An Investigation of Inventory Routing Problem with Carbon Emission and Sustainability Consideration

by

Krishna Priya Muthu

August 31st, 2021

A thesis submitted to the
faculty of the Graduate School of
the University at Buffalo, The State University of New York
in partial fulfillment of the requirements for the
degree of

Master of Science
Department of Sustainable Transportation and Logistics

(This page is intentionally left blank)

Acknowledgements

First, and foremost, I would like to express my deep and sincere gratitude to my advisors Dr. Nallan Chakravarthy Suresh, and Dr. Charles X. Wang for their invaluable knowledge, support, and time to guide me through my research during one of the toughest times in this century.

Secondly, I cannot express enough thanks to the Graduate Coordinator, Kaeleigh Peri for her generosity to share with me the updates, and every little information possible throughout the period of my Masters education at University at Buffalo.

I would also like to extend my gratitude to the University at Buffalo, the State University of New York, for providing all the library facilities, with special protocols during COVID-19 pandemic, and Director of the School of Sustainable Transportation and Logistics, Dr. Stephen Still, for providing me this precious opportunity.

Abstract

This thesis develops a mathematical programming model of inventory routing problem, where a depot supplies products to its several customers over a specific time horizon. The depot has an unlimited supply of products that could be distributed to the customers through its unlimited number of vehicles. The optimization model includes decision variables pertaining to optimum stock levels and optimum transportation routes, with the objective function of minimizing total costs of inventory and transportation in the two contexts: 1) vendor managed inventory (VMI) and 2) retailer managed inventory (RMI). In addition, the study also considers the carbon emission associated with the distribution of products in traditional routing problems and inventory routing problems. The model yields optimum routes from supply sources to retailers.

To solve the mathematical programming minimization model, CPLEX solver is used with python application programming language and analytical libraries. The distances associated with the models are calculated and plotted to evaluate carbon emissions. This comparative study of the two models (VMI and RMI) helps us understand the value of supply chain integration.

This study also provides for a comparison of carbon emissions from existing insights and life cycle assessments associated with different fuels in the market. In addition, this study develops appropriate mathematical models, python code, algorithms of the code and the explanation of code to solve this optimization problem. Preliminary comparisons of emissions show that electric trucks with lithium ion batteries prove to be more efficient than the trucks running on hydro-carbon fuels. The carbon emission associated with electric vehicles proves to be much less compared to the gasoline vehicle by lifecycle assessments made in the recent decade. Carbon emission associated with gasoline vehicles, electric vehicles, hydrogen vehicles are tabulated for comparison.

TABLE OF CONTENTS

CONTENTS

List of Tables

List of Figures

ABBREVIATIONS

API: Application Programming Interface

CH4: Methane

CO2: Carbon dioxide

CPFR: Collaborative Planning, Forecasting, and Replenishing

EPA: Environment Protection Agency

FCEV: Fuel Cell Electric Vehicle

GHG: Greenhouse Gas

IP: Integer Programming

IRP: Inventory Routing Problem

MILP: Multi Integer Linear Programming

NumPy: Numerical Python

OECD: Organization of Economic Cooperation and Development

EOQ: Economic order quantity

Pandas: Python Data Analysis Library

QR: Quick response

RMI: Retailer Managed Inventory

VMI: Vendor Managed Inventory

VMIR-ML: Vendor Managed Inventory- Maximum Level

VMIR-OU: Vendor Managed Inventory Order Up to Level

Chapter 1
INTRODUCTION

CHAPTER – 1

INTRODUCTION

This study is based on evaluating the advantages of integration and illustrating levels of carbon emission. The advantages of integration in the logistics industry is innumerable. It leads to reduced costs, improved supplier-customer relationship, and thus an overall improved business performance. When a depot, 0, has unlimited supplies of good to distribute to the customers, n, with its unlimited number of vehicles, with certain conditions, the stock out is reduced and excess inventory is reduced.

The second part shows that the greenhouse gas level that soared from pre industrialization era to the current year, with the transportation sector contributing to 29% of the CO_2 emissions. The study presents an alternative to minimize the effect vehicles have on the environment. The carbon emission from vehicles in the transportation sector needs to be curtailed. This study also illustrates the optimization of routes and assessment of emissions in various stages of a vehicle being used from manufacturing to exhaust pipe.

Research Questions

This study of the mathematical model dives in to evaluate distances with the implementation of constraints using the Optimization algorithm. Euclidean distances are considered between the depot and customers and among the customers in order to solve. The algorithms and codes used are provided later in the thesis. Furthermore, illustrations of the emissions from various other possible freight choices are provided to arrive at a sustainable alternative. The study is focused on the following questions:

1. What is magnitude of improvement of VMI over RMI?
2. How is the distance calculated in both policies?

3. Is carbon emission controllable?
4. What is the carbon emission associated with conventional trucks, electric trucks, and hydrogen trucks?
5. Can a sustainable option work out by moving to a particular type of vehicle?

Motivation

The motivation for this study is the importance of minimizing inventory cost, logistics cost and the damage to the environment caused by vehicle emissions by the freight trucks, needs attention.

Structure of Thesis

Chapter one elaborates on the description of the problem itself by describing the situation, the assumptions made, and describes an overview of the problem. The solver software that is utilized in the study is also highlighted in the chapter along with the specific mathematical model that is to be implemented. This chapter includes why the topic was chosen and the motivation behind it and the structure of how the thesis has been planned out in to a thesis.

Chapter two includes the definitions of inventory policies, capacitated vehicle routing problem, inventory routing problem with respect to American Production and Inventory Control Society (APICS) dictionary; it also covers the insights from papers related to Vendor Managed Inventory (VMI), and Retailer managed Inventory (RMI) policies, the benefits of implementing the policies that helped the study. Suppliers have clarity about customer demand in the integrated process. This understanding due to sharing of data is bound to make the relationship and the business stronger. Eventually this implies an improved performance. The hypotheses framed by companies based on size of the industries are discussed as well. This helps in improved customer service levels, over all. The paper on Inventory Routing problem encompasses the study and solution of the problem with multiple customers to evaluate the amount of savings generated with respect to traditional policy. The explanation is provided with relation to the travelling salesman problem. The article and papers referred for the study provided the details, assessments, researches and results under various methods.

The authors have also provided explanations for the data gathered for the three-day and six-day horizon. In order to solve the problem, IBM ILOG CPLEX optimization software is used. The CPLEX name for the optimization software was coined from the method simplex; the optimization software was later acquired by IBM, hence the name. The software can solve linear programming, constraint programming, quadratic programming mixed integer linear programming problems. The problem here is a mixed integer linear programming problem. One of the two decision variables, the units of products supplied is an integer, and the decision variables in the math model are multiplied by constants, therefore the problem is linear The solution of such a problem is built using Python along with CPLEX constraints. The solver uses Branch-and-bound algorithm

Chapter three elaborates the problem description in detail. This chapter is called methodology. Here is the detailed explanation of the parameters, evolution of the mathematical model, the decision variables, constants, problem objectives involved, formulation of demand on each day of the horizon. This section also provides the data collected and the source.

Routing constraints, demand constraints and other limitations are expressed in mathematical form; furthermore, the percentage of carbon emission involved via vehicles as per the guidelines on environment protection agency of U.S is gathered. Research papers that provided manufacturing emissions was gathered. Specifications of Tesla Semi, Nikola hydrogen powered vehicles, gasoline freight trucks have also been added for lifecycle carbon assessment. Various methods of Hydrogen productions is shared as a sustainable option by one of the authors. This chapter also tabulates carbon emission in both policies and for various vehicles of similar capacity.

Chapter four is the section that elaborates the program, shares the algorithms, the codes for both policies, and data collected. This section explains the analytical libraries, matplotlib and NumPy implemented in the code. This section shares the data for ten instances from Dr. Speranza's paper in 2007.Here, the instances are explained, the input, calculations, output are explained with graphs, and output results which includes calculation of distances using CPLEX with python application language.

The validation for an instance is also included in the section along with output results. The tabulated results for RMI and VMI are plotted with histogram graphs on tableau. Eventually the histogram graphs for all fuel vehicles are evaluated to prove what sustainable alternative.

CO2 emission is broken into CO2 involved in manufacturing, CO2 in exhaust pipe. Battery manufacturing, electricity production to run an electric truck and also the life cycle assessment for each type of vehicle.

Chapter 5 shares the conclusion from the results received. It provides a framework of the work carried out throughout the study and the problem description, and carbon assessment. References are included at the end of the thesis in Chicago APA style. The references used to do the study, build model, have all been added.

The findings of the research:

The production of lithium ion batteries alone is expensive on the environment. 282600 kgs of CO2 is emitted from the mining and making of lithium ion batteries for one vehicle. In order to reduce the impact of batteries on environment, repurposing could reduce the damage.

The emissions calculated, and plotted involve the life cycle CO_2 emission of that vehicle. Evaluated results indicate electric cars can be a savior but also hydrogen produced by solar splitting would be much less. Using the freight vehicle promptly for its life of 1000000 miles would prove to be immensely helpful on the environment.

Limitations:

- This mathematical program is coded for a few number of customers and would not show reasonable answers for a wider network of customers.
- The calculation of distance between customers is based on Euclidean distance, hence this cannot be implemented on real world problems.
- This is not connected to global positioning system or live traffic data, hence this has very limited usage.
- The carbon emission from different stages did not include all of the stages. Stages such as transportation of fuels, the fuel storage infrastructure also contribute to the harmful emissions.

Chapter 2
Literature Review

CHAPTER – 2

LITERATURE REVIEW

2.1: Inventory Policies

2.1.1: Vendor Managed Inventory Policy

The American Production and Inventory Control Society (APICS) dictionary defines vendor-managed inventory as "A means of optimizing supply chain performance in which the supplier has access to the customer's inventory data and is responsible for maintaining the inventory level required by the customer. Resupply is performed by the vendor through regularly scheduled reviews of the on-site inventory. The on-site inventory is counted, damaged-or-outdated goods are removed, and the inventory is restocked to predefined levels." This policy is also referred to as the integrated inventory policy. Here, based on the integrated data available to the supplier, the supplier himself decides on the time of delivery, quantity to be delivered, and transportation routes to the customers.

2.1.2: Benefits of VMI

Author studies the benefits of VMI in the electronic industry in this research. Four research hypotheses were framed related to the size of organization, logistics, and employee involvement. The "efficient consumer response" in the grocery industry, and "quick response" in the garment industry became significant in the supply initiative. It was implemented by Procter & Gamble, and Wal-Mart.

It is known from industry economics that the key to success is the relationship between supplier and customer. VMI, when implemented had its advantages indubitably such as reduction in safety stock, increase in customer service levels, reduction in cost. There were adverse effects due to limitations such as planning complexity, high costs associated with administration, ineffective ordering and fulfillment, and failing to harness customer specific data. Further, priority treatment brings shortages.

2.1.3 Traditional inventory policy

The paper, Inventory routing problem: The value of integration (Claudia and Speranza, 2015) states s,S is a minimum/maximum inventory policy also called traditional inventory policy or Retailer Managed Inventory (RMI) policy. When the inventory level falls below s, the minimum level, the retailer/customer himself will generate a request for replenishment order that will restore the on-hand inventory to a target, or maximum level, S.

Here, the Reorder Point field is the minimum, or minimum level which triggers an action to replenish the stock up to the Reorder/Order up to quantity field, the maximum, or the number to which the inventory level is restored. The route is optimized for each day in the planning horizon based on the delivery and customers who need replenishment to optimize the transportation cost.

2.1.4 The value of Integration

The paper, Inventory routing problem: The value of integration (Claudia and Speranza, 2015) analyzed the traditional S, s inventory policy and integrated policy. In this paper, inventory routing optimization model for integrated policy, and a classical capacitated vehicle routing optimization problem for S, s policy is solved. Then the costs and characteristics for both policies are analyzed. This paper was aimed at solving problems for instances with 3-day, and 6-day horizons over which the stocks at 10, 20, 30 customers are replenished with no stock out.

This paper proved that VMI outperforms RMI, and showed that with integrated inventory policy, remarkable savings can be achieved. The authors, Luca Bertazzi and Grazzia Speranza, introduced the inventory routing problems and its importance.

2.1.5 Capacitated Vehicle routing problem

According to Inventory routing problem: The value of integration (Claudia and Speranza, 2015), a vehicle routing problem is where optimal transportation routes to traverse are determined to deliver goods to a set of customers. The objective of the vehicle routing problem is to minimize the cost of route traversed. When the vehicles have a limited carrying capacity for the goods that need to be delivered, such a problem is known as a capacitated vehicle routing problem.

2.1.6 Inventory Routing Problem

A type of routing problem in which a decision is made to determine how much to deliver each customer per day and organize the routes to replenish the stock such that there is no stock out. The solution should provide sufficient quantity for customers to cover the consumption each day and also to satisfy the capacity in each route. The objective of this problem is to minimize the routing cost as well as the inventory cost.

This paper called Inventory Routing Problems (by Bertazzi, Speranza, 2012) explores on different IRPs, and their next paper also explores Inventory routing problem with multiple customers (Bertazzi and Speranza, 2013) with explanations in relation with classical routing problems. This method is extended to problems with multi vehicle cases. It also explains the travelling salesman problem and explains IRP in relation to it.

"A travelling salesman problem is the most widely used combinatorial optimization problem. In a complete undirected graph with a cost associated with each edge, the problem is to find a circuit of minimal cost that visits all vertices."

This paper has a simple explanation of the following to introduce a topic:

- Modelling a single vehicle IRP along with decision variables.
- A transportation problem

- An Inventory routing problem
- A production planning problem

The paper (by Coelho and Et al, 2013) discusses the origin of IRP from 1983. This will discuss heuristics to take inventory costs into account, in addition to vehicle routing problems. This literature review takes into consideration the routing problem, its variants, models and algorithms. The evolution of IRP and its continuous development made environmental operations efficient. This is a rich field for researching further in depth.

2.1.7 IBM ILOG CPLEX Optimization Studio

CPLEX is an optimization software package used for solving mathematical models. The solver was named after the simplex optimization method but it offers a variety of algorithms(*CPLEX Optimizer*. (2020, December 4). https://www.ibm.com/analytics/cplex-optimizer). This solver package has high performance algorithms for very large mathematical programming such as linear programming, constraint programming, quadratic programming, mixed integer linear programming, quadratic constraint, etc.

The problem to solve here, is a mixed integer linear programming model for the routing optimization, and inventory and routing optimization. A python language application programming interface (API) is used here for modeling the mixed integer linear programming (MILP) problem. An IBM ILOG decision optimization library docplex.cp is called for solving.

2.1.8 Linear Programming model

This paper focuses on minimizing the route to reduce carbon emission. This includes constraints to minimize the inventory, and distance travelled to serve a set of customers from the supplier's depot and the carbon emission associated with distance traversed when vehicles with fossil fuels, Hydrogen cells from renewable resources, and when electric vehicles are implemented.

2.1.9 Advantages of Vendor Managed Inventory

Vendor managed inventory, or VMI, through its prediction helps businesses improve performance, supply chain streamline, their production processes and improve financial performance.

