvement percentages in terms of energy and distance reduction achieved by applying the EENN algorithm in comparison to NN algorithm for all 27 benchmark problems.

The similar trend behavior of energy and distance graphs reveals this fact that there is a positive correlation between energy consumption and total distance travelled. However this correlation is not perfect. As illustrated in this Figure, there are some cases like problem 5 which energy consumption is improved by the given solution of EENN while distance improvement is negative. There are also some cases like problems 8 and 9 which there is neither energy improvement nor distance reduction. It indicates that none of the vehicle routes have been improved by the second step of EENN algorithm and hence NN for CVRP is performing similarly to EENN. The positive, imperfect correlation between distance and energy indicates that usually by reducing the total distance traveled energy consumption reduces consequently, but it is not necessarily the case all the time. In the other words, the shortest path is not necessarily the most energy efficient path all the time. The computational results of EENN algorithm proves that the idea of “get rid out of heavier customers’ demands as soon as possible” works in terms of energy saving.

Different alpha values do effect the energy consumption and total distance traveled of the vehicles. However, by investigating through graphs having Alpha as their x-axles and

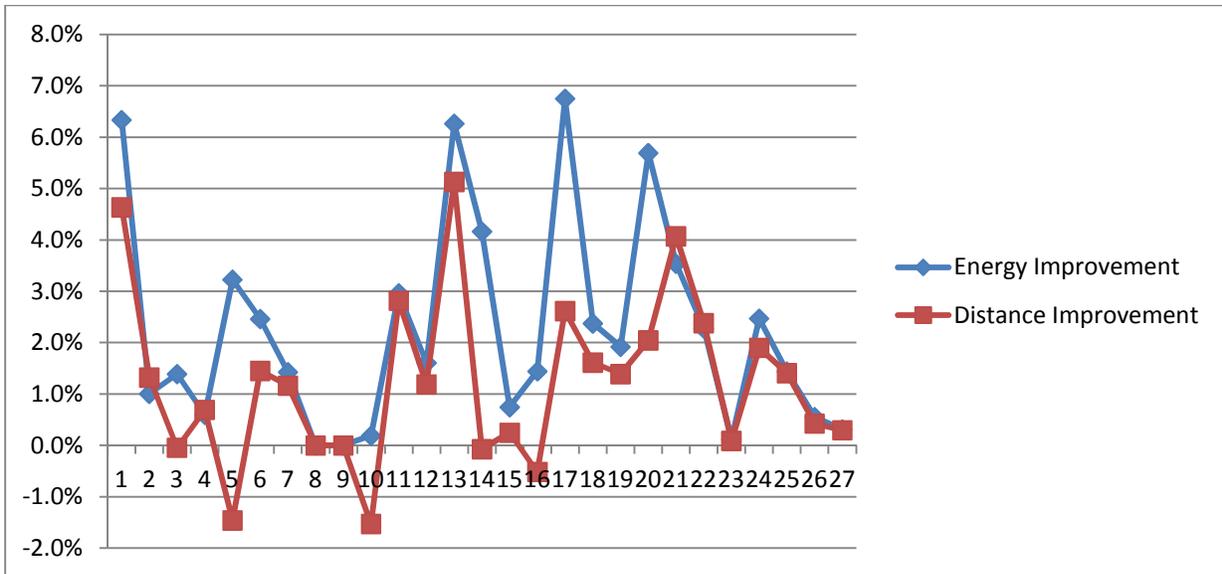


Figure 11: EENN algorithm improvement over NN algorithm

Energy/Distance Traveled as their y-axes (Figures 12 and 13) four general statements could be claimed:

- (i) Most of the times there is fluctuation in response within first 5 alpha values.
- (ii) There is a rising trend while alpha increases.
- (iii) Most of the times the response is constant within long intervals of alpha (5 alphas)
- (iv) There is a strong positive correlation between distance and energy consumption, i.e. the trends of distance and energy graphs are very similar.

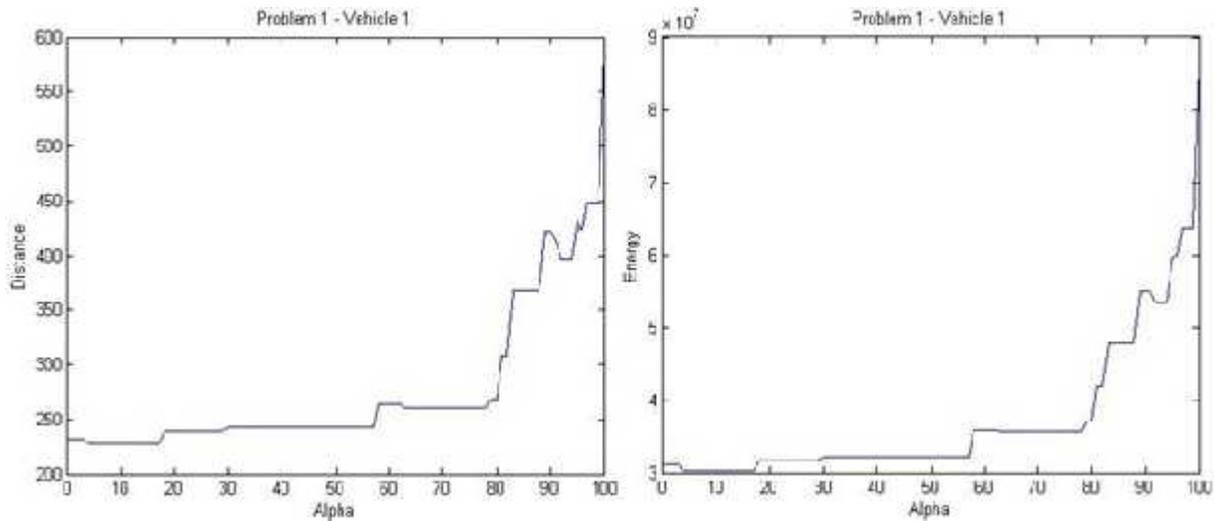


Figure 12: Distance and energy trends of Problem 1, Vehicle 1

Figure 12 shows the Distance and Energy trends of the first vehicle in first benchmark problem. As illustrated by the Figure 12, all four general statements apply here. There is fluctuation in first 5 alphas, a rising trend is obvious, responses are constant over long intervals of alphas, and there is an almost perfect correlation between the distance and energy. However this perfect correlation is not the case all the time. Notice in Figure 13, showing the behavior of second vehicle in Problem 5, the trend in the first tick of the graph is quite different for energy and distance, besides the minimum distance is located at $\alpha = 0\%$, while the minimum energy is located at $\alpha = 11\%$.

