The Pennsylvania State University

The Graduate School

College of Engineering

AN AGENT BASED AND ANT COLONY METAHEURISTIC APPROACH TO THE LAST MILE LOGISTICS PROBLEM

A Thesis in

Industrial Engineering

by

Chaitanya Kaul

Submitted in Partial Fulfillment

of the Requirements for

the Degree of

Master of Science

August 2018

The thesis of Chaitanya Kaul was reviewed and approved * by the following:

Terry Harrison

Professor of Supply Chain and Information Systems

Daniel Finke

Assistant Research Professor

Thesis Adviser

Applied Research Laboratory, Pennsylvania State University

Robert Voigt

Professor of Industrial and Manufacturing Engineering

Graduate Program Coordinator, Department of Industrial and Manufacturing Engineering

*Signatures are on file in the Graduate School.

ABSTRACT

The last mile logistics refers to the last leg of delivery in a supply-chain transportation network. More transporters and delivery runs are required to fulfill this demand and hence make the last-mile logistic a highly inefficient and time-consuming segment of a supply chain. In this thesis, the formulation of the last mile logistics problem has been presented as an optimization problem and is solved by a modified meta-heuristic algorithm and an agent-based simulation technique. For the last-mile logistics problem, the widely studied vehicle routing problem with time windows transportation model was considered, as it abstracts the salient features of numerous logistics and transportation related real-world problems. The ant colony metaheuristic was then modified to find global minimum in case of the last mile problem. In this work, the last mile logistics problem was then also solved as an agent-based simulation model. Due to its efficient behavioral and communicative patterns, the agent-based systems provide a powerful alternative to traditional optimization techniques. In this work, we present a formulation of the last mile logistics problem as an ant colony metaheuristic and as a multi-agent optimization model. The modified ant colony algorithm was experimented on test problems and data, and successful results were obtained. A detailed experimental assessment of the agent based simulation model and our modified ant colony metaheuristic is presented, including the comparison to the traditional centralized algorithms. A detailed analysis of the solution approach is performed as well, generating future research opportunities. The outcomes of this thesis demonstrate that the methodologies return the best-known solutions accomplished by the state-of-the-art algorithms in most of our experimental cases, representing a significant improvement over the previous comparable agent based or metaheuristic studies. The core commitment of this thesis is the more profound and novel understanding of the implications of embracing agent-based and a heuristic approach to solve the last mile network problems and complex transportation-optimization problems.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

At the outset, I would like to thank Prof. Dan Finke who serves as the inspiration and advisor for my thesis. This thesis work would also never come to being without the support of my supervisors, co-advisors and readers. I would also like to express my profuse gratitude towards all my friends and family who supported and motivated me throughout this effort.

Chapter 1

Introduction

As we head into the new decade, the gravitational draw of globalization, the growth of e-commerce and emerging markets - is forcing enterprises of all sizes to build alliances and online logistics systems that efficiently deliver products to customers while providing a worldwide view of operations. Conventional supply-chain companies are growing new methodologies to track requests and respond to changes in the handling and transportation of goods as they travel through the supply chain network from manufacturer to the customer. One crucial component of such a supply-chain framework is the last mile logistics. The last-mile delivery in present day logistics connotes the concluding leg of a transportation network of commodities or services, originating from a hub (ex. fulfilment centers, depots etc.) to the last customer. The focal point of last mile logistics is to deliver things to the end-customer as quick as possible and with the least cost. More often, last mile logistics involves the use of small transporters to deliver items to customers. Even though the name denotes that it is the last mile of delivery, the last mile logistics can sometimes range from a few blocks to even 50 or 100 miles. Last-mile delivery continues to be a logistics riddle for the supply chain organizations, on how to manage a complex distribution network at the lowest costs.

The expression "last mile" was initially used as a part of the telecommunications field network, and has recently been connected to supply chain management. Originally, the effective distribution technique to deliver goods to the point of consumption were not given as much notice. Professor Bernard J. LaLonde (1997) in his research demonstrated the positive effects of last mile delivery technique on consumer satisfaction and benefit. Today buyers have access to a wide assortment of items, across households and global land limits, delivering products specifically to their homes in a matter of hours.

In the last mile logistics, the most popular transport mode adopted is road network, although other methods are also used, according to the characteristics of cities. In case of urban environment, it is generally fragmented; transporters engage different carriers for the delivery to the vendors in the cities. This produces a low load factor of vehicles, a great number of routes and network costs. Several studies have been conducted with the objective to improve the efficiency in delivery logistics, as well as to reduce the related routing and fuel costs. The major challenge for the last mile logistics system is to reduce the above-mentioned concerns and achieve a higher service level.

The aim of the thesis is to answer the question whether there are worthy solutions to such complex and large logistics optimization problems. In this work, two new optimization techniques have been proposed to the last mile logistics problem. The first optimization approach used in this work is the development of a new and modified ant colony metaheuristic capable of searching for a global minima. The second optimization technique proposed in the work is the development of an agent based simulation model using a simulation tool. Both of the optimization techniques were then subjected to comparative performance evaluation. The basic model was developed as a routing problem using two different sets of data for performance evaluation. The novelty of this work is the use of agent based simulation model and a metaheuristic for the lesser-tackled last mile logistics problem. As in the coming future, last mile logistics will be a major business opportunity; a widespread review of the related literature and contributions on the last mile logistics systems is presented in the upcoming section.

1.1 Thesis outline

The aim of our work involves developing two novel optimization models, which aim at minimizing the total distance and the total number of vehicles within the designated time windows by obtaining the optimized routes for the last mile network. In our scenario, the delivery is inefficient due to the limited number of transport vehicles and scattered geographical drop off points (customers) that do not belong to a particular route, thus making it a case of last-mile delivery. The primary business related metrics considered for this work is the measure of transport distance, travel time and number of transport vehicles. Our approach to solving the last mile logistics problem in this thesis is as follows:

1. Formulate the last mile logistic problem as a vehicle routing problem with time windows (VRPTW).
2. Research alternative ways to achieve the optimal solution for the last mile logistics problem.
3. Develop an appropriate modified swarm intelligence-ant colony algorithm model for the last mile logistic that is competitive in its performance and with an aim to achieve a performance level at par or even superior to the state-of-the-art traditional algorithms.
4. Develop an agent based simulation model and perform a geographical information system (GIS) based implementation of the last mile logistics along with similar evaluation of the simulation model with respect to the proposed metaheuristic and exact algorithm.
5. Perform benchmark analysis of developed algorithm over different dataset and past work on similar problems.
6. Provide a detailed description of the solution addressing the challenging areas within the last mile logistics research area, namely
    a. The absence of formal analysis,

b. The absence of significant comparison to the modern optimization methods.

Chapter 2

Problem definition

As discussed briefly in the last section, the last mile logistics is the least economical leg of the supply chain network and according to L. Ranieri (2018) can cover up to 28% of the total logistics cost. Therefore, the enhancement of last mile logistics and optimized network development are very important challenges for researchers. To understand its impact on the overall cost structure of the supply chain network, this section delves deeper into defining and analyzing the problem that is the “last mile logistics”.

2.1 Conventional problems in last mile network

The last mile being the concluding stage of a transport network has traditionally been the most expensive stage of the delivery process according to Ewan Roy (2018). It is less efficient as compared to moving large quantities of goods from one location to another and requires a considerable amount of vehicle fleet and work force to perform each delivery. The cost per delivery also increases exponentially in sparsely populated areas, thus making it cost-prohibitive for transport companies. Bridget McCrea (2016) in her article mentioned that according to Honeywell, the last mile costs could make up to 50% of total logistics costs on any shipment. The author further mentioned that the last mile segment of a supply chain network is full of challenges such as traffic congestion in urban areas, long delivery routes in distant areas, invalid or incorrect customer addresses and delivery personnel scarcity.

Shipping goods via rail networks or cargo ships are often the most cost-effective methods of shipping. However, when goods arrive at a high-capacity depots or ports, they must then be delivered to their final customer destination. This has become the "last mile logistics problem." According to Martin Joerss (2016) et al. the cost of worldwide last mile delivery, can expanse to about €70 billion with the United States, Germany and China contributing to more than 40 percent of the market. Also being a highly uncertain and dynamic market, growth rates as of 2015 are between 7 and 10 percent in developed markets and more than 100 percent in developing markets.

As e-commerce market continues to expand, the last mile of delivery, usually ending up at the customer location has become more difficult and challenging. Last mile logistics problems have become a widespread area of interest for supply chain companies due to the increasing demand for integrated Omni-channel trading. Developing Omni-channel requirements have forced organizations to evaluate current delivery system capabilities and make modifications accordingly. E-commerce models support in increasing sales and profit, however, also pose many network optimization challenges. Most of the e-commerce business ventures focus on last mile logistics because it is a key differentiator for retailers and supply chain organizations. E-commerce presents a vital shift in how customers shop. E-commerce has customers drawing customized packages to their preferred location, mostly their home or office or a nearby store. The push-to-pull business-to-business (B2B) and business to customer (B2C) model has generated an organizational change in fundamental supply chains and the movement of products, location of resources, logistics mode, and enabling technologies and data analytics. As per United Parcel Service (UPS), 50 percent of its domestic demand and deliveries are the last-mile section for e-commerce orders. E-commerce businesses disbursed up to $30 billion on distribution-center activities alone in 2016

There are several growing concerns in last mile logistics. Starting with the urban logistics models, urban logistics may be challenging due to navigating dynamic traffic and parking restrictions hence dramatically increase last-mile delivery fuel costs. One of the major challenges is delivering the products to customers at a specific and ‘effective’ price, and hence more companies are beginning to offer deliver-from-store services, such as The Home Depot and Sears and J.C. Penny Co, according to the Journal of Commerce. Capacity and inventory control is another problem in last mile logistics. Previous works have focused on the importance and challenges surrounding the capacity crisis and the driver scarcity. Transporters must find means to overcome these concerns and meet the new tasks in last mile logistics to remain competitive. Sometimes the "last mile" has been used to describe the urban public transportation network such as the trouble in getting people from a hub, mostly railway terminals, bus depots, and ferry terminals to their final stop. However, urban public transportation network is not the scope of this work.

It should be stressed that “last-mile costs” in this work refers to the “total cost of ownership” of the last mile, i.e. the total last-mile logistics costs per unit transported. The remaining supply chain costs are not always passed onto the transporter or to the customer. This section discusses how these costs are derived from the moment the goods are dispatched (from the transporters’ last distribution center (DC)) until the moment, it is delivered at the customer’s location. As vendors and brands seek to improve the customer value proposition and strive for greater market share, they are continually working to offer a greater range of products and simultaneously reduce logistics lead times and costs. Therefore, vendors and brands are forward positioning a broader assortment of stock-keeping units (SKUs) to fulfillment and distribution centers, where the selection of discrete SKUs for customer orders takes place. As per an A.T. Kearney (2016) article, Amazon has invested more than $13 billion to establish 50 warehouses or fulfillment centers through the United States since 2010. Thus allowing Amazon to offer a large assortment of stock keeping units (SKUs) and provide customers one-day delivery for all SKUs. Following this trend Walmart has also made noteworthy investments in this area, developing picking or fulfillment centers in major urban areas to support two-day delivery. Other major retailers across different product and service categories have now jumped on the trend and are now positioning inventory closer to customer demand points, with dedicated operations or across mutual facilities managed by third party logistics (3PL) organizations. As one can see that, the last-mile logistics is a big business and companies are willing to spend as much as $65 billion in collecting, packing, and transportation to access the $370 billion e-commerce market share as per A.T. Kearney (2016).

Last-mile shipping and delivery is being done largely through three major carrier segments (UPS, FedEx, and the US Post Office) with regional transporters (such as LaserShip and XPO Logistics), and crowd sourced methods (like Uber) consumed the remaining balance. The three national transporters have an extensive, complex, well-coordinated hub-and-spoke logistics delivery network that enhances service level and reduces logistics costs. E-commerce volumes are then grouped with non-direct-to-customer (NDTC) volumes and further match the delivery efficiency through their extensive fleets of vehicles. The three national transporters account for about 85 % of the last-mile delivery market today. UPS has the largest market share generated about $17 billion just in e-commerce deliveries. A.T. Kearney (2016) reports that direct-to-customer sales would account for about half of UPS's $35 billion revenue for US domestic markets. FedEx also estimated about $7 to $9 billion revenue in e-commerce orders and logistics, accounting for around 40 percent of its total ground transport business.

Regional carriers, on the other hand, operate on a much smaller scale but focus to unlock local interstate routes, for example, a route from western Pennsylvania to Toledo or route between Boston and New York. Regional carrier LaserShip focuses on last-mile logistics along the Eastern Coast of USA through its system of logistics hubs that are tactically located close to both fulfillment centers and major customer demand areas. Regional carriers account for around a modest $4 billion and crowd-sourced systems presently amount for less than 1% of last-mile logistics costs, thus limiting revenues to approximate $0.6 billion in same-day deliveries market.