Suppliers develop a better understanding of actual customer demand through VMI process. The reduced communication gap and improved transparency reduces wastage of resources, and excess inventory. VMI inevitably makes the supplier and customer great business partners.

The VMI process facilitates better understanding between supplier and customer's demand. This understanding helps to achieve increased performance and sales. The encouragement achieved through this transparency helps both parties grow.

2.1.10 Branch-and-Cut Algorithms for Multi-vehicle Production and inventory routing

Authors Yossiri Adulyasak, and Jean Francois Cordeau discuss multi-vehicle problems in integrated supply chains in which problems are jointly optimized for production planning, and inventory routing costs. A typical supply chain has a series of activities of production, storage, and distribution. The production planning includes optimization of lot sizes which eventually reduced production cost and the inventory cost with decision variables and constraints associated. Here, the supply chain is treated as a whole operation, hence making much more beneficial.

Inventory routing problem optimizes replenishment quantity and the distribution routes. In addition to the inventory routing problem, when optimizing lot size is added to the model, it transforms IRP into a Production Routing Problem.

To solve both PRP and IRP, this paper introduces multi vehicle formulations for Order Up to Level(OUL), and maximum level(ML). Only a few algorithms can solve IRP because of its complexity. Archetti developed an algorithm called branch and cut approach with single vehicle and ran analysis for three replenishment policies. One, where replenishment was set to Order-up to level, the second where replenishment was up to Maximum Level, and in the third policy where quantity replenished for customers with no stock maximum level. Branch and cut approaches include vehicle index formulation, and non-vehicle index formulation where vehicle tours are indexed with vehicle index.

To find an upper bound for branch-and-cut Optimization-based adaptive large neighborhood search. Heuristic, optimization-based adaptive large neighborhood search heuristic is used. Computing time was set to an hour, and the experiments were run for instances with 15 customers or fewer.

2.1.11 Vehicle Routing Problems Variants

The authors, Paolo Toth and Daniele Vigo have shared the methods to solve transportation problems also known as Vehicle Routing Problems (VRP), and their applications in this book. These problems constitute routing requests, fleet of vehicles, related expenses and the feasibility, and may also vary based on constraints such as time window limits. By planning transportation routes using computerized solvers for real world routing solutions, a significant sum of money is known to have been saved. Cost savings and efficient implementation of the fleet are two of the most important factors that help in saving money along with acceptable computing time.

The most famous academic method is the capacitated vehicle routing problem. In this method the constraint is capacity of the vehicle.

This problem is considered to be an integer programming (IP) or mixed integer programming (MIP) problem. This is solved with a latest solver technology or a branch-and-cut algorithm. There are variants of VRP such as a compact formulation, an extensive formulation for a CVRP. Extensive formulation coined by Balinski and Quandt. The basic nodes to work with are the feasible one. Here, the options of all combinations of arcs are considered, and the minimum is chosen.

The various kinds of VRP depend on the planning period, the transportation requests, constraints affecting route, optimization objectives, network structure, fleet composition and depot location. The book sheds light on how VRP differs from Arc routing problem (ARP), General routing problem(GRP) and it also discusses what algorithm and constraints are applied for what real world situations such as methods used for mail delivery, simultaneous delivery and pickup that optimize the resources spent.

2.1.12: Benefits of Retailer-supplier relationship

This paper by Tyan and Wee (Tyan, Wee, 2002) illustrates retailer-supplier relationships and explains how powerful it is to reduce inventory levels and costs, to improve service levels. This paper also discusses why the VMI is a strategic alliance and the applications of these strategies can lead to future growth of opportunities.

Improving supply chain helps reduce unnecessary expenses and thus adds to profit, making it a strategic alliance. When supply network creates profit in every level as well as helps its suppliers, distributors, and logistic partners grow, it makes an alliance a successful one. Such an alliance has a strong future and long-term benefits. With novel production strategies such as design to order and make-to-order, the forecasting is not as easy to speculate. We also have exaggerated levels at suppliers due to the bullwhip effect while forecasting. A relationship between supplier and customer would greatly reduce excess cost in an alliance.

VMI used by technological industries like Dell, HP, and its close competitors also steer on efficient supply chains. This inevitably means a close monitoring that occurs within the relationship between supplier and retailer improves growth for all involved in the supply chain and reduces variance. This coordination is healthy for the growth for every level and department in a network.

This paper suggests the strategic alliance, partnership, and coordination in Taiwanese grocery industry will be successful just like in the Tech industry.

2.1.13: Importance of Coordination in supply chain

In a supply chain, a number of activities between organizations and departments occur. Since the process of supplying the right quantity to the consumer is the goal of every organization, relationships between the organizations make it possible. This coordination between various ethos of the network enhances the performance on the whole. The paper Supply chain coordination: Perspectives, empirical studies, and research directions (by Arun Kanda, S. G. Deshmukh, 2006) discusses the importance of harmony in supply chain networks and also what technology enhances it.

The most commonly accepted definition of coordination in the literature is "the act of managing dependencies between entities and the joint effort of entities working together towards mutually defined goals" (Malone, and Crowston, 1994).

Material flow, logistics, production planning, purchasing, forecasting inventory, quality assurance, coordination, partnership makes supply chain a dynamic structure. The coordination in this kind of progressive, expanding structure brings in with it challenges in communication and coordination. Independent members do not provide growth in the relationship. There needs to be explicit coordination, and bring together the functions. There needs to be coordinated behavior between the members, organizations, departments in the system and any lack of it jeopardizes the chain.

Coordination across organizational boundaries is a performance enhancer in the supply chain industry. To avoid damage to the network, Lack of managerial quality such as communication skills, and harmony with entities will need to be overcome with effective communication, sharing information, and integration.

Coordination mechanisms such as Supply Chain Contracts that include quantity flexibility, quantity discounts, revenue sharing and buyback; Information Technology such email, internet, Enterprise resource planning software, Point of Sale data; Information sharing such as demand, inventory, information, cost, production schedule, lead time, cycle time; Joint Decision making in replenishment, forecasting, ordering in cost consideration.

The performance ratio is determined with respect to the performance without coordination. Many companies are unaware of dynamics that help SCC (Supply Chain Coordination) performance. These performances could be studied much more deeply, or in combinations, with the help of simulation and fuzzy logic theorems.

2.1.14: Complications in a chain

Outlook of Inventory management combined with routing is being discussed in this paper (by Anderson, 2010). The complications faced in the industry is concentrated on as well

Due to competition in the sector, there is dire need for efficiency that will bring in profits. On the whole the goal is to bring harmony between the units in an organization, and coordinate material flow, production flow, sales, transportation. When transportation units act responsibly by including inventory management a better service quality is achieved. The paper discusses problems in inventory management when combined with routing.

When there exists interdependencies between Inventory management and transportation, there has to be benefits theoretically. Compared to rail and airborne transportation, maritime and road transport come in combination as they are widely used. He has classified the problem, how it impacts the industry and possible solutions.

2.1.15: Power of Information Technology

The power of information technology is studied through this paper(Yu et al., 2001). With the help of information technology, decentralized systems, the supply chain diminishes the problems associated. This decentralized control reduces the bullwhip effect. An increased information sharing among the members improves the industry's chain to Pareto efficiency.

Due to the rapid technology development in manufacturing, information, globalization of business has been expeditious. Manufacturers have moved to newer designs to match the fast pace and moved from conventional structure. As stated, the Supply chain creates win-win situations for all the members. All the latest strategies supply chain VMI, Cross docking, or quick response help advance the business.

Author classifies sources of uncertainty as (i)Suppliers, (II) manufacturer, (III)customers. Uncertainties are caused by machine breakdown, delayed deliveries, and order fluctuations. This in turn amplifies safety stock and resources, thus increasing inefficient uses of resources.

In the supply chain, every member makes a forecast of demand downstream for production planning, inventory control and material requirement planning. This forecast shows a higher variability at the upstream, which dwindled down as it reached downstream. This phenomenon observed is the variability of an upstream member's demand is greater than that of the downstream member. This was found by logistic executives in Proctor and Gamble. This was called the bullwhip effect. The cause of the bullwhip effect is known to have been price fluctuation, demand forecasting, and rationing.

Lack of uncertainties leads to imperfect information, and imperfect information brings deficiencies. To achieve mitigation in deficiencies, a decentralized supply chain can bring perfect information, and thus reduce uncertainties.

The paper has investigated three types of information sharing. It consisted of a two level decentralized supply chain made with retailers and manufacturers. The three types of information sharing were:

Level 1: Decentralized control where there was no information sharing. Every member determined the forecast. This is also called the traditional or (s,S) policy.

Level 2: Coordinated control where both retailer and manufacturer interact. Manufacturer obtains the information of customer demand from retailers and forecasts future demand.

Level 3: Centralized demand where manufacturer can retrieve exact customer information through VMI, and have optimal replenishment decisions and

The results show inventory reduction, cost savings, and also that Pareto improvement is achieved.

This paper by Sari (2008), studies performance improvement obtained from VMI and Collaborative Planning, forecasting, and replenishment (CPFR) under stationary in a four level supply chain. Through comprehensive experiments and statistical analysis of simulated outputs, some important conclusions were made. Firstly, CPFR produced a lower supply chain cost and greater service level. And therefore, companies should invest in CPFR. Secondly, the performance gained in both CPFR, and VMI depend on replenishment time, capacity tightness of the plant, and uncertain demands. Finally, as the uncertainty in demand increases, the VMI performance decreases. Supply chain members may manage the uncertainties through inventory planning, and joint forecasting.

Here authors Claudia Archetti and et al. study an inventory routing problem using Branch and cut methods. The problem in the paper, Archetti et al. (2007) is solved for optimal solution and the next time with a relaxation. The problem considered is to distribute a product from a supplier to *n* number of customers over a time horizon. The supplier keeps track of inventory at each customer and replenishes the quantity such that it reaches maximum inventory level. The problem needs to determine the quantity to be delivered, the route for every discrete time instant. The problem is a mixed integer linear programming problem. A branch and cut algorithm is proposed to solve this inventory routing problem with a single vehicle that replenishes based on deterministic order-up to-level.

An objective function for the problem is modelled to minimize the cost in the process which includes inventory holding cost at supplier, routing cost, inventory holding cost at retailer. This is followed by modelling constraints such as no stock out at the customer, stock out constraint at the supplier, inventory definition for each day, order up to level constraints, capacity constraints, routing constraints, and finally non-negativity constraints for the problem to determine the optimal results.

First the problem is solved for VMIR-OU (Vendor Managed inventory order up to level)level, next it is solved with relaxation called VMIR-ML(Vendor Managed inventory maximum level). Classical vehicle routing problems have been commonly used but inventory routing problem was a novel solution to reduce cost and improve service level around the globe. This article proves the inventory routing problem saves significant money in the supply industry.

2.1.16: Production routing problem with optimization-based adaptive large neighborhood search:

Classical problems of lot sizing and vehicle routing have been around for about 5 decades. Kellogg with its production plan earned a savings of 40 million pounds. An efficient heuristic using an adaptive large neighborhood search . The experiment with instances from Archetti, 2011, and Op-ALNS provides a solution better than former heuristic approaches on benchmark. This approach could be implemented where multi commodity, multi customers are present in the scenario in PRP.

2.2 Sustainability

2.2.1 Sustainability in supply chain:

The supply chain connects the resources, services, and products with consumers across the world at the cost of the environment. Thus, a sustainable approach to have a reduced carbon footprint, would be necessary in the industry for our future.

2.2.2 Climate Change

In the 171 years since 1850, the pre-industrial time, combustion of fossil fuels has caused a spike of 48% of carbon dioxide in the atmosphere. The research article from N. G. C. (n.d.). *Carbon Dioxide Concentration | NASA Global Climate Change*. Climate Change: Vital Signs of the Planet. https://climate.nasa.gov/vital-signs/carbon-dioxide states carbon dioxide content in the atmosphere was known to increase much slowly from Late Glacial Maximum, which was 185 ppm, over 20,000 years ago to 280 ppm in 1850. The current concentration is 416 parts per million in 2017.

Hence combating climate change by swapping for carbon-neutral resources instead of the fossil fuels in transportation would mitigate carbon emission. It would be a great step towards sustainability and to control climate change.

2.2.3 Transition to cleaner fuel

In the paper by Tanay Sidki Uyar, and Dogancan Besikci, 2016, the authors want to communicate the importance, necessity, possibility to transition into implementing electricity and fuel cells made from renewable sources. The investment on renewable energy until 2015 was 290 billion. This continuous increase in investment since 2013 will help lower the production cost of hydrogen and divert from implementation of fossil fuels.

Countries all over the world are realizing there would be no sustainable future if there is no transition from utilizing non-renewable resources. The authors also state that to reach a solution, the complications involved with respect to energy need to be defined, to find feasible solutions that can be implemented, to set apart differences between best available technology for energy, to stop importing resources and utilize locally available resources.

A future with only clean energy is possible with consent and involvement of stakeholders, and approval from local people. A total transition towards a nuclear-free energy, and hydrogen as fuel implies a healthy environment. With economy and ecology support tools, further long term solutions can be decided on future sustainable societies. Programs such as Energy Star, German energy Transition, and Revolution now are followed by a few countries which realized the problems, and thus they are taking initiatives to reduce the size of the problem. After the oil crisis in the 1970s, usage of other energy technology progressed at a faster pace, it did not continue to do so. Fuel cell because of its compactness, cleaner emission, and safety, the technology is considered practical and comparatively little time to refuel.

In order to compete with the current fuel industry, the production rate of hydrogen needs to increase to accommodate all the vehicles around the globe, and then the subsequent problems are ways to store and transport it.

The wind potential is locally available in Turkey and is suggested by the authors as a productive source. It is also suggested that smaller towns and islands could utilize solar energy and wind energy to electrify.

CHAPTER 3
Methodology

CHAPTER – 3

METHODOLOGY

3.1: Problem description

Assume that there are n customers, also called retailers, to whom a product needs to be shipped and a supplier from a depot, denoted by *0(zero)* is responsible for the shipping. From the assumption, it is evident that every route, every route starts and ends at the depot. The product needs replenishment for a set of *n* number of customers belonging to the set, $N' = \{1, 2, 3, \ldots, n\}$. N' is a set of nodes denoting customers only and is a subset of N. The set N includes nodes of all customers and the depot, i.e., $N = \{0, 1, 2, 3, \ldots, n\}$ during a time horizon of *H* days under a set $T = \{1, 2, 3, \ldots, H\}$. Each customer *i*, faces a daily demand $qm_{(i,t)}$, and a daily delivery of $ql_{(I,t)}$. It is assumed that we have a fleet of vehicles, each having a capacity *Q* units. The supplier (also called depot) is assumed to have an unlimited number of vehicles, denoted by *K*, in his/her fleet of capacity *Q*. The supplier is also assumed to have an unlimited supply of the product. The starting inventory at the beginning of the time horizon is denoted by I_s. There is a holding inventory cost per unit of the product, h_i, associated with the customer i. The edges traversed between two customers is denoted by *c(i,j)*, where *I* and *j* are indexes used with customers, and *E* is a list of all possible edges traversed.