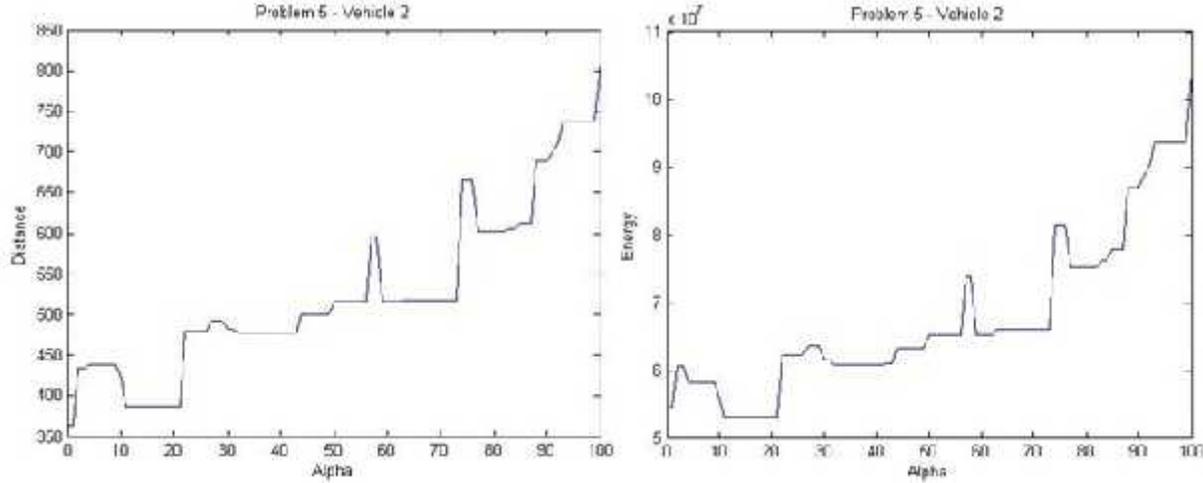


Figure 13: Distance and energy trends of Problem 5, Vehicle 2

Based on the four general statements and Figures 12 and 13, it is decided to vary the alpha from zero to five by one unit and then from five to hundred by five units. This will save on computational times.

4.2 Computational Results for Improved ECW

Since the new proposed method will be compared with the method proposed by Mirzaei and Krishnan (2011), the same benchmark problems from Augerat et al. (1995) are exploited here. The vehicle capacity is considered to be 5000 kg and a tare weight of 5000 kg is assumed for the vehicle. The product unit weight is calculated with following formula:

$$u = \frac{VC_w}{VC} = \frac{5000}{VC} \quad (19)$$

where VC is the capacity of the vehicle in terms of number of products. All assumption for calculating energy consumption are identical to EENN algorithm presented at the beginning of section 4.1 unless the vehicle initial load which is considered to be equal to vehicle capacity:

$$L_0 = VC_w = 5000 \quad (20)$$

This assumption is considered by Mirzaei and Krishnan (2011) and we stick to it in this section in regarding to compare the results of ECW and Improved ECW. Unlike original Clarke and Wright, both ECW and the Improved ECW are sensitive to the direction of the routes. Benchmark problems are solved with Improved ECW algorithm for different values of alpha and for the best solutions total distance traveled and total energy consumption is reported. Alpha values vary from 0 to 1000 for increments of 50. For benchmark number 25 as an instance, total energy consumption and total distance travelled for different values of alpha are shown in Figure 14. Alpha value of 700 which produces the least energy consumption is then selected as the best solution. This best solution of Improved ECW algorithm for each benchmark problem is compared to the ECW and Clarke and Wright algorithms in Table 4.



Figure 14: Problem 25 solved with Improved ECW algorithm for different alpha values

The results are reported in terms of energy consumption and total distance traveled. Energy and distance improvement percentages achieved by exploiting Improved ECW in comparison to

Table 4: Comparison of ECW and Improved ECW

Tare weight=5000 kg. VC=5000 kg		CW		ECW		Improved ECW			ECW Comp. Time (s)	Improved ECW Comp. Time (s)	Dist. Improvement over ECW (%)	Energy Improvement over ECW (%)
Problem	Dist.	Energy	Dist.	Energy	Alfa	Dist.	Energy					
1	A-n32-k5	844	64734449	801	62923185	0	801	62923185	0.08	1.81	0.000	0.000
2	A-n33-k5	694	52417623	697	53583459	350	714	53159962	0.087	2.03	-2.428	0.790
3	A-n33-k6	776	62439637	775	57261209	250	751	54978467	0.087	2.00	3.061	3.987
4	A-n34-k5	807	61937356	799	60279083	0	799	60279083	0.112	2.18	0.000	0.000
5	A-n36-k5	831	64603932	838	62296017	100	839	61442311	0.115	2.60	-0.123	1.370
6	A-n37-k5	705	56094645	720	53735460	600	705	52283525	0.127	2.81	2.104	2.702
7	A-n37-k6	980	74632359	979	74617043	1000	1027	73479476	0.125	2.81	-4.879	1.525
8	A-n38-k5	795	62960861	770	57781241	100	774	56971841	0.136	3.06	-0.470	1.401
9	A-n39-k5	902	69506721	880	65845261	50	895	65269192	0.149	3.32	-1.702	0.875
10	A-n39-k6	883	66945672	873	65019035	0	873	65019035	0.147	3.32	0.000	0.000
11	A-n44-k7	1021	77698933	986	72141634	0	986	72141634	0.22	4.89	0.000	0.000
12	A-n45-k6	1013	79621012	1015	73358168	450	986	73298720	0.232	5.24	2.878	0.081
13	A-n45-k7	1200	90187043	1184	90200539	600	1171	88019767	0.231	5.23	1.090	2.418
14	A-n46-k7	940	71719200	941	68316239	0	941	68316239	0.251	5.60	0.000	0.000
15	A-n48-k7	1115	87453575	1114	83992772	600	1111	82893670	0.291	6.54	0.279	1.309
16	A-n53-k7	1093	83912212	1098	83050640	250	1043	79939652	0.399	8.84	4.989	3.746
17	A-n54-k7	1202	91939271	1215	91142860	500	1212	90956508	0.423	9.44	0.206	0.204
18	A-n55-k9	1111	85977468	1117	84382393	250	1107	83276418	0.453	9.99	0.917	1.311
19	A-n60-k9	1377	106551832	1414	106064857	50	1377	101988709	0.607	13.38	2.646	3.843
20	A-n61-k9	1103	86160883	1065	80980891	150	1070	80031120	0.639	14.14	-0.512	1.173
21	A-n62-k8	1353	102138456	1377	102736651	300	1348	100560979	0.676	14.80	2.102	2.118
22	A-n63-k9	1691	127644613	1691	126143993	500	1664	123650677	0.718	15.65	1.574	1.977
23	A-n63-k10	1359	103647386	1371	102957307	350	1354	102626652	0.714	15.60	1.213	0.321
24	A-n64-k9	1482	112580339	1472	108916746	300	1445	107185825	0.743	16.37	1.815	1.589
25	A-n65-k9	1243	93806636	1281	96803299	700	1238	91019319	0.817	17.92	3.395	5.975
26	A-n69-k9	1209	91379350	1212	92060381	800	1243	91941387	0.97	21.28	-2.572	0.129
27	A-n80-k10	1837	138307147	1822	136256431	0	1822	136256431	1.63	35.33	0.000	0.000