As consumer expectations for shorter lead times increase, developments in disrupting technologies, supply chain organization and the network outline of last-mile logistics will continue to grow. Same-day delivery has advanced rapidly in recent years, led by urban demand generation. According to B.I. Intelligence (2018), same-day delivery economic values could reach about $3 to $4 billion. From the demand perspective, customers, especially in urban areas, expect quicker delivery of online orders. Products for instant fulfillment or instant consumption such as food and luxury items are suitable for same-day delivery. In a survey conducted by Stifel Financial Corp., about 60 % of modern urban customers expect same-day delivery for their online purchases and more than 60 % of customers are willing to adjust for logistics costs for same-day delivery.

Rapid developments in digital technologies and robotics will continue to impact e-commerce logistics. Stanford University (2016) concludes that artificial intelligence will reinvent logistics and transportation. Statistical machine-learning systems are now being employed, for example, to predict long-term SKU demand and sales forecasting. These machine-learning algorithms incorporated use similar demand figures to estimate SKU levels, and the system is refined and informed through a data feedback loop. Improving long-term SKU demand forecasting is critical as vendors aim to deploy a broad range to online customers. Automated drone deliveries are upcoming as well. Some organizations are piloting robot and drone delivery systems in major cities. Self-driving bots have also been tested and are expected to can carry heavy orders up to 40. They help in cutting labor costs by around 75% through a highly accurate global positioning system and computer vision. Logistics automated bots work well in city environments and have fewer regulatory concerns than drones.

Chapter 3

Literature review

An extensive amount of research has been done in the field of transportation business models, which create last mile logistics and same day delivery situations. Crainic et al. (2009) estimated that the urban last mile deliveries is an idea that targets to enhance urban freight transportation network by concentrating on all stakeholders, customers and events in urban areas. Qiu et al. (2005) documented the last mile city logistics as a vehicle routing problem and as the movement of products across all means of transportation and its related events such as warehousing, packaging etc.

3.1 Related work and contributions in last mile logistics

Traditional optimization techniques to the last mile problem in public transit systems have involved the use of buses, rental bicycling infrastructure, and urban transport methods. Other techniques for optimizing the last mile problem in the urban transport domain include car sharing programs and personal rapid transit. Ford Motor Company (2015) patented a "self-propelled transport mode accessible with vehicle", which was proposed as a last mile traveler solution. Other contributions in the field of last mile research involve developing and researching newer and automated means of transportation. Major e-commerce companies like Amazon and Alibaba have explored and have even installed robotic drones for transporting goods purchased online to customers. Amazon has further set up fulfilment centers and lockers in some major cities as a way of merging packages for same-day delivery.

In case of modern urban delivery networks, Ehimke (2012) precisely relates city logistics with the last mile. He defined city logistics (CL) as “the second leg of the delivery network with an aim to include pickup and deliver goods to customers in last mile (LM). Anderson et al. (1996) reflect LM as an essential part of city logistics and describes it as the delivery of final goods in low demand and at high occurrences. Lindner (2011) researched that LM includes a series of events and procedures that are essential for the logistics process from the delivery hub to the final customer drop point of the transportation network. Morganti (2014) calls LM a “small scale delivery of customer products in a city” due to the requirement of de-bundling large items into smaller packages for final delivery. Morris et al. (2009) elaborated LM as the “pick and drop off points to the end consumer in commercial environments”. Aized (2014) researched LM as the concluding step in Omni-channel business and asserted that LM is one essential step in supply chain networks of business-to-business (B2B) and business-to-customer (B2C) models, and is important for effective and economical final delivery of goods.

While other aspects of logistics can be carried out through different means of transportation like sea, air and rail, the last mile logistics is carried out generally via the road. Last mile logistics normally has an average share of 10% in the total city transport. London (2012) established that 90% of last mile logistics are made by road. Ehmike, (2012) recognized last mile drop offs involve delivery over small distances with pickup trucks. The last mile delivery via water or rail is conceivable in cities where the urban infrastructure allows it using appropriate canals or city rail.

Cost effective solutions for the last-mile logistics have been developed in order to increase profit for ultrafast and narrow time windowed deliveries. Cleophas & Ehmke (2014) have discussed a lot of literature on vehicle routings for last mile problems. Punakivi, Yrjölä, & Holmström (2001) have talked about complex delivery designs like reception boxes; Allen et al (2007) further discussed pick-up-points and locker banks. Ploos Van Amstel (2014) has also focused on advanced solutions within the domain of urban logistics. Paloheimo, Lettenmeier, &Waris, (2014) and Chen, Pan, Wang, & Zhong, (2016) have researched the benefits and uses of crowdsourced transport platforms. Iwan, Kijewska, & Lemke (2016) say that a couple of these techniques focus on customer participation to increase productivity and customer service levels. Since in our work focuses on implementation of routing algorithms and simulated models, they have been discussed in the forthcoming section as a part of the literature review.

3.2 Impact of inefficient routing on last mile logistics

The majority of research and technical work carried out in the field of Operations Research have relied on mathematical optimization algorithms to improve the pick-up and delivery problem. Most research studies aim to solve the delivery problem, traditional traveling salesman or vehicle routing problem in order to accommodate the real-life scenarios like optimum delivery orders, vehicles etc. According to R. Wilson (2007), the total cost of cargo transportation within the urban environment was around US$ 435 billion. In addition, the amount of time spent in traffic congestion because of trucks can be up to 50%. The author stated that the best way to approach the last mile delivery problem is via two ways: request reorganization and joint routing.

Most of the research work has been done in request reallocations, however; in case of optimizing efficient routes, Sprenger and Mönch (2012) worked on optimizing routes for large-scale last mile logistics issues in the German food industry. The authors disintegrated the main logistic problem into a set of vehicle routing problems and optimized them using an Ant Colony System (ACS). Wang et al. presented their research on spreading traditional vehicle routing to contain subcontracting and order exchange in parallel coalitions. The authors elaborated the Pickup and Delivery Problem with Time Windows as a mathematical optimization model to include subcontracting in integrated operational transportation planning (IOTP) by utilizing an ALNS iterative heuristic in a prolonged route-based demand exchange mechanism first suggested by Wang and Kopfer to optimize the mathematical logistics models. The suggested model in the authors work supports the use of an intermodal network approach for delivery that employs the existing transportation methods for delivery. Infocomm Development Authority of Singapore (2014) developed a technique to reduce the last mile logistics cost in a smart urban environment by retrieving delivery data from multiple networks through a Smart Nation Platform. Uber in 2015 and Russell (2015) estimated that by acquiring/renting new vehicle fleets or by organizing crowd-sourcing drivers, some logistics companies could increase their fleet size and increase profit as well as gain market share vis-à-vis keeping the logistics cost low.

DHL Trend Research and Mitrovic-Minic et al. (2004) estimate that the last mile deliveries have increased in demand and volumes. Himelstein (1999) had calculated the computational complexity of the last mile network in case of time windows. The customers consider the time windows as important as the price of the products. According to Laudon (2007) and Li (1994), customers mostly concentrate on the significance of time, price and quality of logistics. Gevaers et al. (2011) calculated the last-mile logistics cost to be within 13 % to 75% in some distinct cases of the total transportation costs. Boonkleaw et al. (2009) also researched that in the newspaper delivery industry the logistics cost amount for around 23% of the total cost.

3.3 Popular optimization techniques used for last mile logistics

This sub-section aims at understanding and developing suitable background to the last mile logistics problem and its traditional optimization techniques. The above-mentioned literature review of the thesis discussed the importance of inefficient routing on the last mile network. To model the last mile delivery problem, this section considers the traditional vehicle routing problem with time windows constraint. The Vehicle Routing Problem with Time Windows (VRPTW) is a problem of finding an optimized combination of routes from an origin node (usually a hub) to serve customers at geographically scattered areas within a specific time window. The particular problem constraints are that:

(i) Every customer is served inside a given time interval

(ii) The capacity constraint of all vehicles serving the individual routes is not exceeded.

The VRPTW is an extension of a common class of computational optimization problems that are known to be NP-complete. Consider a general VRPTW example with infinitely large time windows. In such a case, the minimal vehicle fleet objective relates to resolving the decision version of the multiple bin-packing problems. For a problem example with all consumers sharing identical position and considering an infinite fleet capacity, the problem becomes similar to multi-machine scheduling with discharge times and limits, which is also NP-complete. Finally, consider a problem case with all the consumers having equal time-windows, with an infinite fleet capacity as well. An efficient optimized solution to such a problem logically requires the fleet to minimize the distance traveled in order to be able to attend maximum consumers within the time-windows. In this respect, the problem is formulated as the multiple traveling salesman problem. Based on the above-mentioned assumptions the problem is NP-hard. The Vehicle Routing Problem with

Time Windows is one of the most extensively studied routing and logistics problem in operations research. In essence, the VRPTW summarizes the significant features of common real-world problem of allocating goods across a geographical set of nodes using a fleet of capacity constrained vehicle units and with time-based constraints being enforced at individual deliveries to customers. This provides a basic model to study the outstanding structures of the last mile logistics related real-world problems, data, experiment, and compare the productivity of various optimization methods to tackle these problems.

The Vehicle Routing Problem with Time Windows (VRPTW) is a popular and complex combinatorial transportation problem, which has received significant attention and focus in recent years. The vehicle routing problem has mostly been studied because of its real life applications in logistics and supply-chains optimization. As discussed above, many different versions of this problem have been formulated to model real life routing scenarios. Randomized algorithms, metaheuristics and simulation techniques are widely used in combinatorial optimization for obtaining high quality solutions. They are now becoming gradually more widespread for solving computationally hard combinatorial problems of Operations Research domain, such as constraint satisfaction, planning, satisfiability, scheduling and other application domains.

The approach used in this work is to develop a mixed integer constraint programming model that best replicates the real life last mile network in form of a VRPTW and subsequently develop a modified metaheuristic that performs faster than and as well as the exact algorithms. In this section, the solution techniques are discussed and the approach is further explained through the methodology and implementation sections. A modified version of the ant colony metaheuristic is developed for our problem. The Ant System (AS) first introduced by Dorigo (1992) is a meta-heuristic for NP-complete combinatorial optimization problems and was first employed on the famous Traveling Salesman Problem (TSP). Beginning from the basic Ant System, modifications to the basic algorithm have been proposed to find a global solution. The improved algorithm was then tested again on the last mile logistics VRPTW instances. All the versions of AS have in common a robust utilization of the best solutions found to direct the solution search process; they mostly differ in some facets of the search procedure and control. One of the most effective swarm intelligence based applications has been Ant Colony System (ACS) that presented a specific pheromone trail updating technique used to strengthen the search near the best solution. Bullnheimer et al utilized an AS-like metaheuristic to solve the VRP. Gambardella, e al. and John E. Bell et al. have also tried to solve the vehicle routing problem (VRP) by using an ant colony optimization (ACO) algorithm. The modified algorithm is further explained in detail in the upcoming section.

The approach in this work also incorporates an agent-based simulation model to test and implement the problem in a GIS environment. Agent-based simulation models or multi-agent systems contain of a set of agents characterized by some features that interact with each other through a protocol of appropriate rules in a given simulation environment. ABMs can be suitable to replicate many behavioral systems related to economics, supply chain, human resources and social sciences, where the structure can be designed through a system according to Billari, Fent, Prskawetz, &Scheffran, (2006). According to Axelrod (1997) through agent-based models (ABMs), it is possible executing a situation with its features, estimating and discovering its future developments, testing possible alternative decisions, applying different values for the decision variables and examining the effects of these alterations. According to Billari et al., (2006) due to the complexity of the communications occurring among the agents of the system at an accumulated level, the use of ABMs can aid in understanding basic properties and patterns, which could not be inferred nor predicted by the observation of each entity.

3.3.1 Solution techniques

3.3.1.1 Exact algorithms

Solving and optimizing NP-hard problems to optimality is an important topic in general mathematical optimization. Perhaps, the two most significant methods are the constraint optimization programming (COP) and mathematical programming (MP). Improvements in these approaches allow for addressing problem cases of substantial size for some particular problems. The vehicle routing problem and traveling salesman variants, however, have proved to be one of the more challenging problems to be solved to optimality and therefore only medium sized problem cases can usually be solved to optimality.

Lagrange relaxation

A number of research papers that use slightly different versions of the problem were solved using Lagrange relaxation. In optimization, Lagrangian relaxation is a technique, which approximates a problem of constrained optimization. This technique uses a Lagrange multiplier that penalizes violations of inequality constraints. Thus, the relaxed problem can be then solved less expensively. Fisher et al. (1997), Kohl and Madsen (1997) and Holland, (1975), researched the shortest path with cross constraints method followed by Lagrange relaxation. The authors also included flexible splitting followed by Lagrange relaxation and a K-tree technique. Fisher et al. (1997) presented an algorithm for optimizing the VRPTW problem to optimality where the problem was modeled as a $2K$-tree problem.