The objective of the model is to minimize the cost involved in distance travelled by supplier, and the inventory held by retailers: Therefore, the objective of the problem is formulated as: *Minimize* $\Sigma\,\Sigma\ (c[i, j] * x[i, j] + hi[i,j] * I[i, j])$, where *x [i,j]* is the routing constant that takes the value *0* if the customer is not visited, *1* if the customer is visited along with other customers, or *2* if one customer alone is visited in the trip. Euclidean distance traversed between two customers i, and j is denoted by $c[i,j]$. The holding inventory cost per unit of product at each customer is h_i. The function is mathematically defined in categories.

Decision variables:

- Inventory held at the end of the day *t,* at the retailer i, $I_{i,t.}$
- Distance travelled each day by the vehicle which transports the product, $c(i,j)$, *where I and j are customer index*.

Constants:

- The routing constraint if a retailer is visited, $x(i)$ takes the value of *(0, 1, or 2)*.
- The cost is assumed to be *1* unit per unit distance travelled.

Objective:

- To minimize the distance travelled in logistics, the problem is formulated as a mixed integer linear programming problem

where i and j are two different retailers/customers, c is the Euclidean distance between retailers/ customers i, and j.

Capacity Constraints:

- A particular day's ending inventory is the sum of the previous day's ending inventory, in addition to that day's delivered quantity after deduction of the day's demand; *1(b)*.
- Every day's ending inventory is supposed to be greater than or equal to *0* to ensure inventory is not obsolete; *1(c)*.
- The summation of the retailer's delivery that particular day should be less than maximum inventory of the retailer after deducting yesterday's ending inventory, *1(d)*.
- Retailer's demand should not exceed maximum inventory, *1(e)*.

- Retailer's demand should not exceed vehicle capacity, *1(f)*
- Retailer's delivery of the previous day should be the demand of that day
- Variable x takes the value of either *0, 1,* or *2, 1(l)*
- The delivery, ql should exceed zero, *1(k)*

Routing Constraints:

- z takes the value of *0* if it does not traverse the edge, and *1* when it traverses the edge; *1(h)*, *1(i)*
- Routing constraint for variable x, takes the value of *0* when no edge is traversed, *1* when vehicle visits two retailers, or *2* when only one retailer is visited. *1(j)*

3.2: Formulation:

1(b). $I_{eit} = I_{t-1,i} - dm_{(i,t)} + dl_{(i,t)},$ I_{eit} is ending inventory of the customer that day, $dm_{(i,t)}$ is demand for that day t with retailer I, $dl_{(i,t)}$ is delivery made by vehicle to customer I on day t.

1(c). $I_{ei} >= 0$ where I_{eit} is ending inventory of the customer that day,

1(d). $\Sigma\ ql <= Ui - \boldsymbol{I_{ip,(t-1)}},$ ql is delivery made, Ui is max inventory at customer i, $\boldsymbol{I_{ip,(t-1)}}$ is previous day's inventory at customer i

1(e). $qm_i <= U_{i,}$ where qm is the demand for customer I, and Ui is the maximum inventory at customer i

1(f). $qm_i < = Q,$ where qm is the demand for customer I

1(g). $ql_{(i)[t-1]} == dm_{(i)\,[t-1]}$

1(k) $ql >=0,$ where ql is the delivery

1(h). $Z <= 1$

1(i). $Z = (0,1)$, where z is a routing constraint that takes value 0 if edge is traversed, otherwise 1

1(j) $x = 2*z$, where z is a routing constraint

$x = (0, 1, 2)$, where x is a routing constraint that takes value

Data

The secondary data used for tests here is a subset of instances created and tested by Adulyasak, Coelho et al 2012, Archetti et al., 2014a. The received data has a time horizon of 6 days and 3 days horizon. Distance traversed in VMI and RMI are then calculated by solving the problem.

Table 1: Parameters to solve the routing and Inventory problem

Number of customers	n
List of customers	N'
Supplier/ Depot	0
List of supplier and customers/retailers	$N, i \in \{0,1,2, \ldots, n\}$
Maximum capacity of the vehicle	Q
Fleet of Vehicles, where infinity number of vehicles is available	K; where $K = \{1, 2, 3, \ldots, k\}$
Time horizon	H; where $H = \{1,2, \ldots, t\}$
Starting inventory at Supplier, indefinite	I_{ss}
List of customers +supplier	P

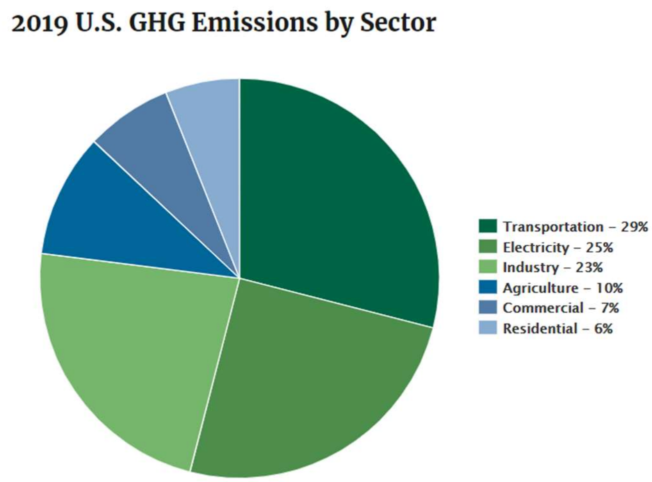
Demand for respective customers on each day	$qm_{i,t}$
That' day's Delivery at the customer	$ql_{i,t}$
Max inventory for the retailer	U_i
Holding inventory cost at the retailer	h_i
Starting inventory at retailer Ending inventory on that day at the customer	I_r $I_{e,i,t}$
Previous day's ending inventory	$I_{p,(t-1)}$
Number of times the edge is travelled	$x = 2z$
Routing binary value for edges visited	z
Index used for the Retailers	i, j
Capacity of the vehicle	Q
List of edges traversed	E
List of *x* coordinates	*locx*
List of *y* coordinates	*locy*

Carbon emission calculations for all three different fuels:

According to the inventory of US Gas Emissions and Sinks (https://www.epa.gov/ghgemissions/inventory-us-greenhouse-gas-emissions-and-sinks), transportation accounts for the largest GHG emission about 29%. Further breakdown of GHG emissions from the Transportation sector is shown by the Figure 3.2, freight trucks contribute about 24% of the total transportation GHG emission. The principal greenhouse gas produced by vehicles is Carbon-di-oxide(CO_2), but vehicles running on fossil fuels also emit methane and

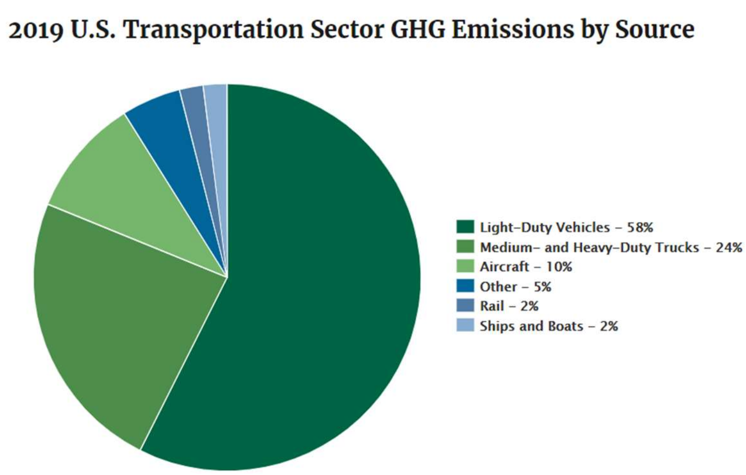
Nitrous oxide. The carbon dioxide emission is further evaluated for the respective vehicles to show that cleaner fuels cause much less emission than electric vehicles.

Figure 1: 2019 U.S. Greenhouse Gas Emissions by sector (EPA, 2019)



Source: https://www.epa.gov/greenvehicles/fast-facts-transportation-greenhouse-gas-emissions

Figure 2: 2019 U.S. Greenhouse Gas Emissions by sources (EPA, 2019)



Source: https://www.epa.gov/greenvehicles/fast-facts-transportation-greenhouse-gas-emissions

Figure 3: GHG Emissions from vehicle manufacturing

Results for energy use and GHG emissions for vehicle manufacturing in 2010 are shown in Table 3.

Table 3 – Results for Vehicle Manufacturing in 2010

Vehicle	CO_2 kg	CH_4 kg	N_2O kg	CO_2e kg	Coal GJ	Natural Gas GJ	Petroleum GJ
Passenger Modes							
CGV	7,600	13	88 (g)	7,900	38	41	21
HEV w/Ni-Mh Battery	8,000	13	95 (g)	8,300	40	41	21
Aircraft w/EIOLCA	16,000,000	72,000	970	18,000,000	57,000	100,000	66,000
Aircraft w/Ecoinvent	2,300,000	2,700	29	2,300,000	6,600	13,000	5,200
HSR	1,100,000	2,200	28	1,100,000	8,700	12,000	15,000
Freight Modes							
Truck Class 8b	54,000	5,300	1,100	500,000	160	370	110
Truck Class 6	32,000	3,200	630	300,000	95	220	66
Truck Class 5	30,000	2,900	580	270,000	87	200	61
Train	3,200,000	360,000	34,000	22,000,000	9,500	23,000	5,600
Container OGV	30,000,000	170,000	2,000	35,000,000	100,000	210,000	110,000
Tanker OGV	69,000,000	390,000	4,500	80,000,000	230,000	480,000	250,000

Fossil fuel truck carbon emission:

According to the EPA, Environmental protection agency, 161.8 grams of Carbon dioxide per ton-mile, and the heavy duty truck (class 8 trailer truck), is assumed to have a Gross vehicle weight of 80,000 lbs (36000 kgs or 40 tons) including payload capacity. CO2 emission that is emitted during commute is translated as:

Carbon dioxide emission for 40-ton vehicle = 161.8 X 40 grams per mile

= 6472 grams per mile

= 14.62 lbs per mile

According to the paper by Mikhail and Arpad, 2011, GHG emissions during production of class 8 trucks is 500,000 kgs of CO2e, carbon emission equivalent from vehicles that cause global warming; the lifespan of such a vehicle is 1000,000 miles.

Hence, the carbon emission of vehicle from production = 500,000 kgs = 1102311.31 lbs

Life Cycle Carbon Emission from Fossil fuel truck:

Tailpipe Emission from vehicle = 0.1618 kgs per ton-mile

Gross Vehicle weight = 40 tons

Total distance travelled by vehicle in a lifetime = 1000,000 miles

Lifetime exhaust or tailpipe emission = Distance travelled in a lifetime x tailpipe emission x GVW

Lifetime tailpipe emission = 1000000 x 0.1618 x 40 = **6472000 kgs** or **14268317.61 lbs** for 1000,000 miles with a gross vehicle weight of 40 tons.

Electric vehicle emission:

Electricity is generated in many ways from either renewable resources or non-renewable resources. The US Energy and Administration website shows the electricity generation in kWh and the resources from which it is generated, and the figure 4 below shows the amount of carbon emission for the same year 2020 from various resources.

A truck of GVW 40 tons is assumed to be the vehicle of choice. The Lithium battery used in a battery electric vehicle/truck is 600 kWh and lasts for eight years. The truck is assumed to travel about 1000,000 miles in its lifespan. The mileage of the vehicle is under 2kwh per mile.

Figure 4: GHG emissions from electric vehicle battery

Table 1. Studies on electric vehicle battery production emissions

Authors	Year	Battery production emissions (kg CO_2e/kWh)	Additional notes
Messagie[a]	2017	56	Assumes vehicle with 30 kWh battery constructed in the European Union, finding that BEVs will have lower life-cycle emissions than a comparable diesel vehicle when operated in any country in Europe.
Hao et al.[b]	2017	96-127	Uses China grid for battery manufacturing. Finds substantial differences between battery chemistries. Batteries produced in U.S. create 65% less GHGs.
Romare & Dahllöf[c]	2017	150-200	Reviews literature, concluding manufacturing energy contributes at least 50% of battery life-cycle emissions. Assumes battery manufacturing in Asia.
Wolfram & Wiedmann[d]	2017	106	Models life-cycle emissions of various powertrains in Australia. Manufacturing inventories come primarily from ecoinvent database.
Ambrose & Kendal[e]	2016	194-494	Uses top-down simulation to determine GHG emissions for electric vehicle manufacturing and use. Manufacturing process energy represents 80% of battery emissions. Assumes manufacturing grid representative of East Asia.
Dunn et al.[f]	2016	30-50	Uses bottom-up methodology, with U.S. electricity used for manufacturing.
Ellingsen, Singh, & Strømman[g]	2016	157	BEVs of all sizes are cleaner over a lifetime than conventional vehicles, although it may require up to 70,000 km to make up the manufacturing "debt."
Kim et al.[h]	2016	140	Study based on a Ford Focus BEV using real factory data. Total manufacturing of BEV creates 39% more GHGs than a comparable ICE car.
Peters et al.[i]	2016	110 (average)	Reveals significant variety in carbon intensities reported across literature based on methodology and chemistry.
Nealer, Reichmuth, & Anair[j]	2015	73	Finds that BEVs create 50% less GHGs on a per-mile basis than comparable ICEs, and manufacturing (in U.S.) is 8%-12% of life-cycle emissions.
Majeau-Bettez, Hawkins, & Strömman[k]	2011	200-250	Uses combined bottom-up and top-down approach. Different battery chemistries can have significantly different effects.

Source: https://theicct.org/sites/default/files/publications/EV-life-cycle-GHG_ICCT-Briefing_09022018_vF.pdf

Figure 5 : U.S. Electricity generation by Source (EPA 2020)

U.S. utility-scale electricity generation by source, amount, and share of total in 2020[1]

Preliminary data as of February 2021

Energy source	Billion kWh	Share of total
Total - all sources	4,009	
Fossil fuels (total)	2,419	60.3%
Natural Gas	1,617	40.3%
Coal	774	19.3%
Petroleum (total)	17	0.4%
Petroleum liquids	10	0.2%
Petroleum coke	8	0.2%
Other gases	11	0.3%
Nuclear	790	19.7%
Renewables (total)	792	19.8%
Wind	338	8.4%
Hydropower	291	7.3%
Solar (total)	91	2.3%
Photovoltaic	88	2.2%
Solar thermal	3	0.1%
Biomass (total)	56	1.4%
Wood	37	0.9%
Landfill gas	10	0.3%
Municipal solid waste (biogenic)	6	0.2%
Other biomass waste	2	0.1%
Geothermal	17	0.4%
Pumped storage hydropower[3]	-5	-0.1%
Other sources[4]	13	0.3%

Source: US Energy Information and Administration; https://www.eia.gov/tools/faqs/faq.php?id=427&t=3

Figure 6: Emissions from electric power sector

CO2 emissions by U.S. electric power sector by source, 2020

Source	Million metric tons	Share of sector total
Coal	786	54%
Natural gas	635	44%
Petroleum	16	1%
Other[1]	11	<1%
Total	1,448	

[1]Preliminary data for 2020. Includes CO2 emissions from the combustion of miscellaneous waste materials made from fossil fuels and by some types of geothermal power generation.