ECW are also presented in Table 4. The results show that in 21 out of 27 cases Improved ECW outperforms the ECW in terms of energy consumption. For the rest of 6 unimproved cases, Improved ECW and ECW are performing similarly ($\alpha = 0$ when Improved ECW is identical to

ECW). Regarded to the total distance traveled, in 15 cases Improved ECW is outperforming ECW, 6 cases ECW outperforms Improved ECW and in the rest 6 cases performance is the same. 5.975 % is the biggest energy consumption improvement which belongs to problem 25 and 4.989 % is the biggest distance traveled improvement which belongs to problem 16. Figure 15 shows improvement percentages in terms of energy and distance reduction achieved by applying Improved ECW algorithm in comparison to ECW algorithm for all 27 benchmark problems. As illustrated in this figure, there is an imperfect, positive correlation between energy consumption and distance traveled, exactly similar to the results which are achieved by comparison of nearest neighbor and EENN algorithms.

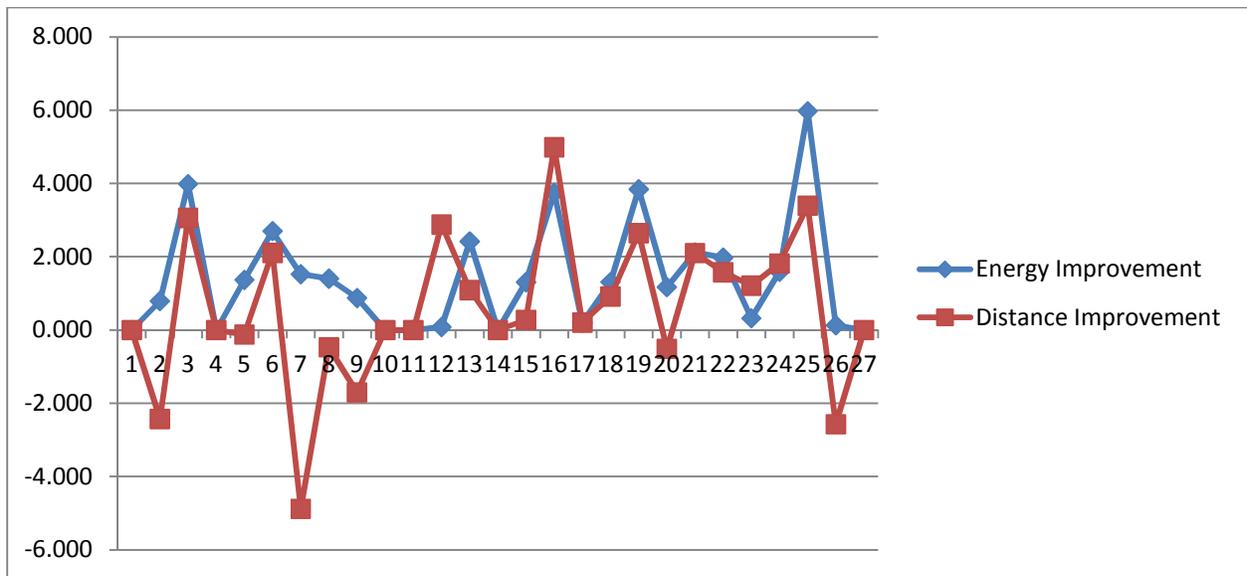


Figure 15: Improvement percentages of Improved ECW over ECW

CHAPTER 5

Conclusion and Future Work

5.1 Comparison of 5 Algorithms

In this research 5 algorithms are discussed: Nearest Neighbor algorithm (NN), Energy Efficient Nearest Neighbor algorithm (EENN), Saving Clarke and Wright (CW), Energy Efficient Clarke and Wright (ECW) and Improved Energy Efficient Clarke and Wright (Improved ECW). In presenting the family of NN algorithm and the way energy is calculated for their performance, it is assumed that the initial load of the vehicle is equal to the total customers' demands (Formula 8). But energy consumption calculation for CW families is based on the assumption that vehicle departs from the depot with full capacity (Formula 13). Adaption of fully loaded assumption allowed the comparisons between CW families since Mirzaei and Krishnan adapted this assumption. In this section all these five algorithms will be compared together based on fully loaded assumption. Formulas 16, 18, 19 and 20 along with a tare weight of 5000 KG are used for calculating energy consumptions. Table 5 shows the computational results.