Dynamic programming

In optimization, dynamic programming technique mostly refers to restructuring a decision by disintegrating into a series of decision phases with respect to time. This is done by describing a series of value functions, with an argument demonstrating the state of the structure at given times. The value function of earlier values can be found by using a recursive method called the Bellman equation. Kolen et al. (1987) and Christofides and Beasley (1984) utilized the dynamic programming method for VRPTW for the first time. Kohl and Madsen (1997) used a branch-and bound algorithm to achieve optimality.

Integer optimization

An integer optimization technique is in which some or all of the decision variables are constrained to be integers. In several instances, it refers to integer linear programming (ILP), as the objective function value and the constraints are linear and integral. The important integer optimization models involve the following techniques:

- **Branch and bound:** A traditional integer optimization algorithm designed for discrete optimization problems and mathematical optimization. The branch-and-bound algorithm contains a systematic list of possible solutions by means of state space search. Set of possible solutions is considered a tree with the full set at the root. The algorithm searches branches of this tree, which signify subclasses of the solution set. Before computing the possible solutions, the branch is crosschecked against an upper and lower bound on the optimal value, and is rejected if it cannot yield a better result than the best one found. The algorithm operates on the effective calculation of the upper and lower bounds of a branch in the search area and arrives at complete enumeration as the n-dimensional volume of search area tends to zero. A. H. Land and A. G. Doig suggested the technique in 1960 for discrete value programming and has become the most widespread algorithm for solving NP-hard problems.

- **Branch and cut**: The branch and cut technique is another combinatorial optimization method for solving integer linear programs (ILPs). Branch and cut algorithm comprise of running a branch and bound algorithm as well as using cutting planes to restrict the linear programming relaxations. The technique solves the linear program problem without the integer constraint by using the simplex algorithm. When a good quality optimal solution is achieved, and the solution has non-integral value for the decision variable that is supposed to be integer, the cutting plane algorithm may then be used to find more linear constraints. The branch and bound part of the algorithm is then initiated at this point. The problem is further split into multiple versions. The new relaxed linear programs are then solved again using the simplex method and the iterations continue. During this process, the non-integer solutions to the linear relaxations work as upper bounds and the integer solutions work as lower bounds. A branch node can then be pruned if an upper bound is lower than a prevailing lower bound. Additionally, when solving the linear relaxations, more cutting planes may be generated, which serve as global cuts.
- **Cutting plane**: Is another optimization method that through iterations improves a viable set of solutions or objective value function by generating linear inequalities, called cuts. Such

procedures are commonly used to find integer solutions to mixed-integer linear programming (MILP) problems. Ralph E. Gomory presented the cutting planes algorithm to solve integer linear programs. Cutting plane algorithms for MILP work by performing a linear relaxation of the given integer program. In this technique, a cut is then added to the relaxed linear program. Hence, the non-integral value is not viable to the linear relaxation. The process is repeated until a feasible and optimal integer solution is found.

3.3.1.2 Metaheuristics

A metaheuristic is powerful method applicable normally to a large number of problems. A metaheuristic denotes an iterative strategy that controls and transforms the operations of minor heuristics by combining intelligently diverse concepts for exploring and exploiting the solution search space. A metaheuristic may influence a complete or partial solution or a group of solutions at each iteration. The family of basic metaheuristics usually contains the popular simulated annealing, Tabu search and genetic algorithms.

Simulated annealing

Simulated annealing algorithm theoretically is similar to the mechanical process, known as annealing, where a solid is heated into a liquid state then chilled back into a recrystallized solid state. In this search technique, the heuristic does not search for the best values in the vicinity of the current value. Instead, it simply draws at random a value from the vicinity. If the value is better than the previous one it is accepted as a new solution, but if the value is worse than the present solution it is only established with a certain probability. This acceptance probability is calculated by a temperature heuristic, which is slowly decreased. By decreasing the temperature, heuristic the selection procedure becomes more and choosier in accepting a new solution. Chiang (1996) developed the three different simulated annealing heuristic methods, which were modified form of the k-node interchange mechanism. Osman (1993) used the similar approach with $k = 1$.

Tabu search

The Tabu search is one of the most commonly used metaheuristics. Introduced by Glover (1989) and Fisher et al. (1997) in Tabu search at the end of each iteration the vicinity of the current value is explored and the best possible solution in the vicinity is chosen as the new current solution. To allow the algorithm to leave a local optimum the current value is set to the best possible solution in the vicinity even if this value is inferior to the current solution. To prevent degeneracy, searching recently chosen solutions are forbidden. This is done by implementing a Tabu list. The Tabu list overruling criteria is called aspiration criteria. Solomon (1987) is used Tabu list to produce the initial solution, in the author's work algorithm moves between the two policies in order to reduce the number of routes the author's algorithm tries to transfer consumers from routes with few consumers to other routes.

3.3.1.3 Evolutionary algorithms

Particle swarm optimization (PSO), ant colony systems (ACS), evolutionary algorithms, and their modifications dominate the area of nature-inspired metaheuristics. A large number of nature-inspired metaheuristics have started to attract attention in the research community.

Genetic algorithms

The genetic algorithms are probabilistic techniques of optimization, which acquired as an initial point in the genetic evolution of a species. The main idea is to reproduce the natural growth of genes generation after generation, by following the phenomena's inheritance and the law of survival defined by Darwin. The first implementation of the genetic algorithms goes back to 1950 when biologists replicated the evolution of the genes. Subsequently, Holland (1975) and Fisher (1994) modified the genetic algorithms to resolve combinatorial constrained optimization problems. As opposed to simulated annealing and Tabu search, genetic algorithms function in a population of the possible solution according to Pirlot (1996). A population of solution genes, each one corresponding to a possible value, represents the search space of the solution. The new genes are generated by use of genetic operators (natural selection, crossing and mutation) on the likely parents chosen from the initial population. The genetic algorithms work on the principle that best parent's solutions produce best children solutions; so the most robust genes of the population have a high probability to be chosen as the parents. The best parent solution are crossed to produce new children who switch the parent solution or the weak solutions of the population. The procedure is repeated and iterated until a population is achieved where all the individuals are very good candidate solutions, analogous to the optimal solution of the problem.

Swarm Intelligence

Swarm intelligence heuristics consist usually of a population of simple agents networking locally with each other and with their environment. The inspiration often comes from biological systems. The agents follow basic rules, there is no centralized control protocol dictating the behavior of individual agents. Examples in natural systems of swarm intelligence include ant colonies, bacterial growth, fish schooling, bird flocking and microbial intelligence. Several widespread algorithms based on these models, comprise of Particle Swarm Optimization (PSO), Ant Colony

System (ACS) etc. and in this work, we utilize a modified version of the ant colony metaheuristic for determining the global minima.

3.3.1.4 Agents-based approaches to optimization

In recent years, several papers proposing agent based models or ABM-based approaches to optimization problems have been published. Generally, such methods combine agent-based models with other optimization methods, distributed or complex systems, heuristic methods etc. Johnson et al. in their research described three most common forms of integration of the optimization, simulation and agent-based models:

- Optimization as a standardization and validation means for ABM,
- ABM used to optimization problems,
- Optimization of ABM to denote constrained maximization.

In this chapter, ABM is used as a technique for solving hard optimization problems. There are a number of multi agent systems (MAS) approaches proposed in the operations research works to solve different types of optimization problems.

In the field agent based modeling there are two different types of agents with different functions with respect to optimization. The first class of agents represent or directly supply the essential resources within optimization model. Agents of the second class perform certain distinct functions. The first category comprises of mostly physical agents such as vehicles, workers, products, resources or machines. The second class, mostly represent parts of software used to perform orders or subtasks such as search strategies, local and global optimization. Barbati et al. reviewed in his research several methods to use agents for solving constrained optimization problems. The author also identified agent-based architectures used to solve scheduling problems. In the independent architecture, many agents organize themselves to solve a problem based on negotiation and contract protocols. Persson et al. and Davidsson et al. compared pros and cons of agent-based approaches versus classical optimization techniques. Their research focused on assessing how well both methods are able to handle some significant properties of the problem domain. Authors in their work proposed a set of parameters allowing comparison of agent-based approaches and classical optimization techniques, including:

- problem size
- cost of communication
- communication and computational stability
- modularity
- time-scale
- variability
- quality of solution
- quality assurance
- integrity

Based on the analysis by Persson et al. and Davidsson et al the properties of the agent-based approaches and optimization techniques match each other and there are a number of ways for merging them. They described two approaches as:

- Supporting agents with a strategy achieved by using optimization techniques. The method can be described as using optimization technique for planning and agents for performing local modifications of the initial strategy in real-time to manage the actual situations when the strategy is executed.

- Embedding optimization algorithms within agents.

The above-mentioned literature illustrates the current state of the art related to the use and to the application of agent-based models as optimization techniques. One of such optimization tools is the concept of an A-Team, introduced by Talukdar et al. The notion of the A-Team has been to develop a software environment for solving a range of computationally hard optimization problems. A java programming language based JADE-based A-Team (JABAT) system supports the creation of the devoted A-Team architectures. Agents used in JABAT guarantee decentralization of computation across many hardware platforms. Parallel processing platforms result in the more effective use of the accessible resources and eventually, a reduction of the computation time.

Chapter 4

Research methodology

In this chapter, the formal framework is provided and used throughout this work along with an overview of the problem. Having defined the problem in the earlier sections and discussed some suitable optimization techniques utilized in modern supply chain optimization in the previous sections, the aim of this section is to develop a generalized approach to the last mile optimization problem.

4.1 Overview

As the problem is a generalized version of the vehicle routing problem with time windows, the approach here is to consider data that replicates the last mile routing with time windows and develop the suitable metaheuristic algorithm and the simulation model for experimentation. The last mile VRPTW problem is modeled for our different scenarios with a strategic plan of minimizing total traveling cost/distance and within constrained time. In this section, the last mile problem is modeled as an integer-programming model. A cost model is also developed to focus on the importance of time and distance parameters on the logistics cost of the last mile delivery. Post development of the optimum real-life model we intend to solve the model using a traditional branch and bound algorithm and our novel ant colony meta-heuristic. Our modified ant colony algorithm works by searching an optimal path in a graph, based on the behavior of ants that search for food by laying down pheromone trails. When developing the model for an ant colony meta-heuristic ant colony, we subjected the parameters to numerous iterations to determine the optimum values of the selection parameters. To model the ant-colony optimization algorithm the following critical parameters in the ant-colony optimization algorithm were determined using an iterative search tuning process. We will further elaborate on our modified ant-colony metaheuristic that aims to locate the global minima. The second part of the methodology and implementation contains the development of an agent-based simulation model using AnyLogic simulation platform and utilizing the global server to replicate real life locations and road networks.

4.2 Problem setup

We begin by developing our problem definition by sourcing a test dataset of supply chain network beginning with the last mile logistics network for the local newspaper delivery for State College, PA. Newspaper distribution by the Daily Collegian office in State College region lacks an optimal plan for vehicle routes from the distribution center to the drop off points. This had made route selection a tedious and confusing task; it also leads to high delivery costs and operational inefficiency. Hence, in our problem, we propose a mixed integer-programming model that attempts to minimize the total distance of the delivery vehicles of the newspaper agency. We model the delivery problem as a vehicle routing problem with time windows, as newspaper are a perishable commodity and need to be delivered in a constrained time window. The vehicle routing problems is a generalized version of the travelling salesperson problem and is one of the most widely studied problems in mathematical optimization. In our model, we have considered only the distribution of newspaper between a Distribution Center (DC) to drop points, not to final customers and within a designated time window. Our last node/city/drop-off point in this model represents the last mile delivery node. The goal of this problem is to optimize the delivery of newspapers to appointed addresses in State College town. The delivery man/vehicle has to deliver packages using the least distance/ cost by finding the shortest route moving from the distribution center. We develop a modified ant colony metaheuristic from our State College newspaper delivery dataset and subject to performance evaluation against an exact algorithm. We then perform parametric evaluation of our algorithm on benchmark datasets to test the heuristic and solution quality. We further develop an agent-based simulation model and perform a GIS based implementation of the last mile logistics along with the performance evaluation of the simulation model.