Source: US Energy Information and Administration; https://www.eia.gov/tools/faqs/faq.php?id=427&t=3

Figure 7: Electricity generation and respective carbon emissions

	Electricity generation	CO2 emissions		
	million kWh	million metric tons	million short tons	pounds per kWh
Coal	947,891	952	1,049	2.21
Natural gas	1,358,047	560	617	0.91
Petroleum	15,471	15	17	2.13

Electricity generation is net electricity generation.
Includes electricity-only power plants. Combined heat and power plants are excluded because some of their CO2 emissions are from heat-related fuel consumption.

- From the above figure, 7 we have Carbon emission to generate electricity from coal as 2.21 pounds CO2/kwh or 1.002 kgs per kwh.
- Carbon emission to generate electricity from natural gas is 0.91 pounds per kwh or 0.412 kgs per kwh
- Carbon emission to generate electricity from petroleum is 2.13 pounds per kwh or 0.966 kgs.

About 40.83% of electricity is generated from coal, 58.5% from renewable resources, and 0.67% from petroleum. Therefore, by breaking down the resource, we arrive at the electricity emissions:

- Power required by vehicle to move a mile = 2kWh
- Carbon emission in 2kWh:

Emission from 40.83% x 2kwh electricity from coal = 40.83% x 1.002 x 2

= 0.8182 kgs

Emission from 58.5% x 2 kwh electricity from natural gas = 58.5% x 0.412 x 2

= 0.482 kgs

Emission from 0.67 % x 2 kwh electricity from petrol = 0.67% x 0.966 x 2

= 0.0129 kgs

CO_2 Emission in 2kWh to move a mile = 0.8182 + 0.4808 + 0.0129 = 1.313 kgs

According to Swedish transportation administration and Environment Institute, and from the research paper by Ellingsen, Singh, and Stromann, the Carbon emission involved in lithium ion Battery production is 157 kgs of CO_2 per kwh:

- Tesla Semi Electric truck uses a 600-kWh battery to run 400,000 miles
- Emission per kwh in lithium ion battery is 157 kgs

Emission per 600 kWh battery = 600 x 157 = 94200 kgs of CO_2

Emissions for 3 batteries in a lifetime = (3 x 94200) =282600 kgs of CO_2

Life Cycle Carbon Emission from Electric truck:

CO_2 emission from Production of Vehicle = 500,000 kgs

CO_2 Emission from 3 Battery Production Emission = 600 x 3 x 157 = 282600 kgs

Carbon emission produced from electricity to run 1 mile in distance = 1.313 kgs per mile

Carbon emission from battery production per mile = 282600 / 1000000 = 0.282 kgs per mile

Carbon emission from electricity used to run 1000000 mile in distance = 1.313 x 1000000 = 1313000 kgs

Lifetime carbon emission from transportation = 282600 + 1313000 = 1595600 kgs of CO2

Hydrogen Powered truck carbon emission:

Hydrogen can be produced in a number of ways such as gasification of subbituminous coal, solar thermal splitting of water, renewable electrolysis, biological water splitting.

Figure 8: Carbon Emission for hydrogen production in kgs per kilogram of Hydrogen.

Table 5
Emissions for all three variants of hydrogen production by gasification.

Emission indicators	kg CO_2 GJ^{-1} of coal			kg CO_2 (kg H_2)$^{-1}$		
Hydrogen production technology	GET, SBC	SH, SBC	SH, L	GET, SBC	SH, SBC	SH, L
Coal production	1.94	1.94	3.24	0.416	0.383	0.698
Transport of coal	0.19	0.19	0.35	0.041	0.038	0.075
Purchase of energy, net	8.41	4.17	11.53	1.799	0.821	2.485
Purchase of electricity, net	8.41	4.17	11.53	1.799	0.821	2.485
Purchase of heat, net	0	0	0	0	0	0
CO_2 direct emissions	7.37	8.27	8.66	1.577	1.630	1.867
CO_2 captured (sequestration)	83.49	83.99	93.48	17.870	16.552	20.154
Energy consumption on sequestration	6.41	6.45	9.35	1.373	1.271	2.017
Total emissions without sequestration	101.40	98.56	117.26	21.703	19.424	25.279
Total emissions with sequestration	24.32	21.02	33.13	5.206	4.143	7.142

According to the paper by Burmistrz and Et al, 2015 "Carbon footprint of the Hydrogen Production process utilizing subbituminous coal and lignite classification", 1 kg (2.204 lbs) of hydrogen production with gasification methodology produced 19.4 kg (42.76 lbs) Carbon dioxide emission using gasification of subbituminous coal. The hydrogen produced from coal is 95%.

As per the aforementioned paper, and the Hyundai specifications (https://www.hyundai.co.nz/hyundai-motor-and-h2-energy-to-bring-the-world-s-first-fleet-of-fuel-cell-electric-trucks-into-commercial-operation-), 80000-pound truck travels can travel 400 kms (or 248.55 miles) on 32.86 kgs of Hydrogen. Therefore, 0.1322 kgs of hydrogen is needed by a hydrogen powered truck to run a mile.

CO_2 Emission per kg of hydrogen = 19.4 kgs

Hydrogen needed to run a mile = (32.86)/248.55 = 0.1322 kgs per mile

CO_2 emission from hydrogen per mile = 0.1322 x 19.4 = 2.565 kgs

Life Cycle Carbon Emission from Hydrogen powered truck:

CO_2 emission from Production of Vehicle = 5000000 kgs of CO_2e

Number of miles the truck runs in a lifetime = 1000,000

Carbon emission from hydrogen production used to run 100000 miles in distance

= 1000000 x 2.565

= 2565000 kgs of CO_2

Lifetime carbon emission for transportation = 2565000 kgs of CO_2

But carbon emission from hydrogen production by Solar splitting = 1 to 1.8 kgs per kg of H_2

Considering the emission to be 1.8 kg/ kg of H_2 = 1.8 kgs per kg of H_2

Carbon emission from hydrogen production used to run 100000 miles in distance

= 1000000 x 1.8

= 1800000 kgs of CO_2

Lifetime carbon emission for transportation = 1800000 kgs of CO_2

Table 2: Carbon Emission of Gasoline, Electric and Hydrogen Powered Vehicles

No.	Carbon Emission in Different Stages	Gasoline Truck	Electric Truck	Hydrogen Powered truck
1	Vehicle production emission per vehicle in lbs of CO2e	500,000	500,000	500,000
2	fuel +Tailpipe emission in lbs/mile for 40 ton capacity	14.264	-	-
3	Carbon emission in lbs for 2 kwh per mile electricity	-	2.89	-
4	Lithium Battery production emission in lbs per battery per mile	-	0.622	-
5	Hydrogen production per unit distance by truck	-	-	5.655
6	Total Carbon dioxide emission lbs per mile for 40 ton capacity	14.264	3.631	5.655

Carbon emission in various stages from respective trucks in kilograms

Table 3: Carbon emission from respective trucks in kgs

Stages	Fossil Fuel Vehicle	Electric Vehicle	Hydrogen vehicle Gasification technology	Hydrogen vehicles by solar splitting
CO_2 in Vehicle production in kgs	500000	500000	500000	500000
CO_2 in Battery production in kgs	0	282600	0	0
CO_2 in electricity production in kgs	0	1313000	0	0
CO_2 in Exhaust emission in kgs	6472000	0	0	0
CO_2 in H_2 production in kgs	0	0	2565000	1800000

CHAPTER 4
IMPLEMENTATION OF METHODOLOGY

CHAPTER – 4

IMPLEMENTATION OF METHODOLOGY

4.1: Explanation of the program

With respect to the customer, the sets of the demand, ending inventory, holding costs, starting retailer inventory, maximum retailer inventory, previous ending inventory, x coordinates and y coordinates are stored in qm, I_e, h, I_r, I_p, $locx$, $locy$. The data stored in all the variables are called to be printed for user verification.

An analytical library named NumPy, that works with multi-dimensional arrays, and matrices is imported. The user is asked to provide the data file and the program reads this file as input. From input file, the number of customers, maximum capacity of vehicles, period of horizon, starting supplier inventory, number of vehicles in the fleet, set of nodes, number of points to be plotted on graph is then stored in the variables, n, Q, H, I_s, K, N, P, .

Matplotlib library that facilitates plotting, and the docplex that facilitates solving linear programs is imported into the program. The model is called IRP for VMI. The annotation for plotting is specified with marker styles, font sizes, and colors. The initial plot showing coordinates and demands for the first day is plotted. The ending inventory for the first day of the horizon is the sum of the previous day's inventory and maximum inventory.

The set of edges of combinations of nodes is developed and printed out for verification, along with the Euclidean distance between the nodes. The variable u is set to the upper bound for capacity, which is also the vehicle capacity.

The loop for the first day of horizon is opened with setting the initial inventory constraints, delivery constraints, active arcs as zero before the calculated values are appended to them. For elements in N, the variable x is provided through the loop if retailer inventory Ir is less than a day's demand, then the route needs to be traversed by supplier in order to eliminate sub tours *1(i)*.

The variables in the constraints are defined for CPLEX as continuous variables and binary variables.

The minimization objective $\Sigma\,\Sigma\;(c\,[i, j] * x\,[i, j] + hi[i,j] * I[i, j])$ is called, and this is followed by calling all the associated constraints:

The first two constraints establish the paths i.e. trip between two nodes, *i* and *j*, should be traversed in a way such that *i* is different from *j*.

- 1(b) Particular day's inventory is the previous day's ending inventory in addition to that day's delivered quantity minus the day's demand.
- 1(c) the customer must never run out of stock
- 1(d) the delivered quantity should be the difference between the maximum delivery and previous day's inventory.
- 1(e) Limit of quantity to be delivered should be less than maximum inventory.
- 1(f) Delivered quantity not to exceed capacity of vehicle
- 1(g) routing constraint, *z* is either *0* or *1*.
- 1(k) Delivered quantity should be greater than zero.

CPLEX solver is called to solve and provide solution status. The arcs traversed by the vehicles are printed in the font size and styles assigned. It also prints the objective of the function. Now, the first day's inventory is assigned to the previous day's inventory as the cycle flows to the second day of the horizon, and the program continues to execute until the last day of the Horizon.

Instances used for the experiment consist of five customers and a supplier with an infinite number of products.

Optimization Algorithm Outline

Step 1. Import required libraries

Step 2. Create variables

Step 3. Declare solver

Step 4. Define objective

Step 5. Define constraints

Step 6. Invoke solver and display results

Detailed Optimization Algorithms

Retailer Managed Inventory Algorithm:

Step 1. Import required libraries

- import NumPy process numbers and generate random numbers

Step 2. Create variables

- Create variable for number of customers
- create variable Q max capacity of the vehicle used

  in Time horizon, H
- starting inventory at supplier is considered infinite by using a large number
- Range function takes values only till the last but one value, so $(n+1)$ used
- N is customer list from 1 to $\boldsymbol{n}$ because python starts a set of numbers from zero as index
- Total number of points to be plotted
- Create dictionary to create input for demands
- Prints all customers with demand in dictionary format

- Create *loc_x* empty list to append values
- *xij* is a dictionary of binary variables *0* or *1*, where *i*, and *j* belong to arcs not nodes

Step 3. Declare limits

Step 4. Define constraints

- we consider c as the cost itself, hence here *x* assumed as *1* when it travels between two different nodes.
- we consider *c* as cost and *x* equals *1*, for *i* in all points, *i!=j, j* is a customer

Step 5. Defining the objective

- objective is to min *cij*xij* where *ij* belong to arcs *A*

Step 6. Invoke solver and display results

- Call CPLEX solver to solve
- prints active arcs or the edges traversed
- prints characteristics of arcs from customer *i* to customer *j*

Retailer Managed Inventory Code:

```
import numpy as np

df= open("abs1n5H3.dat")

data =  df.readLines()

n = int(data[0].split()[0])-1

Q = int(data[0].split()[2])

H = int(data[0].split()[1])

B = int(data[1].split()[3])
```

```
N = [i for i in range(1, n+1)]

print("List of customers: "+str(N))

P = [0] + N

print("List of supplier+customers: "+str(P))

demand={}

U={}

h={}

I_start={}

j=2

for i in N:


demand[i] = int(data[j].split()[6])

    U[i] = int(data[j].split()[4])

    h[i] = float(data[j].split()[7])

    I_start[i] = int(data[j].split()[3])

    j+=1

print("retailer and demand: "+str(demand))

print("Maximum_retailer_inventory: "+str(U))

print("retailer_inventoryholding_cost: "+str(h))

print("retailer_inventory_start: "+str(I_start))

loc_x=[]

loc_y=[]

for i in range(1,len(P)+1):
```

```
    loc_x.append(float(data[i].split()[1]))


    loc_y.append(float(data[i].split()[2]))
print("x coordinates: "+str(loc_x))
print("y coordinates: "+str(loc_y))
import matplotlib.pyplot as plt
plt.scatter(loc_x[1:], loc_y[1:], c='g')
for i in N:



plt.annotate('$demand_%d=%d$' % (i, demand[i]), (loc_x[i]+2, loc_y[i])
)
plt.plot(loc_x[0], loc_y[0], c='b', marker='*')
plt.axis('scaled')
plt.show()
E = [(i, j) for i in P for j in P if i != j]
print("Edges: "+str(E))
c = {(i, j): np.hypot(loc_x[i]-loc_x[j], loc_y[i]-
loc_y[j]) for i, j in E}
print("Euc. dist. between nodes: "+str(c))
from docplex.mp.model import Model
import docplex.mp.solution as solution
mdl = Model('CVRP FOR RMI')
```

```
x = mdl.binary_var_dict(E, name='x')

print(x)

u = mdl.continuous_var_dict(N, ub=Q, name='u')

print(u)

mdl.minimize(mdl.sum(c[i, j]*x[i, j] for i, j in E))

mdl.add_constraints(mdl.sum(x[i, j] for j in P if j != i) == 1 for i i
n N)

mdl.add_constraints(mdl.sum(x[i, j] for i in P if i != j) == 1 for j i
n N)

mdl.add_indicator_constraints(mdl.indicator_constraint(x[i, j], u[i]+d
emand[j] == u[j]) for i, j in E if i != 0 and j != 0)

mdl.add_constraints(u[i] >= demand[i] for i in N)

solution = mdl.solve(log_output=True)

print(solution)solution.solve_status

active_arcs = [a for a in E if x[a].solution_value > 0.9]

plt.scatter(loc_x[1:], loc_y[1:], c='g')

for i in N:

    plt.annotate('$demand_%d=%d$' % (i, demand[i]), (loc_x[i]+2, loc_y
[i]))

for i, j in active_arcs:

    plt.plot([loc_x[i], loc_x[j]], [loc_y[i], loc_y[j]], c='b', alpha=
0.3)

plt.plot(loc_x[0], loc_y[0], c='b', marker='*')
```

```
plt.axis('scaled')

plt.show()
```

Vendor Managed Inventory Algorithm

Step 1. Import required libraries

- Import NumPy
- Open data file
- Read data file

Step 2. Create variables

- Create variable *n* for number of customers or retailers
- Create variable *Q* for capacity of vehicle
- Create a variable *H* for time horizon
- Create a variable *SSI*, starting inventory at supplier
- Create a list, *N*, number of customers
- Print *N*, list of customers
- Create a list *P*, list number of customers and depot
- Create a dictionary for demand, with respect to customer numbers in *N*
- Create a dictionary for *I*, the inventory
- Create a dictionary for *h*, inventory holding cost
- Create a dictionary for *I_start*, starting inventory
- Create empty list for *X* coordinates of *P*, *loc_x*
- Create empty list for *Y* coordinates of *P*, *loc_y*

- Import matplotlib
- Create scatter plot with x, y coordinates, demands for respective points in *P*
- Create a list of edges, *E*, where $i != j$
- Dictionary of Calculate euclidean distances for edges in *E*

Step 3. Declare solver

- Calling module docplex
- create mdl = model name
- Create a binary variable for x with values *0* or *1*, *0* if the edge not traversed, *1* if edge traversed
- Create a continuous variable u to have an upper bound, ub = *Q*, the capacity of the vehicle.