Table 5: Comparison of five algorithms

NN		EENN		CW		EECW		Improved ECW		
Distance	Energy	Distance	Energy	Distance	Energy	Distance	Energy	Alfa	Distance	Energy
991.67	75796658	969.53	74337987	844	64734449	801	62923185	0	801	62923185
822.29	59368424	814.37	58851965	694	52417623	697	53583459	350	714	53159962
970.39	74818020	968.06	74597733	776	62439637	775	57261209	250	751	54978467
859.49	63668591	852.72	62859152	807	61937356	799	60279083	0	799	60279083
929.23	68498402	920.45	67467631	831	64603932	838	62296017	100	839	61442311
869.47	64823150	838.59	62118206	705	56094645	720	53735460	600	705	52283525
1236.94	94427050	1236.94	94344110	980	74632359	979	74617043	1000	1027	73479476
1013.51	78079472	1013.51	78079472	795	62960861	770	57781241	100	774	56971841
1122.43	88529133	1118.08	87277524	902	69506721	880	65845261	50	895	65269192
1001.37	73861594	1001.72	73373846	883	66945672	873	65019035	0	873	65019035
1186.03	92272914	1191.08	91478467	1021	77698933	986	72141634	0	986	72141634
1182.10	90850330	1186.25	89561214	1013	79621012	1015	73358168	450	986	73298720
1505.36	118162377	1461.16	114518771	1200	90187043	1184	90200539	600	1171	88019767
1202.21	89045819	1159.76	84579108	940	71719200	941	68316239	0	941	68316239
1252.80	93988106	1235.54	92524221	1115	87453575	1114	83992772	600	1111	82893670
1319	100920314	1286.54	98288262	1093	83912212	1098	83050640	250	1043	79939652
1547.84	118757918	1542.52	117965907	1202	91939271	1215	91142860	500	1212	90956508
1311.06	101512843	1307.80	100429609	1111	85977468	1117	84382393	250	1107	83276418
1680.03	127736777	1669.09	126876299	1377	106551832	1414	106064857	50	1377	101988709
1284.03	99544633	1246.43	96516837	1103	86160883	1065	80980891	150	1070	80031120
1544.90	115055517	1530.10	114128086	1353	102138456	1377	102736651	300	1348	100560979
1982.03	148519090	1982.03	148519090	1691	127644613	1691	126143993	500	1664	123650677
1698.42	127996935	1694.01	127645180	1359	103647386	1371	102957307	350	1354	102626652
1706.61	130314257	1692.24	128279561	1482	112580339	1472	108916746	300	1445	107185825
1437.38	108629134	1422.76	105836910	1243	93806636	1281	96803299	700	1238	91019319
1422.31	106890879	1424.13	106603556	1209	91379350	1212	92060381	800	1243	91941387
2179.08	162694009	2179.08	162694009	1837	138307147	1822	136256431	0	1822	136256431

To make the comparison easier between algorithms' performances, distance and energy columns of table 5 are presented in Figures 16 and 17 as total energy consumption and total distance traveled respectively. Energy consumption patterns in Figure 16 are quite similar together for all algorithms and only their level of performance is different. The similar pattern could be used to

draw the conclusion that all of the 5 algorithms are coded properly and coding is validated. As shown in the picture, Clarke and Wright family are generally outperforming Nearest Neighbor family and specifically Improved ECW is presenting the least energy consumption.

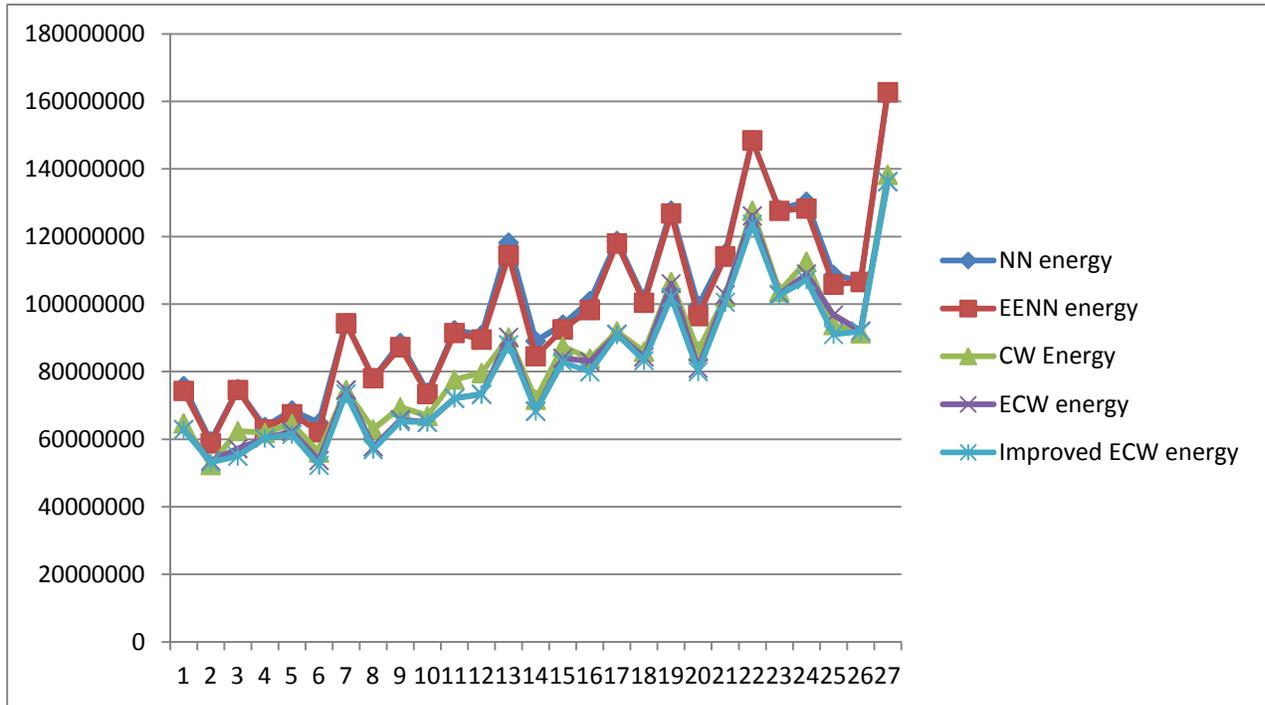


Figure 16: Comparison of five algorithms in terms of energy consumption

Total distance traveled results in Figure 17 also shows that Clarke and Wright family is significantly doing better than Nearest Neighbor algorithms and again the overall patterns of all algorithms are quite similar together and algorithm coding integrity is validated. Improved ECW is presenting the least distance travelled. Comparison of Figures 16 and 17 reveals that the location orders of algorithms' curves on graphs are the same. This again shows the strong correlation that exists between total distance travelled and energy consumption patterns of algorithms.