4.3 Model formulation

In this section, the method to develop a mathematical model for our generalized problem has been discussed. The aim of this section is to explain the mathematical depiction of the last mile logistic problems and to define our modified ant colony optimization for a global optimum for our last mile logistics datasets. The approach and modeling details of the agent-based simulation has been discussed for our last mile datasets

4.3.1 Last mile logistic routing problem model

We begin by developing an integer-programming model for State College newspaper delivery dataset. Some assumptions are required in the model to replicate the real-scenario:

- Routes will start from and end at the depot.
- Vehicle should enter and leave each customer node exactly once.
- The distance travelled is proportional to the cost of the route.
- Demands at each of the customer nodes are known and constant.

Other hard constraints:

- Time windows
- Customer service duration
- Customer ready time

- Customer due time

Our mixed integer program model for the last mile logistic routes is as follows:

The model is given by a fleet of homogeneous vehicles denoted by V, a set of customers C and a directed graph $G = (V, C)$. The graph consists of $|C| + 2$ vertices, where the customers are denoted $1,2, \ldots\ldots n$ and the depot is represented by the vertex 0. The VRPTW has several objectives where, the goal is to minimize not only the cumulative number of vehicles required, but also the waiting time, total travel time, and cumulative travelled distance by the fleet of vehicles. The set of arcs denoted by A represents connections between the depot and the customers and among the customers. No arc ends in vertex 0, and no arc starts from vertex $n + 1$, each arc (i, j) where i, j we assign a cost c_{ij} and a time t_{ij}, which may comprise of service time at customer. Each vehicle has a capacity q and each customer ia demand d_i. Each customer i has a time window (a_i, b_i). A vehicle needs to arrive at the customer before b_i. It can arrive before a_i but the customer will not be serviced before. The depot also has a time window (a_0, b_0). Vehicles may not leave the depot before a_0 and must be back before or at timeb_{n+1}. It is assumed that q,a_i,b_i,d_i, c_{ij} are non-negative integers, while the t_{ij} are assumed to be positive integers. It is assumed that the triangular inequality is satisfied for both the c_{ij} and t_{ij}, the model contains two sets of decision variables x_{ijk} ands_{ik} . For each arc (i, j) where $i \neq j, i \neq n + 1, j \neq 0$ and each vehicle k we define $x_{\mathrm{ijk}} = 1$ if and only if the optimal solution, arc (i, j) is traversed by vehicle k and equal 0, otherwise. The variable s_{ik} is defined for each arc vertex i and each delivery vehicle k and denotes the time delivery vehicle k starts to serve customer i. In case the given vehicle k does not service customer i, s_{ik} does not mean anything. We assume a_0 =0, and therefore $s_{0\mathrm{k}}$ =0, for all k. The aim is to develop a set of low cost routes, for each transport vehicle, such that each customer position is visited exactly once, and all routes originate at node 0 and ends at node $n+1$, and the time capacity constraints are observed. We formulate the last mile problem as a mixed integer-programming model as follows:

$$min \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij} x_{ijk}$$

$$subject\ to$$

$$\sum_{k \in V} \sum_{j \in N} x_{ijk} = 1 \quad \forall i \in C \ \ (1)$$

$$\sum_{k \in V} d_i \sum_{j \in N} x_{ijk} \leq q \quad \forall k \in V \ \ (2)$$

$$\sum_{j \in N} x_{0jk} = 1 \quad \forall k \in V \ \ (3)$$

$$\sum_{i \in N} x_{ihk} - \sum_{j \in N} x_{hjk} = 0 \quad \forall h \in C\,, \forall k \in V \ \ (4)$$

$$\sum_{i \in N} x_{i,n+1,k} = 1 \quad \forall i \in C \ \ (5)$$

$$s_{ik} + t_{ij} - K\big(1 - x_{ijk}\big) \leq s_{jk} \forall i,j \in N, \qquad \forall k \in V \ \ (6)$$

$$a_i \leq s_{ik} \leq b_i \forall i \in N, \qquad \forall k \in V \ \ (7)$$

$$x_{ijk} \in \{0,1\}, \qquad \forall j \in N, \qquad \forall k \in V \ \ (8)$$

The constraint 1 states that each customer is visited exactly once, and 2 means that no vehicle is loaded with more than its capacity allows it to. The next three equations ensure that each vehicle leaves the depot 0, after arriving at a customer the vehicle leaves again, and finally arrives at the depot $n+1$. The inequalities 1 states that a vehicle k cannot arrive at j before $s_{\mathrm{ik}} + t_{ij}$ if it is travelling from i to j. Here K is a large scalar. Lastly, constraint 6 ensures that time windows are satisfied, and constraint 7 are the integrality constraints. Note that an unused vehicle is modeled by driving the empty route $(0, n + 1)$.

4.3.2 Routing cost model

In this subsection, we discuss an informal framework and model that breaks down the constituents a basic last mile logistic cost. Blauwens, De Baere & Van de Voorde (2010) nevertheless, from their research, data from interviews and literature, developed a standard last-mile logistics cost model based on a delivery time function and a distance function. Roel Gevaers et al. then came up with a more thorough and comprehensive cost model for last mile logistics. This thesis utilizes the similar concept and a basic last mile cost has been developed. The generalized model for the cost function is:

$$LMC = T \times a + D \times b + X + \varepsilon$$

Where:

- LMC stands for last mile logistics cost
- T is the duration/time
- a is the time/hour coefficient
- D is the distance travelled
- b is the distance coefficient
- X is the additional costs
- ε stands for a random test error $\sim N(0, \sigma^2)$

This work uses the same cost metrics for performance evaluation of agent-based model and the modified metaheuristic. The work involves modelling and solving the novel last mile logistics/delivery problem by considering the above mentioned cost model through a modified metaheuristic and a comparable agent-based simulation model by considering the delivery challenges for multiple variants of last mile networks. The last mile logistics challenges become more intricate when customers demand a smaller delivery time window. Effective delivery routes in a network, serve as an important factor by many businesses since low-cost logistics helps them attain a higher contribution and profit margin from existing sales. Furthermore, various governments and businesses can reduce distribution cost by deploying delivery-vehicle routing and optimized route models. Optimal delivery routes from the distribution (depot) center to the entire drop off points will resolve the problem of high logistic costs and of late deliveries. Such optimal vehicle delivery routes can also avoid excessive fuel consumption and reduce CO_2 and other toxic gas emissions, thus contributing to a more sustainable environment.

4.4 Algorithms for problem

4.4.1 Evolutionary algorithms: Ant colony

Ant colony optimization (ACO) is a meta-heuristic for solving hard combinatorial optimization problems. Colorni et al. proposed the first ACO algorithm, ant system (AS), as a means of solving the travelling salesperson problem or the vehicle routing problem. AS draws a similarity between the optimization process and the searching behavior of real ants. Based upon observations of real ant system colonies, it was found that ants have the capability to find the shortest cost effective path between their home and a food (F) source.

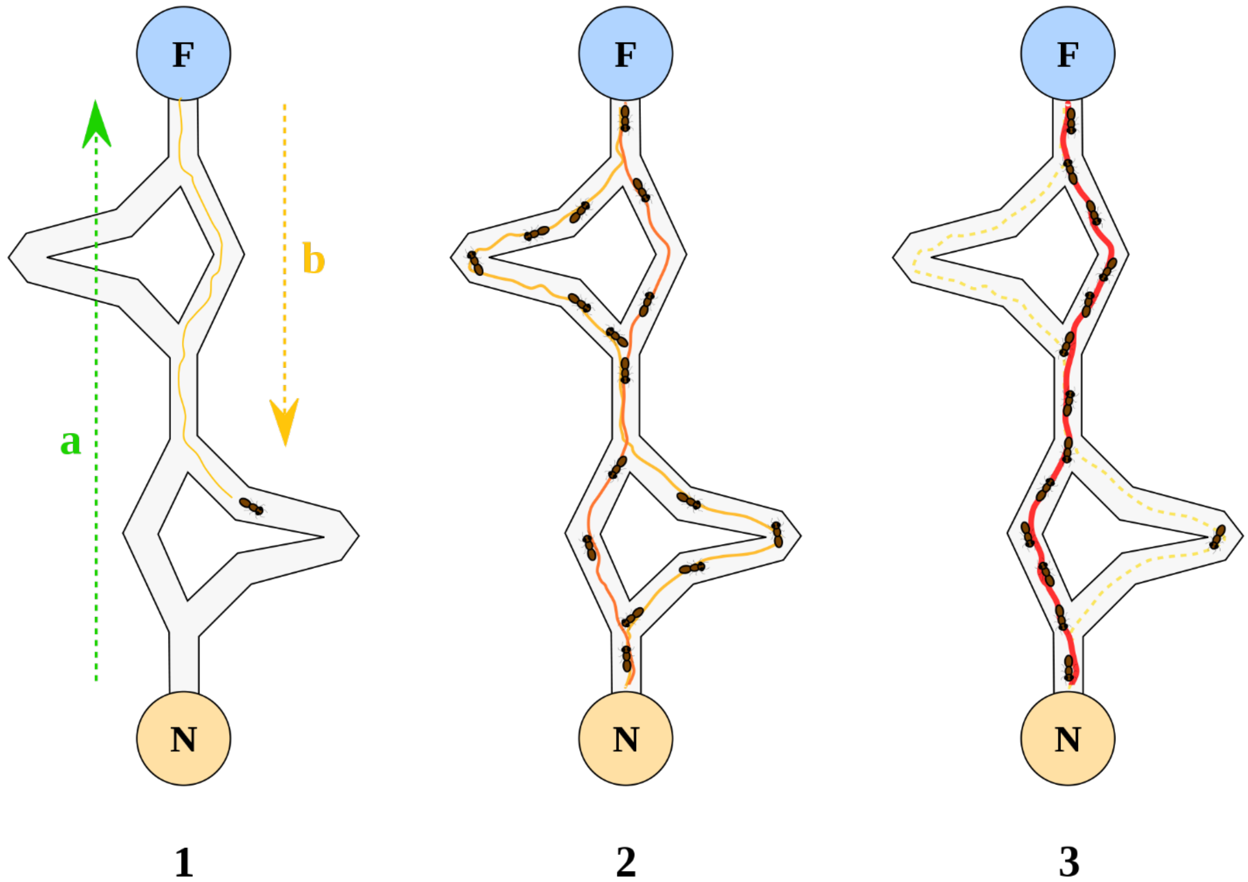

Figure 1: The ant-colony optimization solution method [33]

Furthermore, while traversing between their nest and the food source, the ants lay down a chemical substance called pheromone along the paths. Different entities or ants traversing the path then sense the pheromone left by earlier ants and tend to trace the trail with a robust pheromone concentration. Over a certain period, the shorter routes between the nest and the food source are likely to be traversed more than the longer paths. Hence, the shorter paths gather a greater amount of pheromone, which attracts more ants to the route, thereby further strengthening the path.

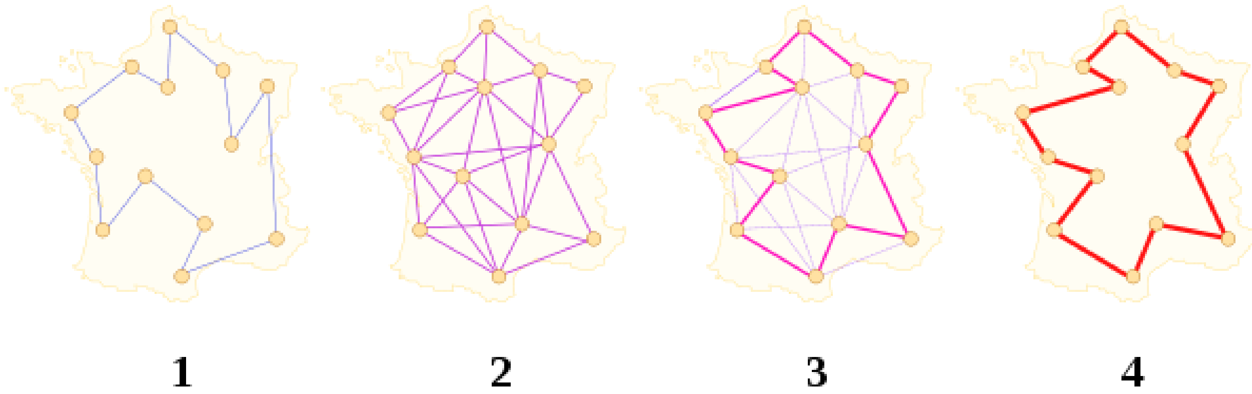

Figure 2: The ant colony optimization of the travelling salesperson problem [34]

To solve the VRP, the ants build vehicle routes by continually choosing nodes to visit, until each node has been visited. Whenever there is option of another city, it would lead to an infeasible value because of vehicle capacity or route length, the depot is chosen again and a new route is assigned. At each iteration, every ant k calculates a set of feasible extensions to its current solution and chooses one of these probabilistically, according to a distribution specified as follows. For ant k the probability p of visiting customer j after i, the last visited customer, depends on the combination of two values. The heuristic uses a population of m ants, which develop solutions step by step.