Step 4. Define Objective

- Minimize distance traversed and holding inventory cost

Step 5. Define constraints

- Inventory >= 0
- Inventory of each day, *I(t)*=inventory of previous day - the day's demand
- *q*<=*U*-previous day's ending inventory
- Demand<capacity of vehicle
- Routing constraints are considered
  - sum $y(I,j,k,t) = 2\ z$
  - $z = 0,1$
  - $q>=0$

- $y = 0$ or 1 for $i>=1$, meaning y is o when an edge between retailers is not visited
- $y=2$ if only one customer is visited, as vehicle traverses back

Step 6. Invoke solver and display results

- Call CPLEX solver
- Prints active arcs
- Display results using plot characteristics
- Keep axes length equal and scaled

Vendor Managed Inventory Code

```
import numpy as np

filename=input('Enter the data file name: ')

df= open(filename)

data = df.readlines()

n = int(data[0].split()[0])- 1

Q = int(data[0].split()[2])

H = int(data[0].split()[1])

SSI = int(data[1].split()[3]

print("SSI, starting supplier inventory: "+str(SSI))

N = [i for i in range(1, n+1)]

print("List of customers: "+str(N))

P = [0] + N

print("List of supplier+customers: "+str(P))

period=H
```

```
print("period, H/T: "+str(H))
demand={}
I_end={}
h={}
SRI={}
U_max={}
prev_end_inv={}
j=2
for i in N:
    demand[i] = int(data[j].split()[6])
    U_max[i] = int(data[j].split()[4])
    h[i] = float(data[j].split()[7])
    SRI[i] = int(data[j].split()[4])
    j+=1
I_end = prev_end_inv=U_max
print("retailer demand: "+str(demand))
print("Ending_retailer_inventory: "+str(I_end))
print("retailer_inventoryholding_cost: "+str(h))
print("starting_retailer_inventory: "+str(SRI))
print("Max_retailer_inventory: "+str(U_max))
loc_x=[]
loc_y=[]
for i in range(1,len(P)+1):
```

```
    loc_x.append(float(data[i].split()[1]))
    loc_y.append(float(data[i].split()[2]))
print("x coordinates: "+str(loc_x))
print("y coordinates: "+str(loc_y))
import matplotlib.pyplot as plt
from docplex.mp.model import Model
plt.scatter(loc_x[1:], loc_y[1:], c='g')
for i in N:
    plt.annotate('$demand_%d=%d$' % (i, demand[i]), (loc_x[i]+2, loc_y
[i]))
plt.plot(loc_x[0], loc_y[0], c='b', marker='*')
plt.axis('scaled')
plt.show()
plt.clf()
mdl = Model('IRP FOR VMI')
E = [(i, j) for i in P for j in P if i != j]
print("Edges: "+str(E))
c = {(i, j): np.hypot(loc_x[i]-loc_x[j], loc_y[i]-
loc_y[j]) for i, j in E}
print("Euc. dist. between nodes: "+str(c))
u = mdl.continuous_var_dict(N, ub=Q, name='u')
print(u)
for T in range(period):
```

```
    x = mdl.binary_var_dict(E, name='x'+str(T))

print(x)
    print("starting retailer inventory: "+str(SRI))
    print("previous end inventory: "+str(prev_end_inv))
    print("ending inventory: "+str(I_end))

cur_end_const=[0]
D_const=[0]
    U_const=[0]
    v_const = [0]
    delivery = [0]*(n+1)
    z=[0]*(n+1)
    y=0
    cnt=0
    for i in N:
        if SRI[i]<=demand[i]:
            delivery[i]=U_max[i]-SRI[i]
            z[i]=1
            cnt+=1
        else:
            delivery[i]=0
            z[i]=0
```

```
    if cnt!=0 and cnt!=1:
        y=1
    elif cnt==1:
        y=2
    Del_const=[0]
    z_const,y_const=[0],mdl.continuous_var(name='y')
    for i in N:
        cur_end_const.append(mdl.continuous_var(I_end[i]))
        v_const.append(mdl.continuous_var(SRI[i]==demand[i]))
        D_const.append(mdl.continuous_var(demand[i]))
        U_const.append(mdl.continuous_var(U_max[i]))
        Del_const.append(mdl.continuous_var(delivery[i]))
        z_const.append(mdl.continuous_var(z[i]))
        mdl.minimize(mdl.sum(c[i, j]*x[i,j] for i, j in E) + mdl.sum(h
[i]*I_end[i] for i in N))
    mdl.add_constraints(mdl.sum(x[i, j] for j in P if j != i) == 1 for
 i in N)
    mdl.add_constraints(mdl.sum(x[i, j] for i in P if i != j) == 1 for
 j in N)
    mdl.add_indicator_constraints(mdl.indicator_constraint(x[i, j], u[
i]+demand[j] == u[j]) for i, j in E if i != 0 and j != 0)
    mdl.add_constraints(u[i] <= demand[i] for i in N)
    mdl.add_constraints(Del_const[i] == U_max[i]-SRI[i] for i in N)
```

```
    mdl.add_constraints(cur_end_const[i] >= prev_end_inv[i]-
demand[i]+delivery[i] for i in N)
    mdl.add_constraints(cur_end_const[i] >=0 for i in N)
    mdl.add_constraints(v_const[i] == True for i in N)
    mdl.add_constraints(demand[i] == U_const[i]-
prev_end_inv[i] for i in N)
    mdl.add_constraints(D_const[i] >= Q for i in N)



mdl.add_constraints(D_const[i] >= 0 for i in N)
active_arcs=list()
    solution = mdl.solve(log_output=True)
    if solution!=None:
        print(solution)
        solution.solve_status
    else:
        print("solution found")
    active_arcs = [a for a in E if x[a].solution_value > 0.9]

 print(active_arcs)
    plt.scatter(loc_x[1:], loc_y[1:], c='g')
    for i in N:
```

```
        plt.annotate('$demand_%d=%d$' % (i, demand[i]), (loc_x[i]+2, l
oc_y[i]))
    for i, j in active_arcs:
        plt.plot([loc_x[i], loc_x[j]], [loc_y[i], loc_y[j]], c='b', al
pha=0.3)
    plt.plot(loc_x[0], loc_y[0], c='b', marker='*')
    plt.axis('scaled')
    plt.show()
    plt.clf()
    plt.close()
    mdl.print_information()
    prev_end_inv=I_end
    #prev_end_inv=SRI
    for i in N:
        if SRI[i]<=demand[i]:
            print("======================================")
            SRI[i]=U_max[i]
        else:
            print("---------------------------------")
            SRI[i]-=demand[i]
    I_end = SRI
    print(I_end)
```

4.2. Inputs : Description of input data:

Input Data Sample:

6	3	120					
1	309.0	334.0	462	158	.30		
2	345.0	168.0	62	93	0	31	.35
3	34.0	174.0	120	180	0	60	.14
4	297.0	367.0	34	51	0	17	.17
5	211.0	389.0	76	114	0	38	.36
6	76.0	304.0	12	24	0	12	.25

These instances are a subset from the instances used in Archetti et al., 2014a. The instances with five customers and a depot are considered for the experiments in this study.

Line one has the number of customers which is denoted by n, days in one horizon denoted by H and maximum capacity of vehicle denoted Q. The second line starts with the depot denoted as *1*; followed by x coordinate and y coordinate of the depot. The next value is the maximum product available at the depot, followed by the maximum units available every day of horizon and the inventory cost at depot.

Starting from line three, the subsequent lines have the customer number, followed by respective x and y coordinates of the location, the ending inventory on day one, maximum inventory at customer, minimum inventory, daily demand, and the inventory holding cost at the customer.

Instances:

Input Data for Abs1n5H3:

6 3 289

1	154.0	417.0		510		193	.30
2	172.0	334.0	130	195	0	65	.23
3	267.0	87.0	70	105	0	35	.32
4	148.0	433.0	58	116	0	58	.33
5	355.0	444.0	48	72	0	24	.23
6	38.0	152.0	11	22	0	11	.18

Objective (Minimize) = c[i, j] * x[i, j] for i, j in E

Figure 9: Output of Data stored in variables for verification

```
List of customers: [1, 2, 3, 4, 5]
List of supplier+customers: [0, 1, 2, 3, 4, 5]
period, H/T: 3
retailer demand: {1: 65, 2: 35, 3: 58, 4: 24, 5: 11}
Ending_retailer_inventory: {1: 195, 2: 105, 3: 116, 4: 72, 5: 22}
retailer_inventoryholding_cost: {1: 0.23, 2: 0.32, 3: 0.33, 4: 0.23, 5: 0.18}
starting_retailer_inventory: {1: 195, 2: 105, 3: 116, 4: 72, 5: 22}
Max_retailer_inventory: {1: 195, 2: 105, 3: 116, 4: 72, 5: 22}
x coordinates: [154.0, 172.0, 267.0, 148.0, 355.0, 38.0]
y coordinates: [417.0, 334.0, 87.0, 433.0, 444.0, 152.0]
```

Figure 10: Output of the edges and distances between the nodes

```
Edges: [(0, 1), (0, 2), (0, 3), (0, 4), (0, 5), (1, 0), (1, 2), (1, 3), (1, 4), (1, 5), (2, 0), (2, 1), (2, 3), (2, 4), (2, 5), (3, 0), (3, 1), (3, 2), (3, 4), (3, 5
), (4, 0), (4, 1), (4, 2), (4, 3), (4, 5), (5, 0), (5, 1), (5, 2), (5, 3), (5, 4)]
Euc. dist. between nodes: {(0, 1): 84.92938243034621, (0, 2): 348.81083698761427, (0, 3): 17.08800749063506, (0, 4): 202.80532537386685, (0, 5): 289.2766841624122, (
1, 0): 84.92938243034621, (1, 2): 264.6393772664983, (1, 3): 101.86756107809786, (1, 4): 213.5158073773462, (1, 5): 226.00884938426637, (2, 0): 348.81083698761427, (
2, 1): 264.6393772664983, (2, 3): 365.8920605861789, (2, 4): 367.6860073486616, (2, 5): 238.04621400055913, (3, 0): 17.08800749063506, (3, 1): 101.86756107809786, (3
, 2): 365.8920605861789, (3, 4): 207.2920644887305, (3, 5): 301.763152157495, (4, 0): 202.80532537386685, (4, 1): 213.5158073773462, (4, 2): 367.6860073486616, (4,
3): 207.2920644887305, (4, 5): 430.99071915761715, (5, 0): 289.2766841624122, (5, 1): 226.00884938426637, (5, 2): 238.04621400055913, (5, 3): 301.763152157495, (5,
4): 430.99071915761715}
{(0, 1): docplex.mp.Var(type=B,name='x_0_1'), (0, 2): docplex.mp.Var(type=B,name='x_0_2'), (0, 3): docplex.mp.Var(type=B,name='x_0_3'), (0, 4): docplex.mp.Var(type=B
```

Output Graphs for RMI and VMI Routing:

Figure 11: Day 1 for RMI

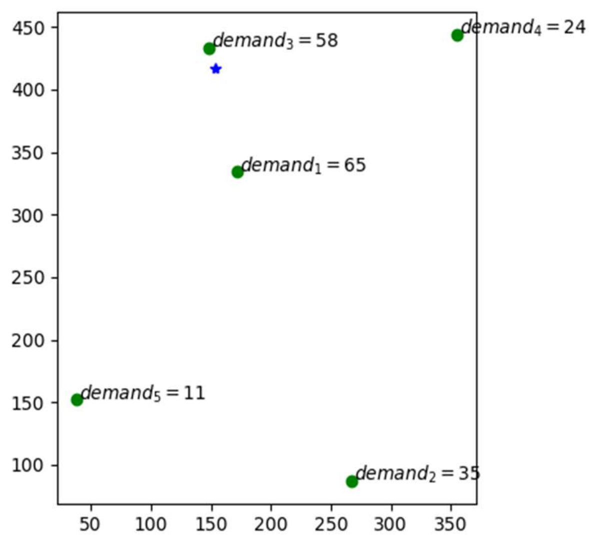


Figure 12: Day 2 for RMI

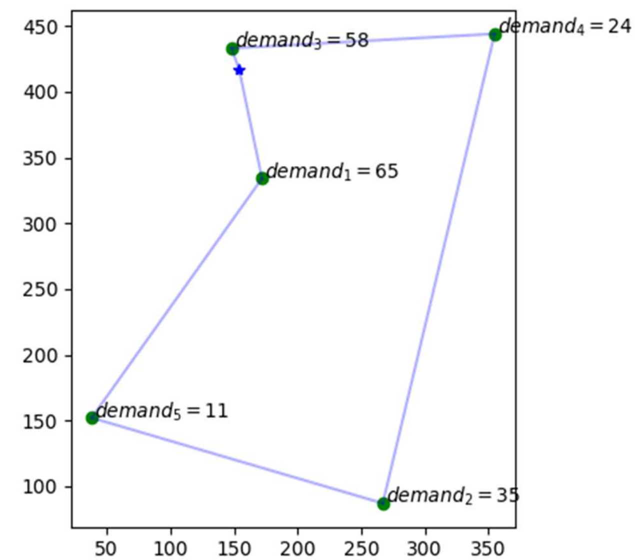


Figure 13: Day 3 for RMI

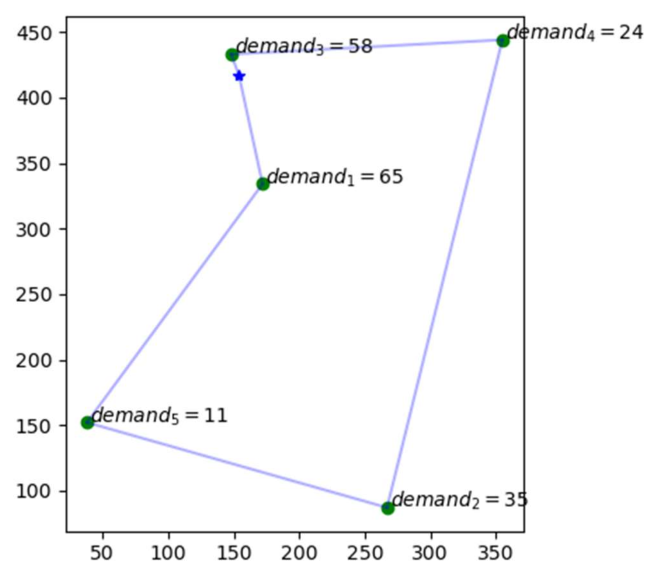


Figure 14: Output from CPLEX.