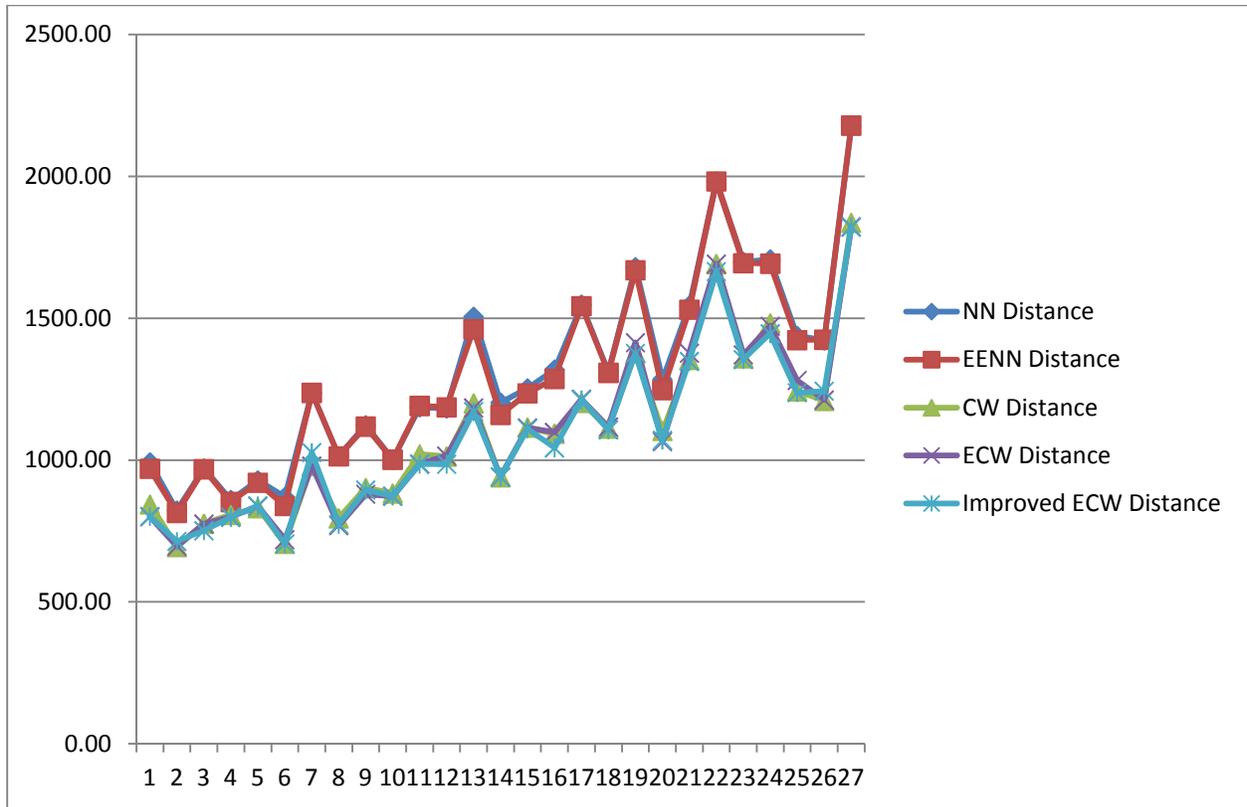


Figure 17: Comparison of 5 algorithms in terms of total distance travelled

5.2 Conclusion and future work

This research proposes two heuristic algorithms, Energy Efficient Nearest Neighbor algorithm (EENN) and Improved Energy Efficient Clarke and Wright algorithm (Improved ECW) to address the Capacitated Vehicle Routing problem. Unlike the traditional approach of distance minimization to the objective function of VRP, energy consumption is considered here as the objective function of proposed algorithms. The energy consumption of the vehicle over a travel from customer i to customer j is defined as the product of gravity constant “ g ”, distance and vehicle weight between cities i and j . The idea that using the theory of “get rid out of heavier customers’ demands as soon as possible” could save on energy consumption is examined in EENN and the computational results approve this theory. Although Nearest Neighbor algorithm

does not belong to the family of high quality solution heuristics, but its modification to EENN by using this theory made a base to improve a high quality solution heuristic, ECW. Improved ECW generally produces better solutions than ECW (an average of 1.44% improvement for 27 benchmark problems), however due to the addition of a parametric term to the saving calculation, the computational times are deteriorated comparing to ECW. Generally if an average quality solution with a fast computational time is desired the EENN is desirable. On the other hand if a high quality solution along with an average computational time is required the Improved ECW is recommended.

The computational results for benchmark problems determined that there is an imperfect, positive correlation between energy consumption and total distance travelled of the vehicles. This imperfect correlation implies that the shortest path is not necessarily the most energy efficient path. However if we could be able to get significantly better solutions in terms of distance traveled by exploiting meta-heuristic procedures, the significant reduction in the total distance traveled could overcome that imperfect positive correlation and the meta-heuristics' solution could outperform the proposed energy efficient heuristics in this paper even in terms of energy consumption. The development of Meta-heuristics with energy consumption cost function could be considered as future work in this research.