When all the ants have created their tour, the best solution is chosen to encourage the search for even better values in the next iterations. The most important element of an ant system is the laying down of pheromone trails. In a typical ant colony system, pheromone trails are used in combination with the optimizing value for creating a new solution. The information kept in the pheromone trails and the implementation of this information is the key component of an ant system.

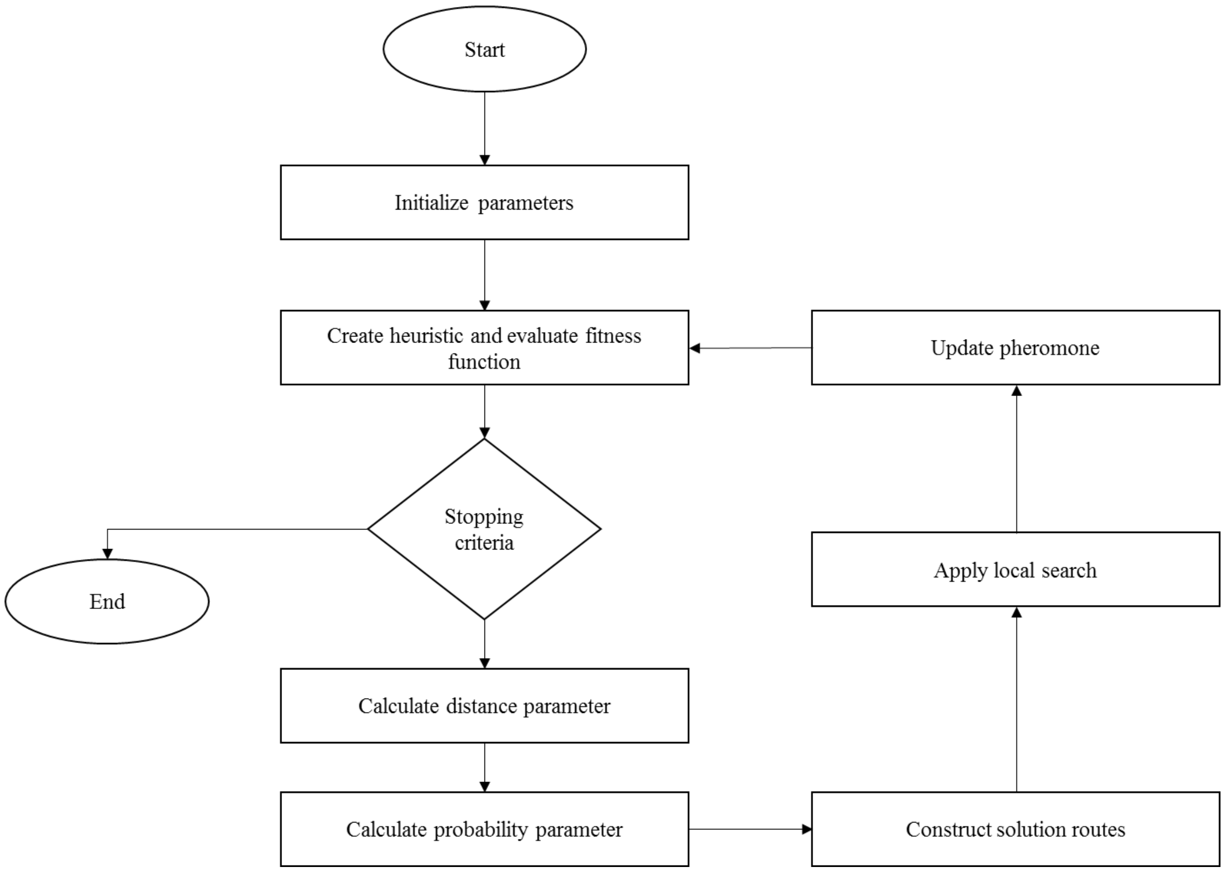

Figure 3: The ant-colony optimization algorithm flow chart

The algorithm works by searching an optimal path in a graph, based on the behavior of ants that search for food by laying down pheromone trails. When developing model for a swarm intelligence meta-heuristic ant colony, we subjected the parameters to numerous iterations to determine the optimum values of the selection parameters. To model the ant-colony optimization algorithm the following critical parameters in the ant-colony optimization algorithm were determined using an iterative search tuning process. (α = relative value of the trail, β = relative value of the visibility & ρ = evaporation rate).

- Visibility levels:

$$\eta_{i,j} = \frac{1}{d_{i,j}}$$

- Probability of selecting address j from i for ant k:

$$p_{i,j}(t) = \frac{[\tau_{i,j}\,(t)]^{\alpha} *\ [\eta_{i,j}]^{\beta}}{\sum_{k \in alw_k}[\tau_{i,k}\,(t)]^{\alpha} *\ [\eta_{i,j}]^{\beta}}$$

- Pheromone update equation:

$$\tau_{i,j}(new) = \ 0.1 * \rho * \ \tau_{i,j}(old) + \ \sum_{k=1}^{m} \Delta\tau_{i,j}^{k}$$

In the results and discussion section, we have illustrated the selection of optimum parameters for the ant colony model.

4.4.2 Optimization approach: Proposed modified ant colony metaheuristic

To solve the last mile logistic model that we formulated, we developed a novel modified version of the ant colony optimization metaheuristic. This novel algorithm is modified to obtain a global optimized value to the problem.

In the proposed algorithm, first, the number of m ants being associated with m random route vectors $(a_{initial}^{x}, (k = 1, 2 \ldots m))$ (or all of them may be set to the same value). Then, modifications based on the pheromone trail are then applied. In the proposed ant colony based algorithm, quantity of pheromone $\tau_{i,j}$ only intensifies around the best solution value obtained from the previous iteration and all ants turned towards there to search a solution. The solution vector of each ant is updated at the beginning of each iteration using the following formula:

$$a_t^x = a_{t-1}^x \pm dx \; (t = 1,2 \ldots I).$$

Where a_t^x is solution vector of the k^{th} ant at iteration t, a_{t-1}^{best} is the best solution obtained at the iteration $t-1$ and dx is a route vector from a set $[-Q, Q]$ of range to determine the length of step size. At the end of each iteration, the quantity of pheromone $\tau_{i,j}$ is updated. First, the quantity of pheromone $\tau_{i,j}$ is reduced to simulate the dissipation process with the following formula:

$$\tau_{i,j}(new) = 0.1 \times \tau^2(old)$$

Then, it is only increased around the best solution value obtained from the previous iteration.

$$\tau_{i,j} = \tau^2(old) + (0.1 \times f(a)_{t-1}^{best})$$

This process is repeated till a stopping criteria of the number of maximum iteration (I). In formula (+) sign uses when point a_t^x is on the left of minimum. On the other hand, (-) sign uses when point a_t^x is on the right of minimum. The direction of movement is defined by:

$$\bar{a}_{initial}^{best} = a_{initial}^{best}(\pm)(a_{initial}^{best} \times 0.1).$$

Setting $Q(new) = 0.1 \times Q(old)$ at the end of each iteration I to not pass over global value (I is number of maximum iteration). Thus, the length of jumping will gradually decrease.

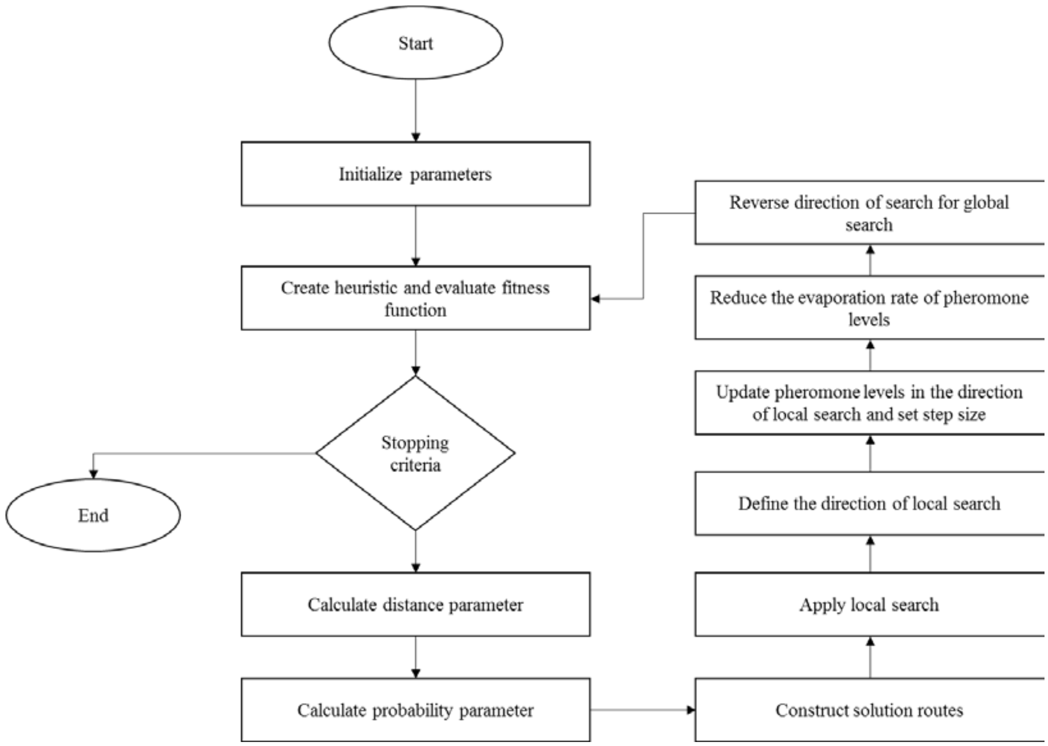

Figure 4: The modified optimization algorithm flowchart

4.4.3 Optimization approach: Branch and Bound algorithm

4.4.3.1 COIN –OR (Branch &Bound)

The COIN Branch and Bound (B & B) solver is an open-source mixed-integer linear program (MIP) solver written in C++ programming language and is used primarily as a library to create modified branch-and-bound solvers. The solver relies on the COIN Open Solver Interface (OSI) and an LP solver to communicate with the user's preference of solver. For branch generation, B& B uses the CGL Library (CGL). We briefly describe the basic branch-and-bound algorithm by way of following steps:

Step 1.Given a MILP model to optimize where some variables have integer values, linearly relax the integrality requirements. Obtain a lower bound on the MILP’s objective value by solving the relaxed problem by an LP solver. If the LP solution has integral results for the MILP's integer variables, we are optimal. The MIP-feasible solution gives an upper bound on the MILP's objective value. The solution is optimal when the lower and upper bounds are equal.

Step 2. Select an "integer" value with a non-integer branch. Generate two branch nodes, with the branching value having a lower bound and the other with branching value having an upper bound. Add the two branch nodes generated to a search solution tree.

Step 3. Select a generated node off the tree

Step 4. Develop another LP relaxation and solve.

Step 5. Check the optimal LP solution; prune the generated node by one of the following.

- If LP solution is infeasible, prune the node.
- Else, if the optimal LP solution value of the generated node exceeds the upper bound, prune the node.
- Else, if the optimal LP solution of the generated node does not exceed the upper bound and the value is feasible to the MILP. Update the upper bound and the value solution, and prune the node.

Step 6. If unable to prune the node, then generate branch. Select non-integral variable to branch on and add to search tree. This is the framework of a "branch-and-bound" algorithm.

4.5 Optimization approach: Agent based simulation model

Agent-based simulation (ABS) is a method that models the individual entities as agents in a simulation. Such a simulation technique is particularly useful for operations research modelers when it allows searching optimum parameter configurations of an optimization-simulation model.

Accordingly, the main research question answered in this paper is the relevance and use of an ABS optimization technique for an NP-Hard routing optimization problem. Similar to the optimization model proposed by Golden (1987), this thesis proposes a multi-objective VRPTW model, combined with capacitated vehicle constraints. To model the real-life VRPTW scenario as realistic as possible, in this thesis, the agents have been categorized into:

1. Depot
2. Customers
3. Vehicles
4. Orders

The single agent distribution center or the depot transport goods to a population of agents modelled as customers with multiple agents as vehicles with their capacity being constant. The demands generated by the customers are modeled as an order agent and the locations are deployed on a GIS map. The time windows are also modeled in the order agent. The logic flow process and implementation have been discussed in the upcoming section. The aim is to select the optimal routes, where three optimization objectives or performance metrics are considered:

1. Minimize the total distance of all assigned routes
2. Minimize the average number of vehicles assigned
3. Punctual service levels within the time windows (this work does not include time constraint as a performance metric).