From the graphs on day 1, 2, and 3, distance sums to 2282.10 units

```
                          ------------
Total (root+branch&cut) =    0.17 sec. (0.74 ticks)
solution for: CVRP FOR RMI
objective: 1141.05
x_0_3=1
x_1_0=1
x_2_5=1
x_3_4=1
x_4_2=1
x_5_1=1
u_1=193.000
u_2=117.000
u_3=58.000
u_4=82.000
u_5=128.000
```

Figure 15: Day 1 for VMI

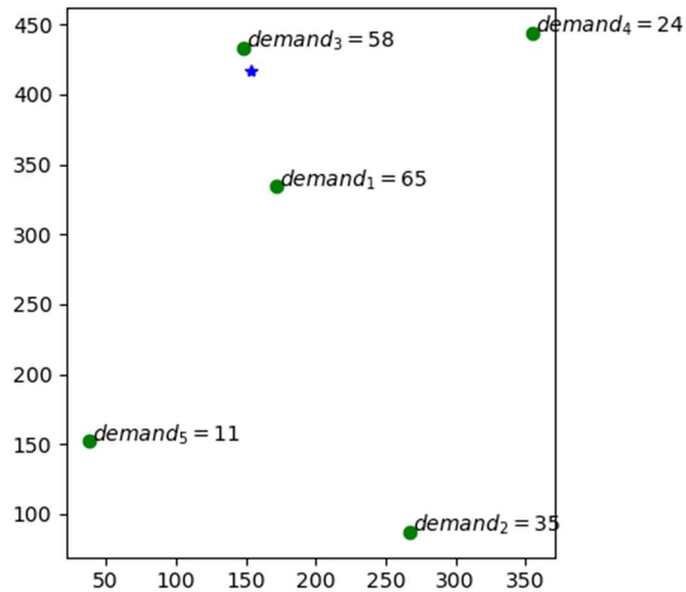


Figure 16: Day 2 for VMI

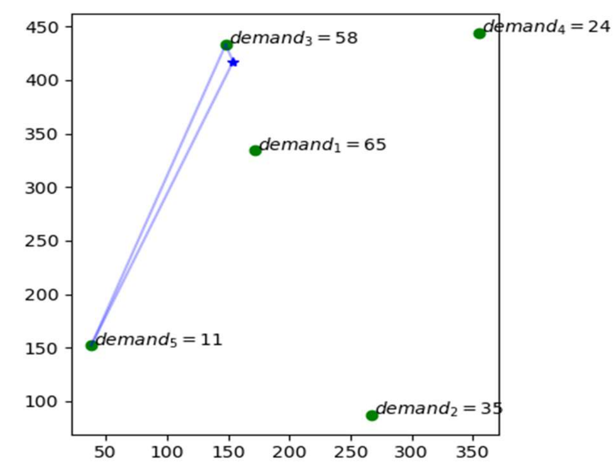
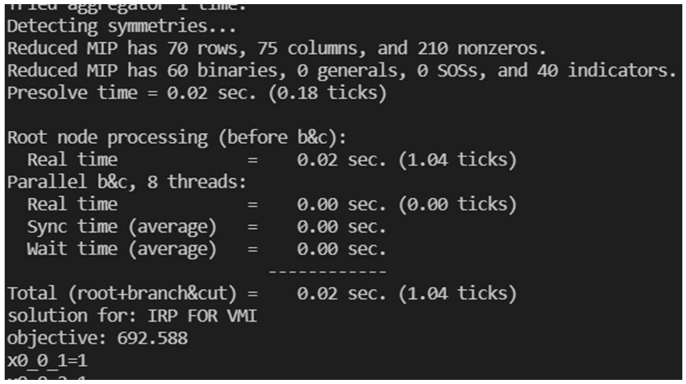


Figure 17: Output of Solution for Day 2



```
Detecting symmetries...
Reduced MIP has 70 rows, 75 columns, and 210 nonzeros.
Reduced MIP has 60 binaries, 0 generals, 0 SOSs, and 40 indicators.
Presolve time = 0.02 sec. (0.18 ticks)

Root node processing (before b&c):
  Real time             =    0.02 sec. (1.04 ticks)
Parallel b&c, 8 threads:
  Real time             =    0.00 sec. (0.00 ticks)
  Sync time (average)   =    0.00 sec.
  Wait time (average)   =    0.00 sec.
                          ------------
Total (root+branch&cut) =    0.02 sec. (1.04 ticks)
solution for: IRP FOR VMI
objective: 692.588
x0_0_1=1
```

Distance traversed by the trucks on day 2

Distance of (0 – 3 – 5 – 0) = 17.088 + 301.76 + 289.27 = 608.11 units

Figure 18: Day 3 for VMI

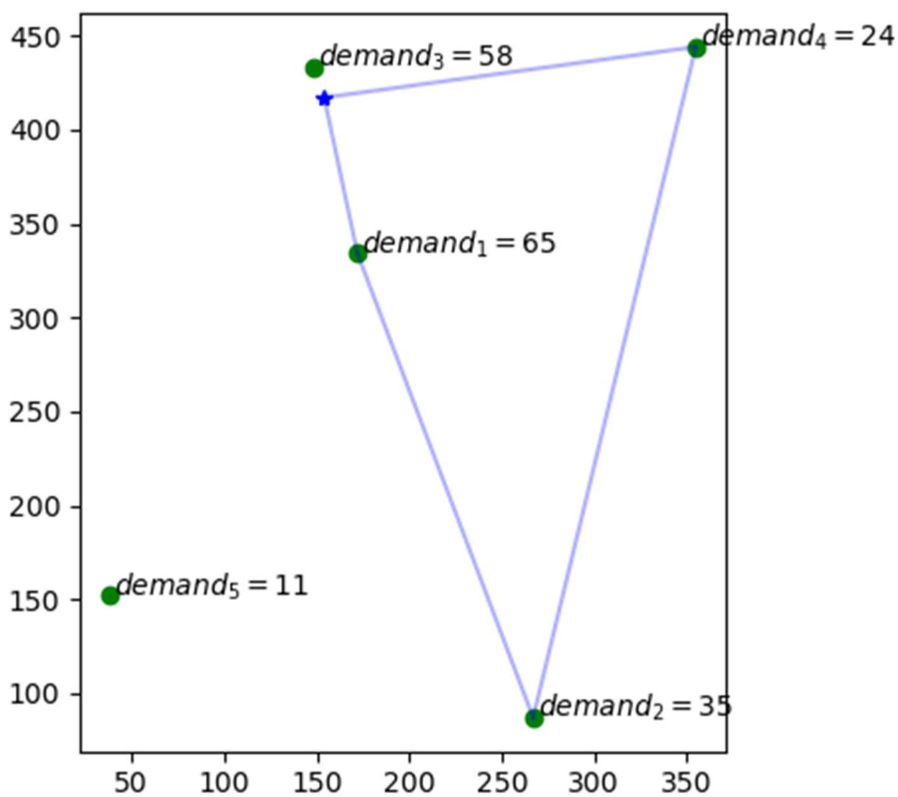


Figure 19: Output on Solution Day 3.

```
Root node processing (before b&c):
  Real time             =    0.00 sec. (0.23 ticks)
Parallel b&c, 8 threads:
  Real time             =    0.00 sec. (0.00 ticks)
  Sync time (average)   =    0.00 sec.
  Wait time (average)   =    0.00 sec.
                          ------------
Total (root+branch&cut) =    0.00 sec. (0.23 ticks)
solution found
Objective:993.9700924193728
Model: IRP FOR VMI
 - number of variables: 275
   - binary=90, integer=0, continuous=185
 - number of constraints: 315
   - linear=255, indicator=60
```

Day 3 Objective = 993.97

Distance traversed by the trucks in vendor managed inventory is

Distance of (0 – 1 – 2 – 4 – 0) = 84.92 + 264.63 + 367.68 + 202.8 = 920.03 units

Total distance traversed in the horizon = 1528.14 units

Input Data for Abs3n5H3:

```
6  3  456

  1    464.0    251.0       674       304      .30

  2     18.0      2.0   87  174   0   87      .48

  3    301.0    262.0   86  172   0   86      .37

  4    445.0    301.0   65  130   0   65      .40

  5     67.0    334.0  106  159   0   53      .49

  6    115.0    456.0   26   39   0   13      .33
```

Data:

Figure 20: Output of Data

```
PS C:\Users\krisn\PY36_CODES_MIP> & C:/Users/krisn/miniconda3/envs/PY36/python.exe c:
List of customers: [1, 2, 3, 4, 5]
List of supplier+customers: [0, 1, 2, 3, 4, 5]
retailer and demand: {1: 87, 2: 86, 3: 65, 4: 53, 5: 13}
Maximum_retailer_inventory: {1: 174, 2: 172, 3: 130, 4: 159, 5: 39}
retailer_inventoryholding_cost: {1: 0.48, 2: 0.37, 3: 0.4, 4: 0.49, 5: 0.33}
retailer_inventory_start: {1: 87, 2: 86, 3: 65, 4: 106, 5: 26}
x coordinates: [464.0, 18.0, 301.0, 445.0, 67.0, 115.0]
y coordinates: [251.0, 2.0, 262.0, 301.0, 334.0, 456.0]
```

Output Graphs for RMI and VMI Routing:

Figure 21: Day 1 for RMI

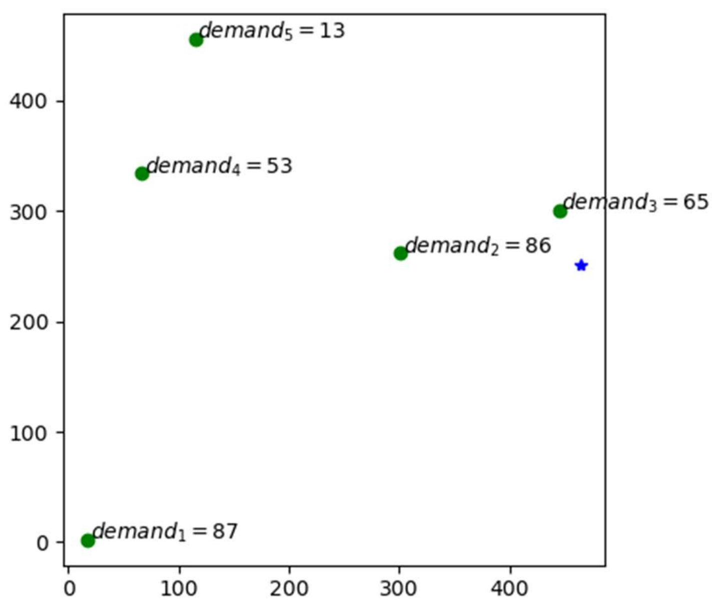


Figure 22: Day 2 for RMI

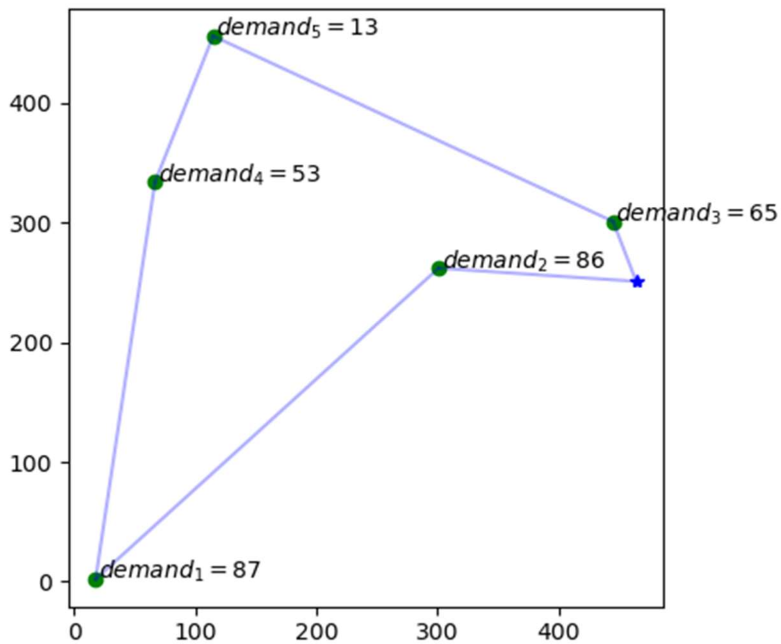
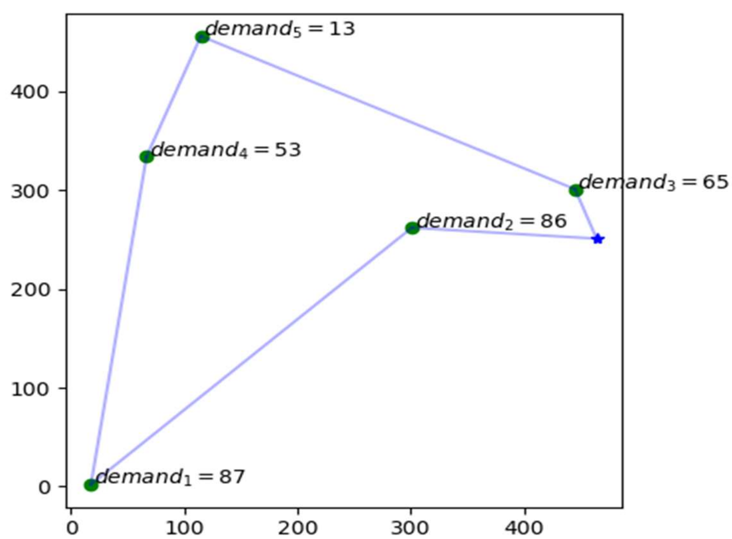