REFERENCES

LIST OF REFERENCES

- Altinel, K., & Öncan, T. (2005). A new enhancement of the Clarke and Wright saving method of the vehicle routing problem. Vol. 58, pp. 954-961). The journal of operational research society:(JSTOR).
- Augerat P, B. J. M., Benavent E, Corberan A, Naddef D, Rinaldi G (1995). Computational results with a branch and cut code for the capacitated vehicle routing problem, Technical report RR 949-M, University Joseph Fourier, Grenoble, France.
- Baker, B. M., & Ayechev, M. A. (2003). A genetic algorithm for the vehicle routing problem. Vol. 30, pp. 787-800). Computers & Operations Research:(Sciencedirect).
- Bell, J. E., & McMullen, P. R. (2004). Ant colony optimization techniques for the vehicle routing problem. Vol. 18, pp. 41-48). Advanced engineering informatics:(Elsevier).
- Buxey, G. M. (1979). The vehicle scheduling problem and Monte Carlo simulation. Vol. 30, pp. 563-573). The journal of the operational research society:(JSTOR).
- Charoenroop, N., Boonsong, & Eungwanichayapant, A. (2010). City bus routing model for minimal energy consumption. Vol. 11, pp. 19-31). Asian journal on energy and environment.
- Christofides, N., & Eilon, S. (1969). An algorithm for the vehicle-dispatching problem. Vol. 20, pp. 309-318). Operational Research Quarterly.
- Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. Vol. 12, pp. 568-581). Operations research:(INFORMS).
- Cooper, M. C., Lambert, D. M., & Pagh, J. D. (1997). Supply chain management: more than a new name for logistics. Vol. 8, pp. 1-14). International journal of logistics management.
- Delucchi, M. A., Burke, A., Lipman, T., & Miller, M. (2000). *Electric and gasoline vehicle lifecycle cost and energy-use model*, Institute of Transportation paper UCD-ITS-RR-99-04.
- Eksioglu, B., Volkan Vural, A., & Reisman, A. (2009). The vehicle routing problem: a taxonomic review. Vol. 57, pp. 1472-1483). Computers and Industrial Engineering:(ScienceDirect).
- Gaskell, T. J. (1967). Bases for vehicle fleet scheduling. Vol. 18, pp. 281-295). OR:(JSTOR).
- Gendreau, M., Hertz, A., & Laporte, G. (1994). A tabu search heuristic for vehicle routing problem. Vol. 40. Management science:(JSTOR).
- Gendreau, M., Laporte, G., Musaraganyi, C., & Taillard, E. D. (1999). A tabu search heuristic for the heterogeneous fleet vehicle routing problem. Vol. 26, pp. 1153-1173). Computers

& Operations Research:(Pergamon).

- Ghiani, G., Guerriero, F., Laporte, G., & Musmanno, r. (2003). Real-time vehicle routing: solution concepts, algorithms and parallel computing strategies. Vol. 151, pp. 1-11). European Journal of Operations Research:(Elsevier).
- Giaglis, G. M., Minis, I., Tatarakis, A., & Zeimpekis, V. (2004). Minimizing logistics risk through real-time vehicle routing and mobile technologies. Vol. 34. International Journal of Physical Distribution & Logistics Management:(Emerald Group Publishing Limited).
- Gillett, B. E., & R., M. L. (1974). A heuristic algorithm for the vehicle dispatching problem. Vol. 22, pp. 340-349). Operations research:(JSTOR).
- Golden, B. L., Raghavan, S., & Wasil, E. A. (2008). *The vehicle routing problem: latest advances and new challenges*. Operations Research/Computer Science Interfaces Series Springer, New York.
- Gusikhin, O. Y., MacNeille, P., & Cohn, A. (2010). Vehicle Routing to Minimize Mixed-Fleet Fuel Consumption and Environmental Impact. Vol. 1, pp. 285-291). Proceedings of ICINCO.
- Holmes, R. A., & Parker, R. G. (1976). A vehicle scheduling procedure based upon savings and a solution perturbation scheme. Vol. 27, pp. 83-92). Operational research quarterly (1970-1977):(JSTOR).
- Kara, I., kara, B. Y., & Yetis, M. K. (2007). Energy Minimizing Vehicle Routing Problem. In A. Dress, y. Xu & B. Zhu (Eds.), *Combinatorial Optimization and Applications, LNCS* (Vol. 4616, pp. 62-71). Heidelberg: Springer Berlin.
- Lambert, D. M., & Cooper, M. C. (2000). Issues in supply chain management. Vol. 29, pp. 65-83). Industrial marketing management:(Elsevier).
- Lambert, D. M., Cooper, M. C., & Pagh, J. D. (1998). Supply Chain Management: Implementation Issues and Research Opportunities. Vol. 9, pp. 1 - 20). International Journal of Logistics Management.
- Laporte, G. (1992). The Traveling Salesman Problem:An overview of exact and approximate algorithms. Vol. 59, pp. 231-247). European Journal of Operational Research:(Elsevier).
- laporte, G. (1992). The vehicle routing problem: an overview of exact and approximate algorithms. Vol. 59, pp. 345-358). European journal of operations research:(Elsevier).
- Mirzaei S. and Krishnan, K., 2011, MINLP Formulation of Energy Efficient Vehicle Routing Problem, proceeding of IERC, Reno, Nevada.
- Mole, R. H., & Jameson, S. R. (1976). A sequential route-building algorithm employing a generalised saving criterion. Vol. 27, pp. 503-511). Operational research quarterly (1970-1976):(JSTOR).

- Paessens, H. (1988). The saving algorithm for the vehicle routing problem. Vol. 34, pp. 336-344). *European journal of operational research*:(ScienceDirect).
- Papadimitriou, C. H., & Steiglitz, K. (1998). *Combinatorial Optimization*. Mineola, NY: Dover Publications.
- Simpson, A. G. (2005). *Parametric Modelling Of Energy Consumption In Road Vehicles*. The University of Queensland, Brisbane.
- Toth, P., & Vigo, D. (2002). *The vehicle Routing Problem*. Philadelphia, PA: Society for Industrial & Applied Mathematics, SIAM.
- Yellow, P. C. (1970). A computational modification to the saving method of vehicle scheduling. Vol. 21, pp. 281-283). *Operational Research Quarterly* (1970-1977):(JSTOR).

APPENDICES

APPENDIX A

MATLAB CODE FOR NN ALGORITHM

```

function [Answer,tdd,tee,T]=capNsquaree2(x,D,w,w0,C)
    % this algorithm is the capacitated VRP usnig NN method. Inputs are x,
    % coordinates of cities, D, demand of cities, C, capacity of the
    % vehicle, w0, tare weight of the vehicle, and w, unit weight of
    % demands.
tic
tee=0;tdd=0;
n=length(x);% coordinates of nodes is a nx2 matrix.
A=ed(x); % is a nxn matrix representing the binary euclidean distance of nodes.
S=A;
Answer=zeros(6);% this is an initial value to initiate the while loop
D1=100; % this is an initial value to initiate the while loop

while sum(Answer(:,6))<sum(D1);
for i=1:n;
    S(i,i)=inf;
end

[a(1,2),a(1,3)]=min(S(1,:)); % start from depot(node 1) and find the min distance in row 1.
a(1,1)=1;
S(:,1)=inf; % make column 1 infinity to ensure that node 1 is not gonna be chosen in next rows.
S(:,a(1,3))=inf; % make column of next selected node infinity to make sure...
D1=D*w; %total weight of each demand
a(1,6)=D1(1,a(1,3));% weight of first selected customer's demand

    m=2;
    while sum(a(:,6))<C % a loop that assigns cities to a vehicle as far as its capacity allows

        [a(m,2),a(m,3)]=min(S(a(m-1,3),:)); %building second and third columns of matrix a.
second column
        % is the value of min distance and third column is the location of
        % it (column in matrix S).
        a(m,6)=D1(1,a(m,3));
        a(m,1)=m;
        S(:,a(m-1,3))=inf; % make selected nodes' columns infinity.
        m=m+1;
        if min(min(S))~=inf; % this is for the last vehicle which the weight of remained cities may be
less than capacity of the vehicle. in this
            % the code will be stuck in the while loop. to get rid out of this
            % problm, a "continue-Break" term is used.
            continue;
        else