4.6 Benchmarking

As discussed earlier in the section, the thesis also employs a benchmarking mechanism to compare the performance of the proposed heuristic and the agent based simulation model. Having discussed the type of dataset earlier, the heuristic used for the same has been discussed in detail below. The two common and high quality heuristics used for the benchmarking process are:

4.6.1 Benchmarking algorithm: Solomon Heuristic

Solomon's 100-customer cases are divided into six problem groups depending on the Euclidean distribution of the customers and the width of the time window horizon. The latter is set by a time window at the hub, which outlines the earliest start time and latest end time for each route. The customers are randomly distributed in cases of type R, while they are clustered in cases of type C. Cases of type RC are a blend of types R and C. With regard to the time windows, cases of type 1 have a short window and each vehicle can only visit a small number of customers. On the other hand, cases of type 2 have a long horizon and each vehicle can visit a large number of customers. Hence, these characteristics lead to classes R1, R2, C1, C2, RC1 and RC2, with 8 to 12 cases in each class. All cases are in Euclidean space and the time units corresponds to the distance units. The cumulative number of vehicles (CNV) and cumulative total distance (CTD) over the 56 cases are determined for each method. The heuristic used in the Solomon example is a specific case of route building. The heuristic algorithm begins with all probable single-customer route. In every step we compute which two routes can be shared with the maximum saving between customers i and j. A time focused nearest-neighborhood algorithm is established by describing the savings as a combination of distance and distance. In Solomon (1987), the time feature is not part of the savings function. Instead, the routes are limited by how large the waiting times get.

4.6.2 Benchmarking algorithm: Homberger Heuristic

Gehring and Homberger had considered larger VRPTW cases with 200, 400, 600 and 1000 customers' locations. For each size, six different categories, with 10 instances in each category, were generated using the Solomon's technique. Thus, there are 60 cases for each problem size and

300 cases. Gehring and Homberger (1999) developed a similar two-phase evolutionary algorithm with Tabu search. In the initial research, the heuristic search algorithm developed an evolution system to reduce fleet size, while in the second part of the research the Tabu search heuristic is applied to travelled distance minimization. During the development process, the recombination of individuals is avoided, with one offspring generated per parent via mutation. For the assessment of individuals, a modified fitness function is utilized. It consisted of two values: the number of customer served by the vehicle and the sum of marginal delays of these customers. The two-part solution method is parallelized assuming cooperative autonomy, while each autonomous search thread is performed with different structure settings.

Chapter 5

Implementation

In this section, the proposed algorithm has been implemented and the agent-based model has been developed using the AnyLogic simulation tool. For modelling, we used python IDE (spyder) for the ant colony metaheuristic and ‘google optimization’ tool python for the exact branch and bound model. Post developing the meta-heuristic model we tuned our ant colony meta-heuristic model to achieve the optimum values for our parameter. We employed the ‘ACO-Pants’ python library for the modelling of the algorithms. The results for the parameter tuning can be found in the figures below. We then subjected the models to numerous iterations by considering the different addresses from our dataset (the depot and the last mile address remained the same in all cases) to test for heuristic quality. In case of the AnyLogic agent based model, the simulation model was developed by considering entities like customers, depot, order and vehicles as agents and placing them on a GIS scale. Both our models were processed on Mac OS High Sierra with 2.6 GHz quad-core Intel Core i7, Turbo Boost up to 3.5GHz, with 6MB shared L3 cache and with memory of 16GB of 2133MHz LPDDR3

5.1 Proposed modified ant colony metaheuristic implementation

For part 1 of the experimentation that we conducted for determining suitable parameter values, we observed that the algorithm was robust in the sense that fairly good solutions could be obtained for varying parameter values. This is mostly due to the contribution of the local search strategy to the overall solution quality. To illustrate the effect our algorithm we provide in Figure 5-8 the results for multiple sample runs of a problem instances of the Dataset 1 described in our methodology section. We see that the global search procedure provides a significant improvement in the total distance traversed. The figures report the average distance for different values of the

- heuristic information parameter β,
- evaporation rate ρ,
- pseudo-random parameter z, and
- number of best ants used for pheromone update.

These solutions are achieved by changing the value of one parameter only and keeping the remaining parameters at constant values. As expected, changing the values of several parameters simultaneously would lead to poor solutions.

5.1.1 Parameter tuning

Parameter tuning is the process to find and control the correct combination and values of an algorithm's parameters for each individual problem. Researchers have differentiated between parameter tuning, Dorigo et al. have tested with parameter tuning for ACO to solve VRPTW. They gave procedures on the boundary of the basic parameters in ACO:

- α: the relative value of the trail, $\alpha \geq 0$
- β: the relative value of the visibility, $\beta \geq 0$
- ρ: evaporation rate, $0 \leq \rho < 1$

From the fact that researchers are trying to tune the parameters in ACO and they are using different parameter values in ACO:

1. The performance of ACO is impacted by its parameter values. This is because parameters can indirectly determine the amplification and diversification of the search process.
2. There are no universal defined parameter values, which can be used in ACO to solve all constrained optimization problems efficiently. The differences in optimal parameter selection come from different problem classes, optimization problem type, and problem instance.

The nature of each parameter in ACO affects intensification and diversification. They are as follows:

- Number of ants m Ant number m determines the diversification in the solution space. A greater ant number will cause a broader search space. This is due to the expectation that more areas will be explored.
- Relative value of pheromone trail α and heuristic value β. These two parameters control the intensification and diversification based on the pheromone trail distribution and heuristic value distribution. If the distribution of pheromone trail is highly congregated, then α will effect ACO to intensify towards past ants, otherwise ACO will implement diversification. Similarly with β, if the distribution of heuristic value is highly congregated, then β will influence ACO to increase towards heuristic value, otherwise ACO will implement diversification.
- Evaporation rate ρ determines how important the new information gathered by ants in newer iterations by assigning a weight $\rho * (1 - \rho) * (n - k)$ to the pheromone trail values gained by iteration k-th at iteration n, with a condition $n > k$. Therefore, ACO only considers pheromone trails from iterations, which have a significant value of $\rho * (1 - \rho) * (n - k)$. A high value of evaporation rate will reduce the number of

significant iterations and vice versa. The intensification and diversification scheme is dependent on these significant iterations.

- Parameter transition rule manages the choice of ants, whether to intensify with a probability p_0 based on value components, which maximize $\tau_i, \alpha, \eta, \beta$ or to intensify/diversify with a probability $1 - p_0$ according to the pheromone trail update information distribution and heuristic information distribution.

Researchers dynamically tune the parameter values in ACO. This is done to control intensification and diversification in ACO. This section discusses the parameter tuning in our modified ACO. These methods can be grouped into static, adaptive, and self-adaptive parameter tuning. Parameter tuning is a significant component to control the intensification and diversification in ACO. The experimentally concluded parameter values for ACO are; ant number m = problem size, α = 1, β = 8, ρ = 0.4, and Q= 25. The resulting graphical representation is as follows:

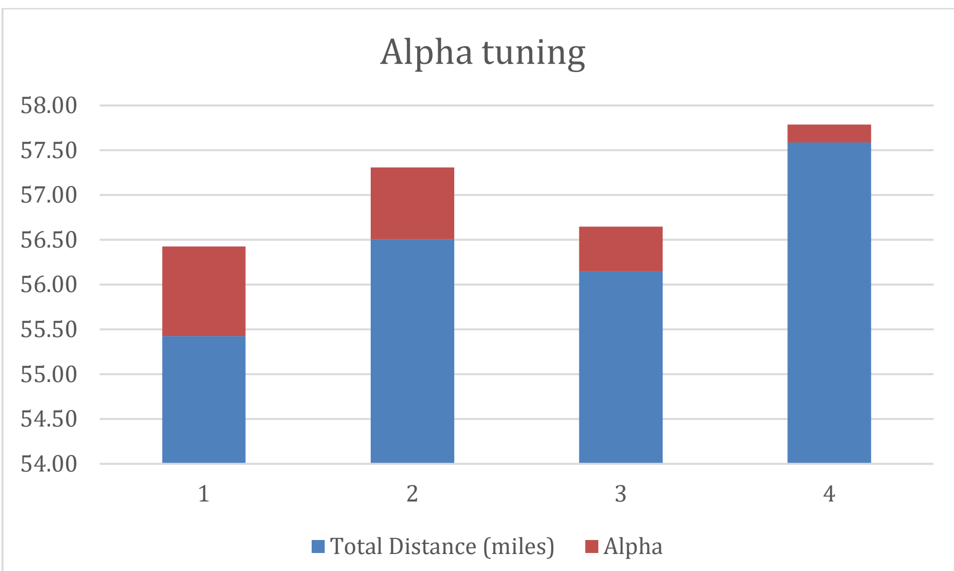

Figure 5: Relative importance of the alpha tuning for ACO

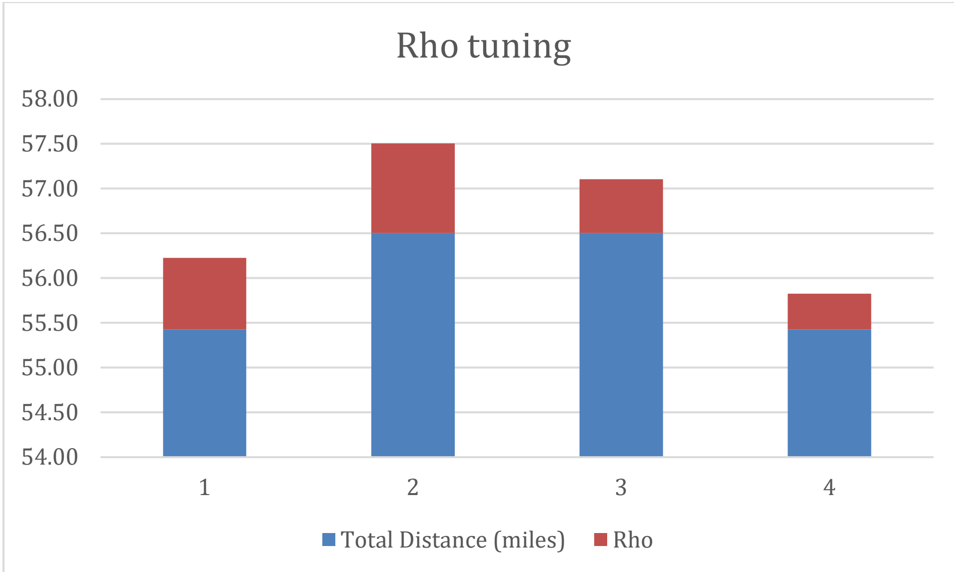

Figure 6: Evaporation rate tuning for ACO.