Figure 23: Day 3 for RMI



Distance traversed while in retailer managed policy is 2864.90 units

Figure 24: Day 1 for VMI

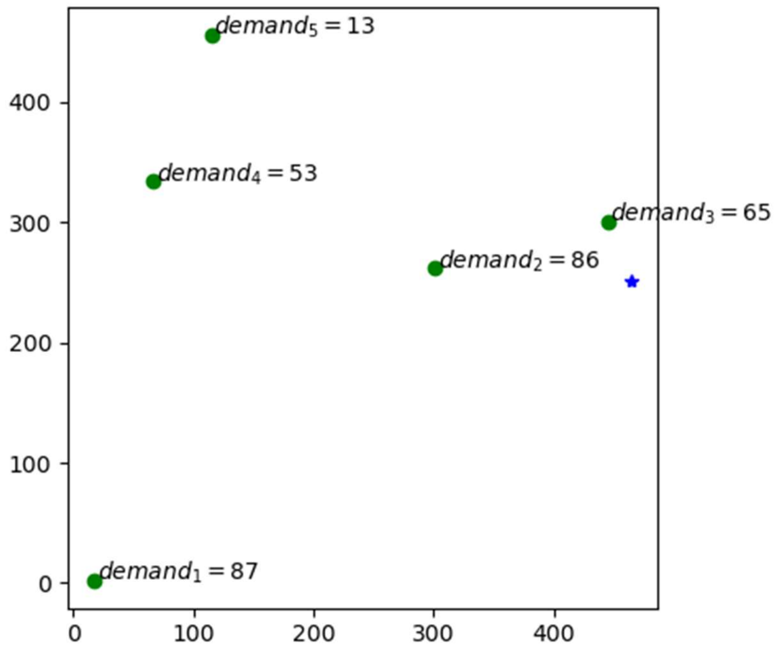
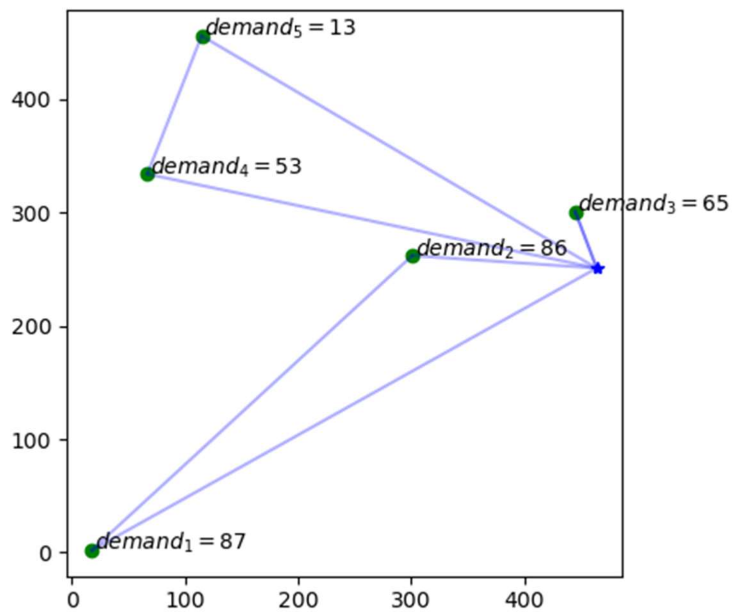


Figure 25: Day 2 for VMI



Day 3 for VMI

-

Distance traversed by the trucks in vendor managed inventory is

Distance of (0 – 2 – 1 – 0) = 163.37 + 384.3 = 510.8 units

Distance of (0 – 3 – 0) = 53.48 + 53.48 = 106.96 units

Distance of (0 – 5 – 4 – 0) = 404.75 + 131.1 = 405.58 units

Total distance traversed = 2106.86 units

Figure 26: VMI Objective

```
Total (root+branch&cut) =    0.03 sec. (1.03 ticks)
solution for: IRP FOR VMI
objective: 2396.83
u_1=87.000
u_4=53.000
x0_0_2=1
x0_0_3=1
x0_0_5=1
x0_1_0=1
x0_2_1=1
x0_3_0=1
x0_4_0=1
x0_5_4=1
```

Instances:

Abs1n5H3

6 3 289

1	154.0	417.0		510		193	.30
2	172.0	334.0	130	195	0	65	.23
3	267.0	87.0	70	105	0	35	.32
4	148.0	433.0	58	116	0	58	.33
5	355.0	444.0	48	72	0	24	.23
6	38.0	152.0	11	22	0	11	.18

Abs2n5H3

6 3 237

1	309.0	334.0		462		158	.30
2	345.0	168.0	62	93	0	31	.35
3	34.0	174.0	120	180	0	60	.14
4	297.0	367.0	34	51	0	17	.17
5	211.0	389.0	76	114	0	38	.36
6	76.0	304.0	12	24	0	12	.25

Abs3n5H3

6	3	456					
1	464.0	251.0	674		304		.30
2	18.0	2.0	87	174	0	87	.48
3	301.0	262.0	86	172	0	86	.37
4	445.0	301.0	65	130	0	65	.40
5	67.0	334.0	106	159	0	53	.49
6	115.0	456.0	26	39	0	13	.33

Abs4n5H3

6	3	268					
1	119.0	168.0	372		179		.30
2	191.0	336.0	53	106	0	53	.20
3	69.0	349.0	21	42	0	21	.19
4	94.0	235.0	24	48	0	24	.24
5	422.0	279.0	67	134	0	67	.22
6	153.0	108.0	28	42	0	14	.40

Abs5n5H3

6	3	351					
1	274.0	85.0	621		234		.30
2	364.0	171.0	38	57	0	19	.33
3	336.0	437.0	94	141	0	47	.41
4	242.0	169.0	144	216	0	72	.47
5	278.0	224.0	81	162	0	81	.36
6	192.0	260.0	30	45	0	15	.48

Abs1n5L3

6	3	289					
1	154.0	417.0	510		193		.03
2	172.0	334.0	130	195	0	65	.02
3	267.0	87.0	70	105	0	35	.03
4	148.0	433.0	58	116	0	58	.03
5	355.0	444.0	48	72	0	24	.02
6	38.0	152.0	11	22	0	11	.02

Abs2n5L3

6 3 237

1	309.0	334.0		462		158	.03
2	345.0	168.0	62	93	0	31	.04
3	34.0	174.0	120	180	0	60	.01
4	297.0	367.0	34	51	0	17	.02
5	211.0	389.0	76	114	0	38	.04
6	76.0	304.0	12	24	0	12	.03

Abs3n5L3

6 3 456

1	464.0	251.0		674		304	.03
2	18.0	2.0	87	174	0	87	.05
3	301.0	262.0	86	172	0	86	.04
4	445.0	301.0	65	130	0	65	.04
5	67.0	334.0	106	159	0	53	.05
6	115.0	456.0	26	39	0	13	.03

Abs4n5L3

6 3 268

1	119.0	168.0		372		179	.03
2	191.0	336.0	53	106	0	53	.02
3	69.0	349.0	21	42	0	21	.02
4	94.0	235.0	24	48	0	24	.02
5	422.0	279.0	67	134	0	67	.02
6	153.0	108.0	28	42	0	14	.04

Abs5n5L3

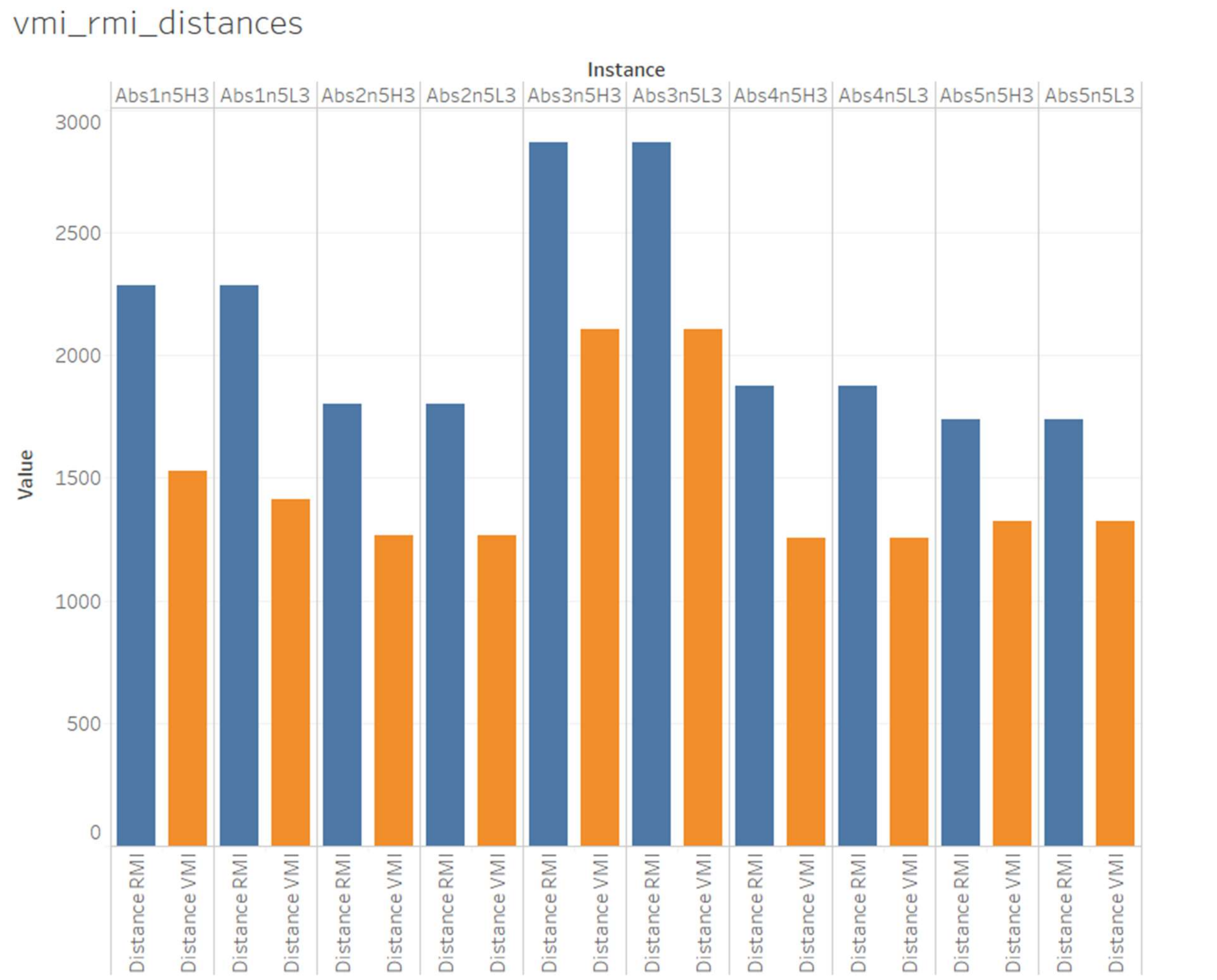
6 3 351

1	274.0	85.0		621		234	.03
2	364.0	171.0	38	57	0	19	.03
3	336.0	437.0	94	141	0	47	.04
4	242.0	169.0	144	216	0	72	.05
5	278.0	224.0	81	162	0	81	.04
6	192.0	260.0	30	45	0	15	.05

Table 4.1: Instances with their Objectives and respective distances

Sl. No.	Instance	VMI Objective	RMI Objective	Distance Traversed in VMI	Distance Traversed in RMI
1	Abs1n5H3	1686.558	2282.10	1528.14	2282.1
2	Abs2n5H3	1379.14	1801.89	1265.62	1801.89
3	Abs3n5H3	2396.83	2864.9	2106.86	2864.9
4	Abs4n5H3	1344.31	1873.11	1257.29	1873.11
5	Abs5n5H3	1580.56	1736.31	1322.45	1736.31
6	Abs1n5L3	1423.97	2282.1	1411.37	2282.1
7	Abs2n5L3	1277.5	1801.88	1265.62	1801.88
8	Abs3n5L3	2136.79	2864.9	2106.86	2864.9
9	Abs4n5L3	1265.61	1873.99	1257.28	1873.99
10	Abs5n5L3	1349.38	1736.29	1322.43	1736.29

Figure 27: RMI and VMI Distances comparison



4.3 Validation : An experiment with known solution

Hand calculations along with travelling salesman problem technique was implemented on the instance abs1n5H3.dat to validate the program. The code uses a branch and cut method, to solve the problem. The problem was solved with code, and showed an objective of 692.58 on day two, and 993.97 on day three. The solve time was 0.01 seconds. The output shows the number of linear, binary variables, number of cutting planes, or constraints and an arc that is plotted based on the tour to supply products. The objectives of hand calculated variation turned out to be 650.35 on day two, 1015.04 on day three.

For validation, the input data of all retailers is tabulated as shown below. The inventory level at each retailer is calculated on a daily basis as shown in the inventory table by deducting the previous day's demand. This step is done in order to figure out the day when respective retailers' replenishment is necessary.

Figure 28: Data and Hand Calculations for instance abs1n5H3.dat

Hand Calculations for the instance abs1n5H3.dat

(Min) Z = ΣΣ [C(i,j) * y(i,j) + h(i)*I_end)]

(Min) Z = Distance travelled + inventory cost

Input Data:

Customers + Depot	0	1	2	3	4	5
Starting retailer inventory in units	0	195	105	116	72	22
Retailer demand in units	0	65	35	58	24	11
Inventory cost unit currency	0	0.23	0.32	0.33	0.23	0.18

Inventory Table

Day 1 retailer inventory	0	130	70	58	48	11
Day 2 retailer inventory	0	65	35	stock up	24	stock up
Day 3 retailer inventory	0	stock up	stock up	116	stock up	22

Figure 29: Edge Distances between nodes

Distance Table

Edges between the retailers (0 - 5)	**0**	**1**	**2**	**3**	**4**	**5**
0	0	84.92	348.81	17.08	202.8	289.27
1	84.92	0	264.63	101.86	213.51	226
2	348.81	264.63	0	365.89	367.68	238.04
3	17.08	101.86	365.89	0	207.29	301.76
4	202.8	213.51	367.68	207.29	0	430.99
5	289.27	226	238.04	301.76	430.99	0

Once the inventory downsizes to one day's demand, the supplier supplies the products along with optimizing the route. Here, in this instance, retailers 3, and 5 need replenishment on the second day while all other retailers need to be replenished only on the third day. Optimization of route is done with Traveling Salesperson Problem technique on Microsoft Excel with evolutionary solver.

The Euclidean distances between depot and retailers, and between different retailers is tabulated for convenience.

On day one of the three-day Horizon, the retailers are stocked up to maximum capacity. Hence, there is no necessity of traveling.

On day two, retailers 3, and 5 need to be replenished from depot 0. Hence the table for three vertices is created in order to locate the route taken by vehicle.

Figure 30: Distance travelled on Day 2

Vertices	**0**	**3**	**5**	**0**
Index	1	2	3	1
Distance	17.08	301.76	289.27	
Total distance travelled [c(i,j)] =	**608.11**			
Inventory holding cost =	inventory held * cost at 3, and 5		**= 42.24 unit currency**	
Objective, Day 2 =	distance travelled + inventory held =		**650.35**	**unit currency**

Total distance travelled on day two sums up to 608.11.

Distance between 0 – 3 = 17.08 units

Distance between 3 – 5 = 301.76 units

Distance between 5 – 0 = 289.27 units

Total distance traversed = 17.08 + 301.76 + 289.27 = 608.11 units.

The objective sums up to the cost of distance travelled (with one being the price per unit distance) , and cost of inventory held at 3, and 5.

The objective, for day two is 608.11 + 42.24 = 650.35 units

Figure 31: Objective on Day 2

Day 2 Code Results:	
Route	**0 - 3 - 5-0**
Objective in unit currency	**692.588**

Figure 32: Day 3 Tour calculation

Day 3 Tour					
Vertices		**0**	**1**	**2**	**4**
	Index	1	2	3	4
From	**0**	0	84.92	348.81	202.8
	1	84.92	0	264.63	213.51
	2	348.81	264.63	0	367.68
	4	202.8	213.51	367.68	0

	0	**1**	**2**	**4**	
Index	1	2	3	4	1
Distance	84.92	264.63	367.68	202.8	
Total distance travelled [c(i,j)] =	**920.03**				
Inventory holding cost =	inventory held * cost at 1, 2, and 4		**= 95.01 unit currency**		

On day three, retailers 1, 2, and 4 are to be visited. Hence a table of Euclidean distances is prepared once again for the respective vertices. The evolutionary solver is set to bring the shortest possible path from 0 to 1, 1 to 4, and from 4 to 0, the depot.