```

```

    break
end
end
qq=find(a(:,1)); %counts the number of cities assigned to a vehicle
g=length(qq);

a(g,2)=A(a(g-1,3),1);% the last city assigned to a vehicle goes over the capacity of the vehicle.
    %instead of that, the rout to the depot will be
    %replaced.
a(g,3)=1;
a(g,5)=a(g,2)*a(g,4);
a(g,6)=0;
td=sum(a(:,2)); % total distance traveled
tdd=tdd+td;
a(1,4)=w0+sum(a(1:g,6));
for i=2:g;
    a(i,4)=a(i-1,4)-a(i-1,6);
end

    for i=1:g;
        a(i,5)=a(i,2)*a(i,4);
    end
te=sum(a(:,5));
tee=tee+te;
S(:,1)=A(:,1);
Answer=vertcat(Answer,a);% adds up all the routs of each single vehicle to the "Answer"
a=[];
end
T=toc;

end

```

APPENDIX B

MATLAB CODE FOR EENN ALGORITHM

% This algorithm is based on etour algorithm, proceeds to make routs as
 % far as vehecle capacity allows. When different alfas for the first
 % vehicle is examined, the customers included in the best rout
 % (shortest distance)are excluded from customer set and other routs start to
 % be created.

```

n=length(x);
vv=0;cc=0;
A=ed(x);
S=A;
    for i=1:n;
        S(i,i)=inf;
    end
    U=zeros(n,n);
    for h=1:n;
        U(h,:)=S(h,+)/min(S(h,:));% deviding each row over minum value in row
    end
Answer=zeros(2);D1=100;
while sum(Answer(:,2))<sum(D1);
Matrix = [101,2];
for alfa=0:100;

        D1=D*w; %total weight of each demand
        V=zeros(n,n);
        for i=1:n;
            for j=1:n;
                V(i,)=D1;
            end
        end
        V1=zeros(n,n);
        for i=1:n;
            for j=1:n;
                V1(i,j)=max(V(1,:))/V(i,j); % Deviding max value in each row over all values in row.
            end
        end

        K=((1-alfa/100)*U)+(alfa/100*V1);% V1 is related to weigths and U is related to distances.
        for i=1:n;
            K(i,i)=inf;

```

```

end

b=zeros(n,5);
[b(1,2),b(1,3)]=min(K(1,:)); % start from depot(node 1) and find the min distance in row 1.
b(1,1)=1;
K(:,1)=inf; % make column 1 infinity to ensure that node 1 is not gonna be chosen in next
rows.
K(:,b(1,3))=inf; % make column of next selected node infinity to make sure...
b(1,6)=D1(1,b(1,3));
m=2;
while sum(b(:,6))<C

    [b(m,2),b(m,3)]=min(K(b(m-1,3),:)); %building second and third columns of matrix a.
second column
    % is the value of min distance and third column is the location of
    % it (column in matrix S).(values for distance are not correct
    % since they are added with a portion of demands)
    b(m,6)=D1(1,b(m,3));
    b(m,1)=m;
    K(:,b(m-1,3))=inf; % make selected nodes' columns infinity.
    m=m+1;
    if min(min(K))~=inf; % this is for the last vehicle which the weight of remained cities may
be less than capacity of the vehicle. in this
        % the code will be stuck in the while loop. to get rid out of this
        % problem, a "continue-Break" term is used.
        continue;
    else
        break
    end
end
y=length(find(b(:,1)));

b(y,2)=A(b(y-1,3),1);% the last city assigned to a vehicle goes over the capacity of the vehicle.
    %instead of that, the rout to the depot will be
    %replaced.
b(y,3)=1;
b(y,6)=0;
a1=zeros(n,3);
a1(1,1)=1;
a1(1,2)=A(1,b(1,3));% ( to get real distances from matrix a)
a1(1,3)=b(1,3);
a1(1,6)=b(1,6);
for m=2:y;
    a1(m,3)=b(m,3);
    a1(m,6)=b(m,6);
end

```

```

for m=2:y;
    a1(m,2)=A(b(m-1,3),b(m,3));
    a1(m,1)=m;
end

td1=sum(a1(:,2)); % total distance traveled
a1(1,4)=w0+sum(a1(:,6));
for i=2:y;
    a1(i,4)=a1(i-1,4)-D1(1,a1(i-1,3));
end

for i=1:y;
    a1(i,5)=a1(i,2)*a1(i,4);
end
te1=sum(a1(:,5));
for i=1:101
    Matrix(i,1)=i-1;
end
Matrix(alfa+1,2)=td1;
Matrix(alfa+1,3)=te1;
ROUTES{alfa+1,1}=horzcat(a1(:,3),a1(:,6));

end
[v,o]=min(Matrix(:,2));
disp(v);
vv=v+vv;
[c,d]=min(Matrix(:,3));
cc=c+cc;
best=ROUTES{d,1};
r=length(find(best(:,1)))-1;
for e=1:r;
    U(:,best(e,1))=inf;
end
Answer=vertcat(Answer,best);
best=[];

end