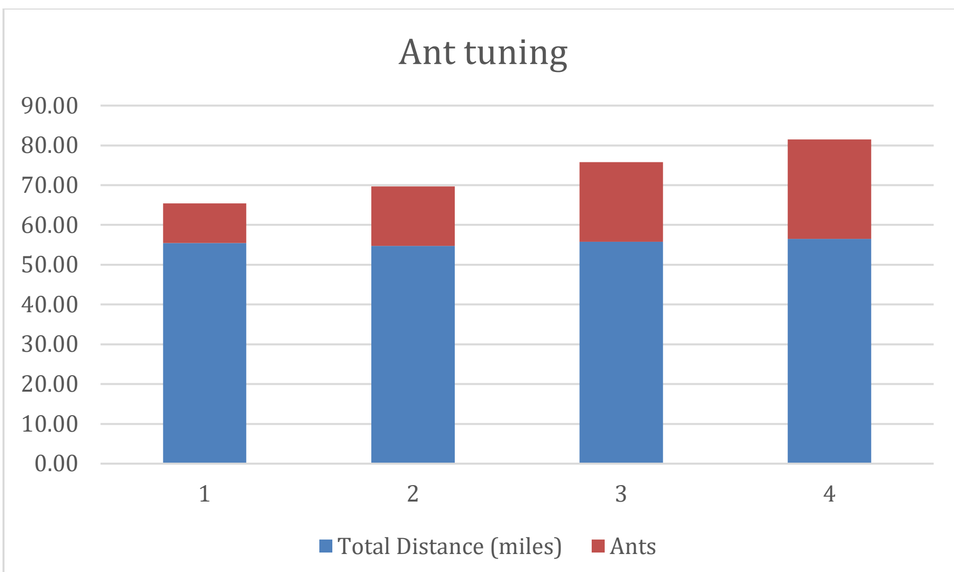

Figure 7: Number of ants tuning for ACO

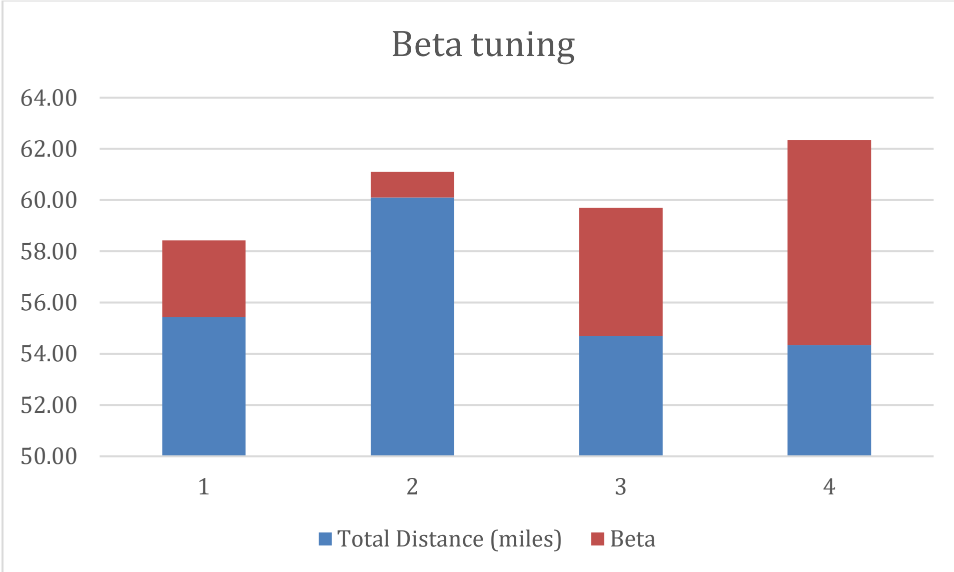

Figure 8: Relative importance of the beta tuning for ACO

5.2 Proposed agent based simulation-optimization model implementation

We developed our agent-based simulation by modelling our problem in real time GIS map. We modelled our distribution center, drop off locations, vehicle and orders as agents. The distribution center is located at Brussels, Belgium. We sourced our data from Belgium supply chain network for the distribution points as customers. From AnyLogic OSM server we were able to get real time road distances hence replicating real life scenario. The simulation is extended to include a constrained optimization experiment to calculate the ideal number of average vehicles necessary to maintain the demand and time windowed constraint of the model. The simulation-optimization model consists of multiple agents containing of a network of local customers, a single depot, a fleet of delivery vehicles, and orders. The model employs the integrated real-time GIS functionality to map the customer and depot agents in the model and automatically calculate the road network routes from the GIS provider. Orders are generated by the customers with fixed demand and time windows and are then received by the depot. These orders are modeled as agents and connect the distributing center and the customer. Post receipt of an order, vehicle assignment is scheduled with a routing logic embedded within the distribution center agent. Once the value is assigned to the route, the order is delivered within the time window to the requesting customer, serviced, and sent back to the hub.

Before developing the simulation model, the simulation goal and approach has to be defined based on the VRPTW. To create a hierarchical model, an object-oriented analysis method is adapted to calculate the roles and attributions of each agent. Post the object-oriented analysis, statistical analysis are essential for vehicle count, distance travelled, determination depot utilization and customer service level. Finally, a brief overview of agent development is demonstrated using the AnyLogic simulation software.

The delivery logic of the simulation model for the last mile VRPTW can be summarized in following steps:

(1) Development of the communicating agent network.

(2) Delivery process

(3) Simulation results

5.2.1 Development of the communicating agent network: As stated above, this is to calculate and initialize each essential attribution of the modeled objects. With initialization of customer points, distribution center, roads network and distribution vehicles. From AnyLogic features, objectified modeling method is adopted as mentioned before. The model structure is then divided into four groups according to the functions:

(1) Communication class

(2) Agent class

(3) Animation

(4) Dataset.

5.2.2 Delivery process: The delivery process is implemented by Order modelled as agents, which assign and control the movements of all vehicles. The AnyLogic employs the OSM server and shortest route logic to define the delivery process.

5.2.3 Model framework:

The agent based simulation model framework will be deployed in the following steps:

1. Set locations of the depot and the customers using the GIS function of AnyLogic server. The routes are automatically selected from the same GIS server.
2. Develop the process of demand orders and communicate the orders between the depot and customer agents.
3. Develop the depot's logic flow including order initiation and processing, vehicle assignment and return.
4. Create assimilation-optimization experiment to determine the optimal number of vehicles and minimum distance travelled.

5.2.4 Agent development:

Using the AnyLogic agent based modelling software, the agent structure and logic flow has been defined for our last mile routing problem in this subsection. The previously discussed four types of agents employed in our model are further explained in detail below:

a) Distributor/Deport agent: In the model, the depot agent is developed by considering the receipt of orders and assigning routes to vehicles issuing the OSM server functionality of AnyLogic. The process flow is shown in figure

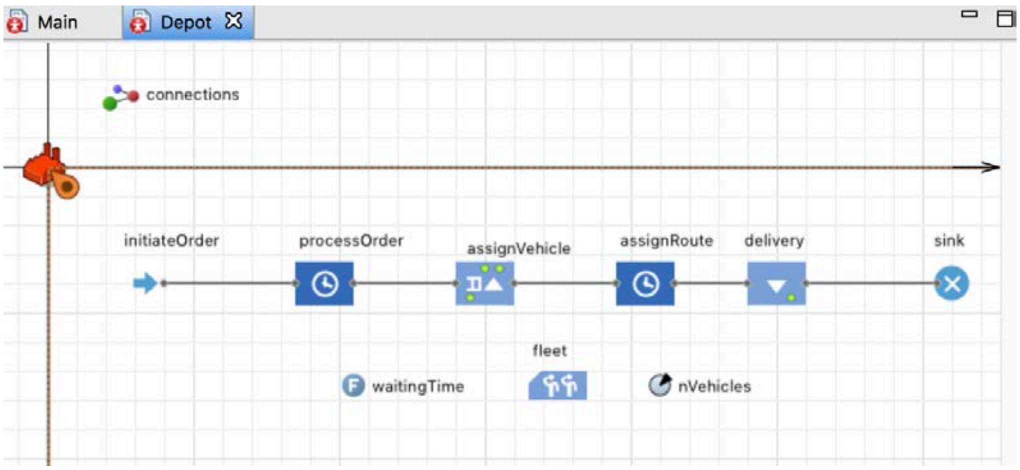

Figure 9: Process flow of depot agent

b) Order agent: In this agent, the important parameters like time windows and customer demand priority are assigned. The communication between the customers and depot is also established through the order agent.

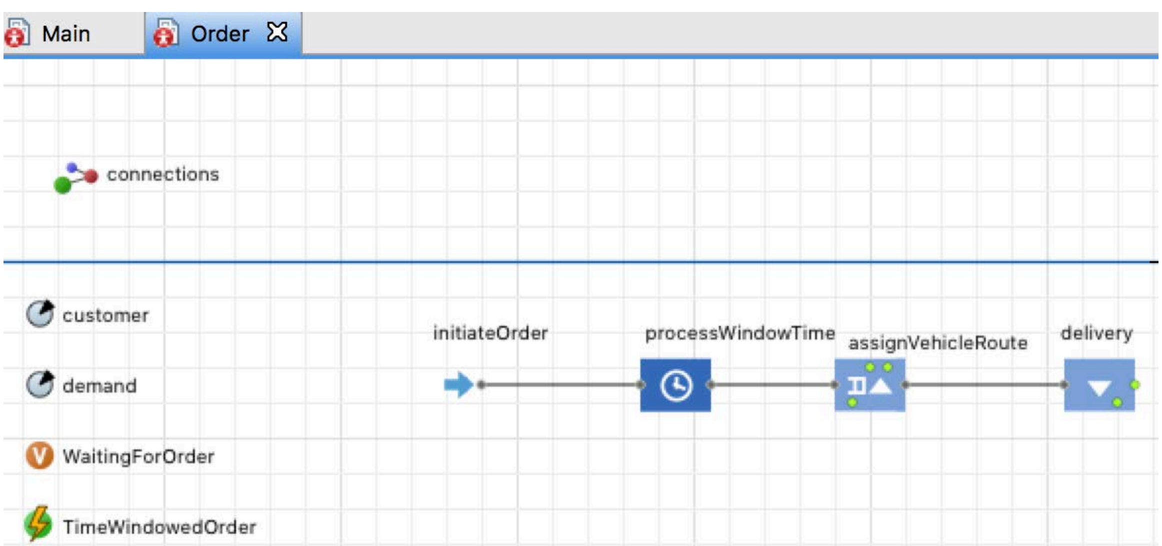

Figure 10: Process flow of order agent

c) Vehicle agent: The vehicle agent is the most basic agent in our model that receives routes and order details from the depot.

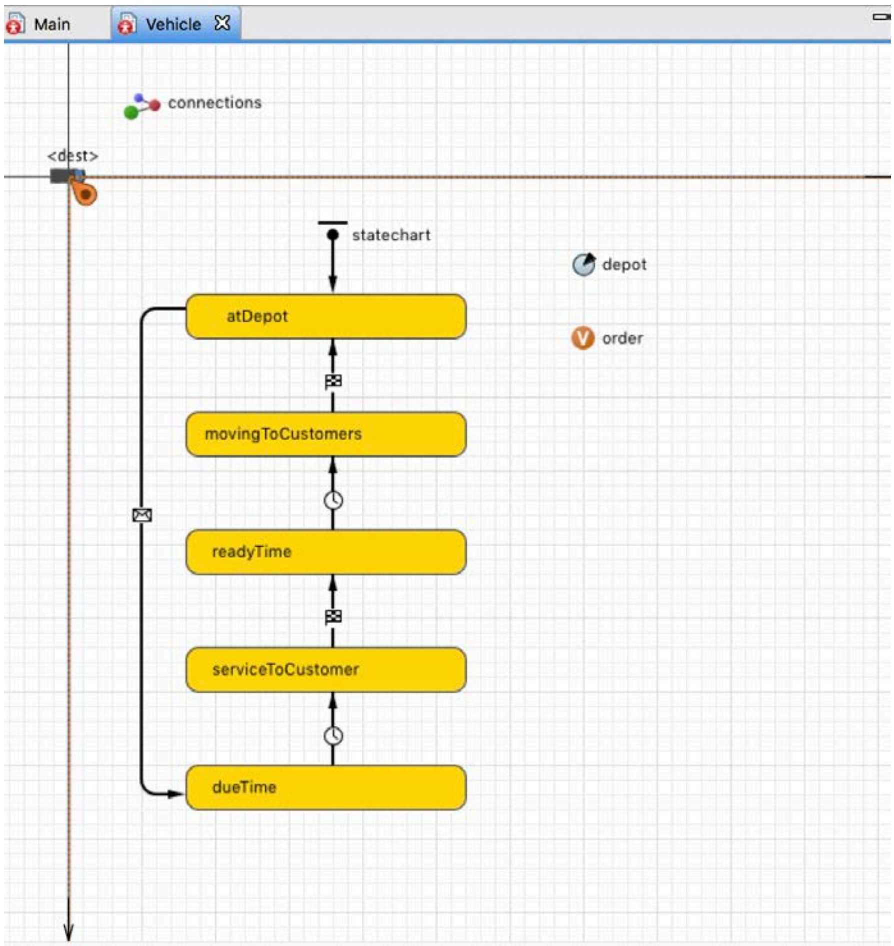

Figure 11: Process flow of vehicle agent

d) Customer agent: In the figure, the customer agent process-flow has been defined. The customer agent has been modeled to communicate to the depot through agents and assign constraints to the order per customer.