Distance travelled on day 3:

Distance between 0 – 1 = 84.92 units.

Distance between 1 – 2 = 264.63 units.

Distance between 2 – 4 = 367.68 units.

Distance between 4 – 0 = 202.8 units.

Sum of distances = 920.03 units.

The inventory holding cost for 1, 2, and 4 = 95.01 unit currency.

The objective on day three = 920.03 + 95.01 unit cost

The summation of objectives = objective on day 3 + objective on day 4

= 1015.04 + 650.35

=1665.39.

The horizon objective result from the code is 1686.558 cost units.

Figure 33: Objective for Day 3

Route	0 - 1 - 2 - 4 - 0	
Objective, Day 3 =	920.03 + 95.01 =	**1015.04**
Total distance travelled in 3 day Horizor	**1528.14**	

Figure 34: Day 3 Code objective

Day 3 Code Results:	
Route	**0 - 1 - 2 - 4**
Objective in unit currency	**993.97**
Horizon Objective	**1686.558**

Table 4: Output Results: Explain the output results of the model:

Instances	Distance traversed in VMI	Carbon emission factor with Truck run on Gas (14.264)	Carbon emission for Electric Truck (3.631)	Carbon emission for Hydrogen powered Truck (5.655)	Carbon emission from Solar split hydrogen (3.086 lbs per mile)
Abs1n5H3	1528.14	21797.39	5548.676	8641.6317	4715.840
Abs2n5H3	1265.62	18653.97	4465.107	7157.0811	3905.703
Abs3n5H3	2106.86	31053.01	7433.002	11914.293	6501.769
Abs4n5H3	1257.29	18531.2	4435.719	7109.975	3879.996
Abs5n5H3	1322.45	19491.59	4665.604	7478.4548	4081.08
Abs1n5L3	1411.37	20802.18	4979.313	7981.2974	4355.487
Abs2n5L3	1265.62	18653.97	4465.107	7157.0811	3905.703
Abs3n5L3	2106.86	31053.01	7433.002	11914.293	6501.769
Abs4n5L3	1257.28	18531.05	4435.684	7109.9184	3879.966
Abs5n5L3	1322.43	19491.3	4665.533	7478.3417	4081.018

For the file abs1n5H3.dat, the objective after the first day is displayed along with the number of variables, in this case it is 185 variables out of which 60 are binary, and 125 are continuous. This problem has a total of 210 constraints out of which 170 are linear constraints, and 40 are indicator constraints.

CHAPTER 5
Conclusion and Future Research

CHAPTER – 5

CONCLUSION AND FUTURE RESEARCH

Here, in this study the integrated inventory problem was analyzed against traditional inventory problems using mixed integer linear programming with the help of optimization solver. Formulation of objectives and constraints on problems was made to minimize cost of the inventory policies, the objectives were thus optimized respectively. The optimization solver uses branch and bound concept. The data for 10 instances were considered in the study. Data instance included number of customers to be served, duration of time horizon, capacity of the vehicle in the first line. The next line is depot number followed by its x and y coordinates, followed by demand for the horizon, every day demand in the horizon, the production price. The following lines start with customer or retailer numbers, followed by the coordinates, two-day inventor, maximum inventory, minimum inventory and holding inventory cost. The constraints applied to the linear program helped bring an optimized solution.

When the instance is fed and run, the routes are plotted by the optimization software considering the shortest route for the demand needed in RMI, on every day basis. In VMI, objective is solved subjected to the routing constraints, stockout constraints, demand constraints and hence vehicle goes to stock up only once or twice the horizon. This study focuses on illustrating the routes for both policies and carbon emission involved in both policies for inventory problems using optimization.

Here in the problem, inventory held at the end of the day, the distance travelled by the vehicle are the decision variables that are under our control to minimize the objective of model. The routing constraint, x which takes the value of 0, 1 or 2 is a constant as well the cost assumed to travel a unit distance is considered one-unit currency. The objective to minimize is the cost or when decision variables are individually considered, the distance and inventory are to be minimized. The capacity constraints here in Inventory routing problem is Day's ending inventory never equals to zero. It is never obsolete. The inventory is always stocked to maximum level. The inventory does not cross maximum inventory.

To begin writing program import required analytical libraries associated with python. Create the variables for the problems, declare solver, call the objective, define all the constraints, run the solver to display distances, and routes.

For further reduction of carbon emission, alternative fuels with cleaner sourcing and production methods is considered a reasonable and sustainable option. Such options need not require hydrocarbons from underneath the earth. The data gathered from various papers on the emissions that assessed carbon emission from different vehicles, is illustrated in chapter 4 with tables and graphs.

Consequently, the transition to vehicles running on battery, and vehicles running on cleaner green hydrogen such as the one produced by solar splitting reduces carbon footprint. This saves resources and reduces the detrimental effects on the environment. The usage of fossil fuels makes global warming a perpetual cycle, hence alternative fuels could be an answer to transportation.

From the data gathered the ideal option of a vehicle that is sustainable is the electric truck running on lithium ion batteries. The illustration of carbon emission shown in the paper is gathered from various assessments made in the recent decade. The missions from different stages such as CO2 equivalent emission in vehicle production, fuel production and exhaust pipe emission are brought in together. Certain data are assumed. Such a vehicle proves to be promising.

Limitations and Assumptions made:

- This code cannot be connected with live data or traffic.
- Free CPLEX edition allows only a thousand variables and a thousand constraints
- Takes time to run bigger real models with many more constraints.
- An availability of unlimited supply of goods at depot seems unrealistic.
- The distance calculated between depot and customers or amongst customers is assumed to be Euclidean or a straight line, hypotenuse.
- Euclidean distances cannot be applied for real scenario to get exact results.

This research could not only help with inventory, decision making, freight planning and transportation but also to get an insight on how damaging the conventional freight vehicles are. Such insights can bring people together to bring about a change to help with carbon neutrality Using the vehicles for its lifetime of 1000,000 reduces rate of emissions.

From the aforementioned chapters, the emission from a regular freight truck of 36 tons is told to emit about 6472 tons in a lifetime or 14.62 lbs per mile, where as the novel electric freight trucks of a similar capacity emit only about 1595 tons in a lifetime or 3.631 lbs per mile; most of the emission for electric vehicle is from lithium ion battery and electricity generated from non-renewable resources. Any further reductions would need deeper research on fuels.

Future research extensions:

- Implementing computer science heuristics to solve the problem will quicken the process. Solving the inventory routing problem for more than 10 customers on the free version was a challenge. Any number higher than this with heuristics would be much quicker. With explanations in relation with classical routing problems. This method could be extended to problems with multi vehicle cases.

- Evaluation of emissions could be perfected by including various other elements such as logistics of vehicle manufacturing, transportation of electricity, infrastructure to store fuels. Repurposing of lithium batteries. The problem could be extended to a wider network of customers. Possibly alternate fuels from other materials would help us achieve close to perfect answers.
- The emission analysis here is preliminary. With large scale adoption of electric vehicles into implementation, there is improved technology and powerful batteries. A deeper future research on the components that go in to making could be considered. A study much in detail with respect to components that go into vehicle manufacturing could perfect the research on life-cycle process changes.
- Green house gas emissions include carbon monoxide, Sulphur dioxide, nitrogen dioxie which is not included in the study. With respect to distance fuels used, the necessary study could be branched to solve the problem.

REFERENCES

REFERENCES

1. Cplex, I. I. (2009). V12. 1: User's Manual for CPLEX. *International Business Machines Corporation*, *46*(53), 157.
2. Archetti, C., & Speranza, M. G. (2016). The inventory routing problem: The value of integration. *International Transactions in Operational Research*, *23*(3), 393–407. https://doi.org/10.1111/itor.12226
3. Bertazzi, L., Speranza, M.G. Inventory routing problems with multiple customers. *EURO J Transp Logist* **2,** 255–275 (2013). https://doi.org/10.1007/s13676-013-0027-z
4. Formulations and Branch-and-Cut Algorithms for Multivehicle Production and Inventory Routing Problems. (2014). *INFORMS J. on Computing*, *26*(1), 103–120. https://doi.org/10.1287/ijoc.2013.0550
5. Environmental Protection Agency. (2021) *Inventory of U.S. Greenhouse Gas Emissions and Sinks* (EPA 430-R-21-001). Environmental Protection agency. https://www.epa.gov/sites/production/files/2021-02/documents/us-ghg-inventory-2021-main-text.pdf
6. *US Energy Information Administration. (2021)* How much of U.S. carbon dioxide emissions are associated with electricity generation? *https://www.eia.gov/tools/faqs/faq.php?id=77&t=3*
7. Blackstone, J. H., Cox, J. F., & APICS--The Educational Society for Resource Management. (2005). *APICS dictionary*. Alexandria, VA: APICS.
8. Lee, D.-Y., Elgowainy, A., Kotz, A., Vijayagopal, R., & Marcinkoski, J. (2018). Life-cycle implications of hydrogen fuel cell electric vehicle technology for medium- and heavy-duty trucks. *Journal of Power Sources*, *393*, 217–229. https://doi.org/10.1016/j.jpowsour.2018.05.012

9. *How the World Passed a Carbon Threshold and Why It Matters*. (n.d.). Yale E360. Retrieved April 11, 2021, from https://e360.yale.edu/features/how-the-world-passed-a-carbon-threshold-400ppm-and-why-it-matters
10. Change, N. G. C. (n.d.). *Carbon Dioxide Concentration | NASA Global Climate Change*. Climate Change: Vital Signs of the Planet. https://climate.nasa.gov/vital-signs/carbon-dioxide
11. Skelley, A., & Images, G. (2016). Sustainability in supply chains. *Supply Chain*, 8.
12. *Hydrogen and fuel cells. (2016). De Gruyter Open.*
13. Wang, S., & Jiang, S. P. (2017). Prospects of fuel cell technologies. *National Science Review*, *4*(2), 163–166. https://doi.org/10.1093/nsr/nww099
14. Lee, D.-Y., Elgowainy, A., Kotz, A., Vijayagopal, R., & Marcinkoski, J. (2018). Life-cycle implications of hydrogen fuel cell electric vehicle technology for medium- and heavy-duty trucks. *Journal of Power Sources*, *393*, 217–229. https://doi.org/10.1016/j.jpowsour.2018.05.012
15. Arshinder, Kanda, A., & Deshmukh, S. G. (2008). Supply chain coordination: Perspectives, empirical studies and research directions. *International Journal of Production Economics*, *115*(2), 316–335. https://doi.org/10.1016/j.ijpe.2008.05.011
16. Adulyasak, Y., Cordeau, J.-F., Jans, R., 2013. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing* 26, 1, 103–120.
17. Andersson H, Hoff A, Christiansen M, Hasle G, Løkketangen A (2010) Industrial aspects and literature survey: Combined inventory management and routing. Comput. Oper. Res. 37:1515–1536.

18. Archetti, C., Bertazzi, L., Laporte, G., & Speranza, M. G. (2007). A Branch-and-Cut Algorithm for a Vendor-Managed Inventory-Routing Problem. *Transportation Science*, *41*(3), 382–391. https://doi.org/10.1287/trsc.1060.0188

19. Sari, K., 2008. On the benefits of CPFR and VMI: a comparative simulation study. *International Journal of Production Economics* 113, 575–586.

20. Yu, Z., Yan, H., Edwin Chen, T.C., 2001. Benefits of information sharing with supply chain partnerships. *Industrial Management and Data Systems* 101, 114–121.

21. Tyan, J.,Wee, H.-M., 2003. Vendor managed inventory: a survey of the Taiwanese grocery industry. *Journal of Purchasing and Supply Management* 9, 11–18.

22. Kuk, G., 2004. Effectiveness of vendor-managed inventory in the electronics industry: determinants and outcomes. *Information and Management* 41, 645–654.

23. Toth, P., Vigo, D., 2014. *Vehicle Routing: Problems, Methods, and Applications (2nd edn), MOS-SIAM Series on Optimization*, Vol. 18. SIAM, Philadelphia, PA.

24. Bertazzi, L., Speranza, M.G., 2012. Inventory routing problems: an introduction. *EURO Journal on Transportation and Logistics* 1, 307–326.

25. Bertazzi, L., Speranza,M.G., 2013. Inventory routing problem with multiple customers.*EUROJournal on Transportation and Logistics* 2, 3, 255–275.

26. Bertazzi, L., Speranza, M.G., Savelsbergh, M.W.P., 2008. Inventory routing. In Golden, B., Raghavan, R., and Wasil, E. (eds) *The Vehicle Routing Problem Latest Advances and New Challenges, Operations Research/Computer Science Interfaces Series*, Vol. 43. Springer-Verlag, New York, pp. 49–72.

27. Coelho, L.C., Cordeau, J.-F., Laporte, G., 2013. Thirty years of inventory routing. *Transportation Science* 48, 1–19.

28. Adulyasak, Y., Cordeau, J.-F., Jans, R., 2013. Formulations and branch-and-cut algorithms for multivehicle production and inventory routing problems. *INFORMS Journal on Computing* 26, 1, 103–120.

29. Adulyasak, Y., Cordeau, J.-F., Jans, R., 2012. Optimization-based adaptive large neighborhood search for the production routing problem. *Transportation Science* 48, 1, 20–45.

30. Coelho, L.C., Cordeau, J.-F., Laporte, G., 2012. Consistency in multi-vehicle inventory-routing. *Transportation Research Part C* 24, 270–287.

31. Coelho, L.C., Laporte, G., 2013. The exact solution of several classes of inventory-routing problems. *Computers and Operations Research* 40, 2, 558–565.

32. Archetti, C., Bianchessi, N., Irnich, S., Speranza, M.G., 2014a. Formulations for an inventory routing problem. *International Transactions in Operational Research* 21, 353–374.

33. Archetti,C., Boland,N., Speranza,M.G., 2014b.A Matheuristic for the multi-vehicle inventory routing problem.Technical Report Working paper WPDEM 2014/3, Dipartimento di Economia e Management, Universit`a degli Studi di Brescia, Italy.

34. *carbon emission—U.S. Energy Information Administration Search Results*. (n.d.).

35. Caceres, H., (2020). (*Industrial-Ucn/Jupyter-Examples*, 2020/2021) https://github.com/industrial-ucn/jupyter-examples/commit/962a23a16b2933e28cf19c8cd55ccf5c9d987ebd

36. Chester, M., & Horvath, A. (2011). Vehicle Manufacturing Futures in Transportation Life-cycle Assessment. *UC Berkeley: Institute of Transportation Studies at UC Berkeley*. Retrieved from https://escholarship.org/uc/item/1qp3f0vc

37. https://www.tesla.com/semi

38. https://fuelcelltrucks.eu/project/nikola-two/

39. Hall & Lutsey, n.d.(2018). Effects of battery manufacturing on electric vehicle life-cycle greenhouse gas emissions. *The international council on clean transportation.*

40. (*Frequently Asked Questions (FAQs) - U.S. Energy Information Administration (EIA)*, n.d.)

41. https://www.eia.gov/tools/faqs/faq.php?id=427&t=3