```

APPENDIX C

MATLAB CODE FOR IMPROVED ECW

```

function [route,cost,energy,T,minc,mincp,mine,minep]=EEVRPOCA2(cc,dc,D,cd,vc,viw,u)
tic
cost=inf(51,2);energy=inf(51,1);
for alfa=[0,5,10,15,20,25,30,35,40,45,50];
% cc: cutomer coordinate is a nx2 matrix
% dc: depot coordinate is a 1x2 matrix
%D: is the Euc distance matrix in which depot is the last node
% cd: customer demand is a nx1 matrix
% vc: vehicle capacity in terms of weight
% viw: vehicle initial weight with full capacity
% u: unit product weight (nx1);
% p: profit
% this function gets the above mentione input and returns the best route
% based on the saving algorithm for energy

if (isempty(cc)==0 && isempty(dc)==0)
N=length(cc);
d=[cc;dc];
D=ed(d);
else
N=length(D)-1;
end;
z=zeros(N,N);
for i=1:N;
for j=1:N;
z(i,j)=viw*D(N+1,j)+(viw-u(i)*cd(i))*D(i,N+1)-...
(viw-u(i)*cd(i))*D(i,j)+u(i)*cd(i)*D(j,N+1)...
+alfa*10*(cd(i)*u(i)+cd(j)*u(j))/sum(cd)*u(i);
end
end
L=zeros(N*N,3);
a=1;
b=N;
k=1;
while k<=N; % put all the savings in order to sort them out later
L(a:b,1)=z(:,k);
L(a:b,3)=k;
a=b+1;
b=b+N;
k=k+1;
end

```

```

a=1;
b=N;
% first coloumn of L is savings , second and third coloumn are (i,j)
% respectively.
while b<=N*N;
    k=1;
    for i=a:b;
        L(i,2)=k;
        k=k+1;
    end
    a=b+1;
    b=b+N;
end
i=1;
k=1;
G=zeros(N*N-N,3);
while i<=N*N %elimintaing the (i,j) when i=j;
    if L(i,2)~=L(i,3);
        G(k,:)=L(i,:);
        k=k+1;
    end;
    i=i+1;
end

[x,i]= sort(G(:,1),'descend');
B = G(i,:);
% first coloumn of B is the sorted saving, 2nd and 3rd coloumns
% are the related link i-->j

n=N*N-N;
R=zeros(N,N);
i=1;
r=zeros(N,1);
rd=zeros(N,1);
iec=0;

while (i<=n && iec==0);

    [p1,p2]=find(R==B(i,2)); % max saving, x coordinate
    [p3,p4]=find(R==B(i,3)); % max saving, y coordinate
    iep1=isempty(p1);
    iep3=isempty(p3);
    Lp1=length(find(R(p1,:)));
    Lp3=length(find(R(p3,:)));
    C=find(r==0);
    k=C(1);

```

```

    % if in link (i,j) non of them are assigned
if (iep1==1 && iep3==1 && (rd(k)+u(B(i,2))*cd(B(i,2))+ ...
    u(B(i,3))*cd(B(i,3)))<=vc)
    R(k,1)=B(i,2);
    R(k,2)=B(i,3);
    % if in (i,j) i is assigned and j is not assigned and i is at the
    % end of the route
elseif (iep1==0 && iep3==1 && Lp1==p2 && (rd(p1)+u(B(i,3))*cd(B(i,3)))<=vc);
    R(p1,p2+1)=B(i,3);
    %if in (i,j) i is not assigned and j is assigned and j is at the
    % beginning of the route
elseif (iep1==1 && iep3==0 && p4==1 &&(rd(p3)+u(B(i,2))*...
    cd(B(i,2)))<=vc);
    j=Lp3;
    while j>=1
        R(p3,j+1)=R(p3,j);
        j=j-1;
    end;
    R(p3,1)=B(i,2);

    % route mergers
    % if both are assignend, i at the end of one route
    %and j in the beginning of another route
elseif (iep1==0 && iep3==0 && Lp1==p2 && p4==1 && ...
    p1~=p3 && (rd(p1)+rd(p3))<=vc);
    for l=1:Lp3;
        R(p1,Lp1+1)=R(p3,l);
        R(p3,l)=0;
    end
end
end

% defining a new matrix RD, as the demand matrix of the routes
% cd(R) means instaed of routes their demands weight are considered.
i=i+1;
RD=zeros(N,N);
for o=1:N;
    for oo=1:N;
        if R(o,oo)~=0;
            RD(o,oo)=cd(R(o,oo))*u(R(o,oo));
        end
    end
end
end
rd=sum(RD,2);
for I=1:N; %in this loop we find the taken and not taken routes
    e=find(R(I,:), 1);
    if isempty(e)==1;

```

```

        w=0;
    else
        w=1;
    end;
    r(I,1)=w; % if r(i,j)=1--> route i is taken, else its not
end
iec=isempty(find(r==0, 1));
end

% calculating cost and energy
f=R;
P=zeros(1,N);
k=1;
%finding the customers not assigned in the first attempt
for i=1:N;
    e=find(f==i, 1);
    if isempty(e)==1;
        P(1,k)=i;
        k=k+1;
    end
end
% if there is any not assigned customer then put them in another route
% and attach it to the set of routes
if isempty(find(P,1))==0;
    F=[f;P];
else
    F=f;
end

%finding the number of routes in the F
NZ=find (F(:,1));
% V will be equal to the number of routes
V=length(NZ);
% cost of the network will be calculated in C matrix
C=zeros(V,1);
% energy of the network will be calculated in E matrix
E=zeros(V,1);
for i=1:V;
    n=length(find(F(NZ(i),:)));
    j=1;
    d=0;
    co=0;
    e=0;
    while j<=n-1
        C(i,1)=co+D(F(NZ(i),j),F(NZ(i),j+1));

```

```

E(i,1)=e+(viw-(d+u(F(NZ(i),j))*cd(F(NZ(i),j))))*D(F(NZ(i),j),F(NZ(i),j+1));
d=d+u(F(NZ(i),j))*cd(F(NZ(i),j));
j=j+1;
co=C(i,1);
e=E(i,1);
end
% energy/cost form depot to the first customer on the route and form
% the last customer to the the depot
C(i,1)=C(i,1)+D(N+1,F(NZ(i),1))+D(F(NZ(i),n),N+1);
E(i,1)=E(i,1)+D(N+1,F(NZ(i),1))*viw+D(F(NZ(i),n),N+1)*(viw-
(d+u(F(NZ(i),n))*cd(F(NZ(i),n))));

end

cost(alfa+1,1)=sum(C);
cost(alfa+1,2)=alfa;
energy(alfa+1,1)=sum(E);
energy(alfa+1,2)=alfa;
route=F;
end
T=toc;
[minc,mincp]=min(cost);
mincp=mincp-1;
[mine,minep]=min(energy);
minep=minep-1;
end

```