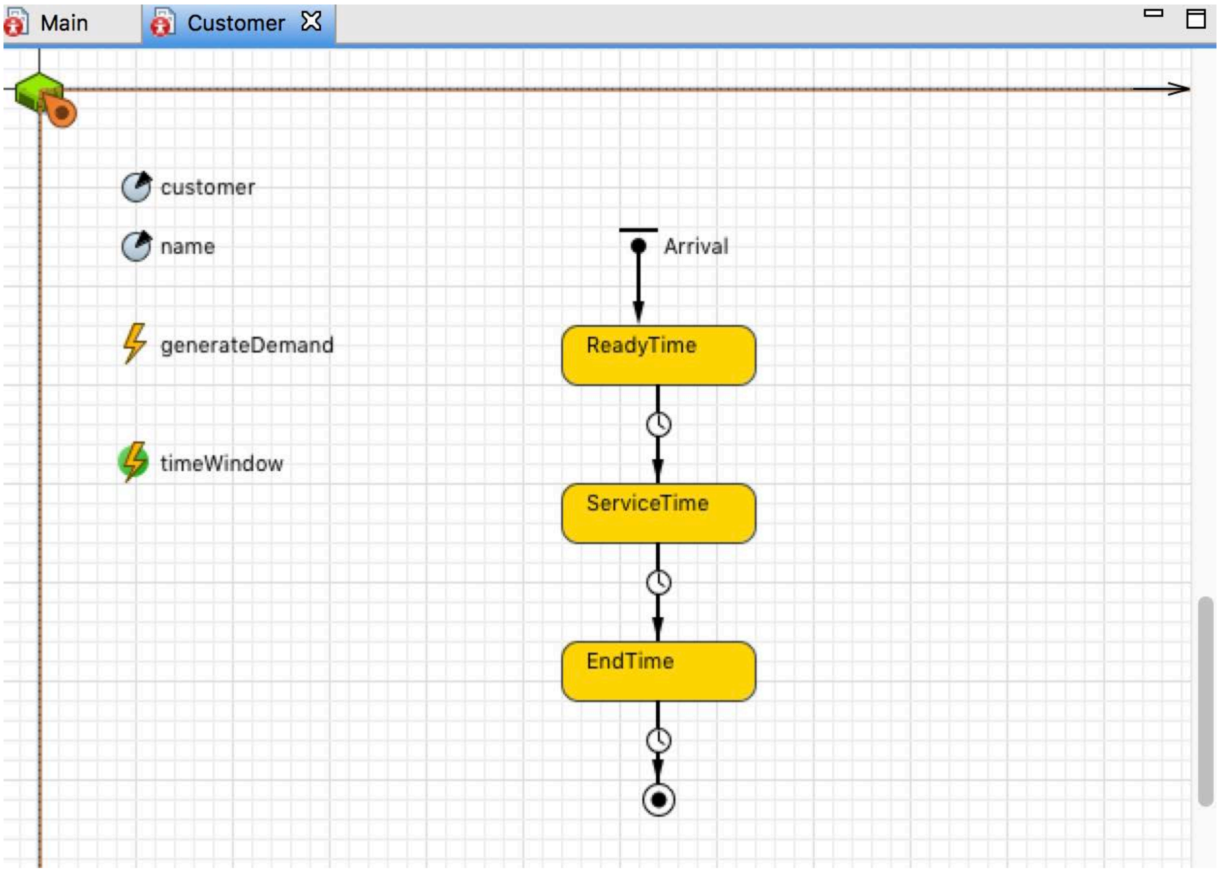

Figure 12: Process flow of customer agent

The simulation results: To simulate the last-mile logistics distribution process, the basic objective and optimizing functions are defined and the results are focused on two performance metrics:

(1) total traveled distance

(2) average number of vehicles

Data preprocessing

Preprocessing was also adopted to determine the eligibility of technicians to serve each customer to simplify the model and speed up computation. Since VRPTWs belong to NP-hard problems as proposed in Clarke (1964), it is tough to arrive at optimal solutions for large-scale problems using the model. In the approach, the Manhattan distance had to be converted into real time GIS distances and coordinates. Hence the distance callback by a function that calculates the distance between any two cities. The function developed in Python calculates the distance between any two points on earth from their latitudes and longitudes. A function that computes the distance between geographical coordinates is based on the Haversian function. The Haversine function defines the global distance between two coordinates given their longitudes and latitudes.

Chapter 6

Experimentation & analysis

In this chapter, data collected for the basic model is analyzed. The process of data collection was examined in the research methodology. Thereafter the collected data was further studied and subjected to experimentation, to explore the performance and relationships between important variables.

6.1 Experimentation

Experiment 1: Performance evaluation of proposed modified ant colony heuristic

Since we used the exact branch and bound algorithm to authenticate our results from the metaheuristic model, we performed a solution quality analysis and a computational complexity analysis to determine the higher performing algorithm.

Experiment 1: Dataset I description

For developing and optimizing our model, we sourced our data from the Daily Collegian newspaper agency. In our dataset, we have the location and address of the printing press as the depot in this case and the addresses of our drop off points (customer nodes). Since the dataset only provided us with actual addresses of the location, we used the Python IDE and the library 'geopy' for geocoding, which helped us in transforming an address of a location to the latitudinal and longitudinal coordinate location on the earth's surface. This enabled us to measure the distances in miles for both our algorithms. Further, we used the python library "gmplot' to successfully plot our coordinates and also provide us with the optimal route on the Google Maps.

Experiment 1: Results and analysis

The solution quality analysis was conducted by considering different sets of addresses (delivery nodes). This can be found in the figure below. Further, we conclude our work by determining the relative error for the metaheuristic we selected.

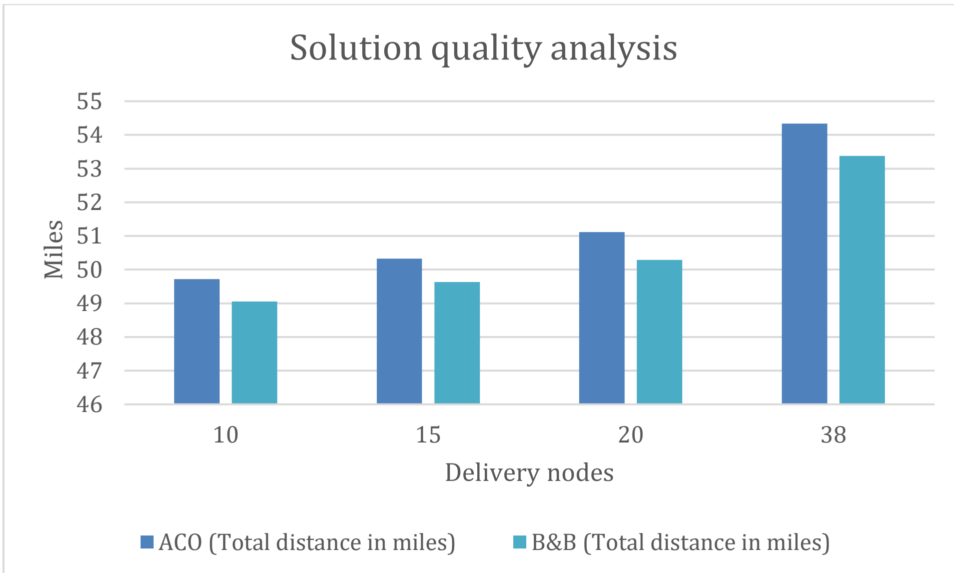

Figure 13: Solution quality analysis

The computational complexity of an algorithm is an important criteria to judge its performance. We determined the time complexity of both or algorithms for four cases of addresses or delivery nodes. This can be found in the figure 14 below.

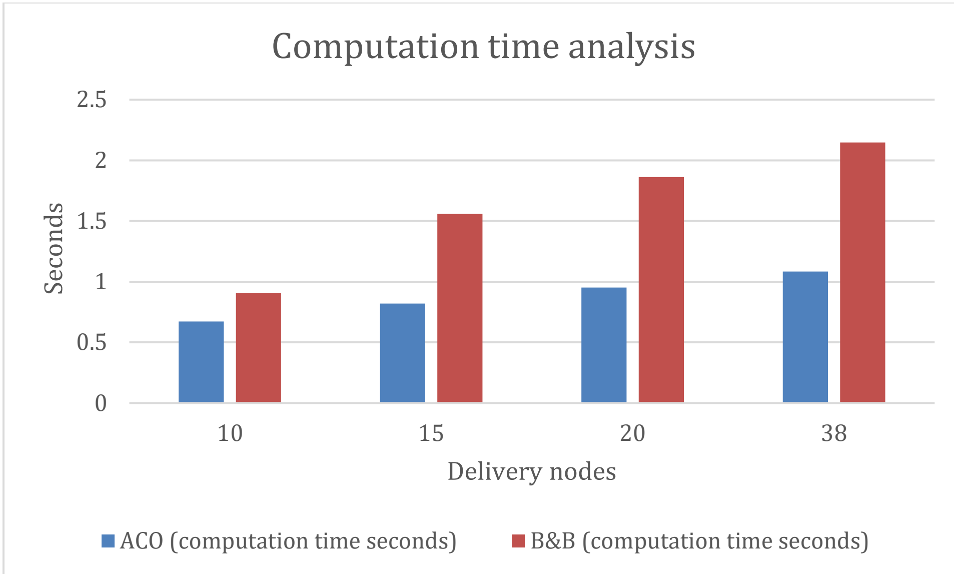

Figure 14: Computational time analysis

The final output from the coordinates are illustrated on the figure 15 below using the ‘gmplot’ python library.

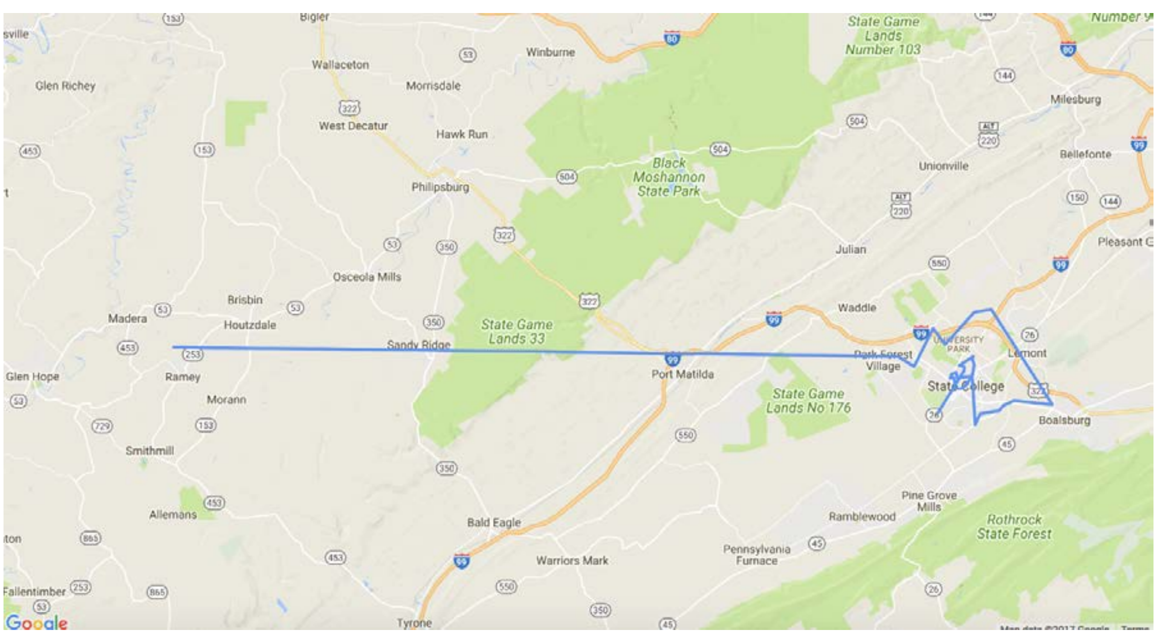

Figure 15: Optimal route selected through ‘gmplots’

Experiment 2: Benchmark evaluation

We then subjected our novel modified ant colony algorithm to a benchmarking test. The two most significant benchmarks used generally for evaluating the VRPTW algorithms are Homberger's benchmark and the Solomon's benchmark

Experiment 2: Dataset III description

We also selected few data points to test our algorithm; the two most utilized datasets were the Solomon's benchmark and Homberger's benchmark. The Solomon's benchmark contains 56 problem cases with 100 customers each. Two supplementary datasets are provided with 50 and 25 customers per case. These two supplementary datasets are used generally to calculate the performance index of the exact algorithms that cannot deal with the larger problem instances and hence are not used in the estimation of the approximate methods. The Homberger's benchmark dataset is an extension of the Solomon's data. In Homberger's benchmark, problems are created using the same configurations regarding the customer positions, demands and time windows. However, the problems instances described are of significantly larger size with 5 sets of 60 customers being provided with dimensions ranging from 200 to 1000 customers There are six example types provided — the R1, R2, RC1, RC2, C1, and C2 type. For each case, there is a slightly different arrangement with respect to customer location on the plot as well as the time windows properties. The network is considered being a comprehensive graph of the customer positions in a Euclidian space. For C1 and C2 types, the customer positions are arranged in clusters. For R1 and R2 classes, the customers are randomly positioned. The RC1 and RC2 case types then pool the previous two types with a combination of both random and clustered customer positions. The C1, R1 and RC1 differ from C2, R2 and RC2 in terms of the arranging horizon. The C1, R1 and RC1 cases feature a smaller horizon resulting in routes of around 10 customers on the average while the C2, R2 and RC2 problems have a larger horizon providing for routes of about 50 customers each.

Experiment 2: Results and analysis

Table 7-1. Benchmark tests of modified ACO in terms of cumulative travel time (sec).

Instance type	Solomon problems (100 customers)	Gehring & Homberger (200 Customers)	ACO (200 Customers)	Gehring & Homberger (400 Customers)	ACO (400 Customers)	Gehring & Homberger (600 Customers)	ACO (600 Customers)	Gehring &Homberger (800 Customers)	ACO (800 Customers)
C1	830	2836	2790	7813	7780	14914	14955	26786	26803
C2	592	1913	1954	3988	3775	7919	7928	12543	12568
R1	1203	3734	3688	8997	8948	20979	20963	34875	35014
R2	955	3072	3052	6618	6682	13623	13619	21770	21858
RC1	1357	3616	3628	8939	8915	18750	18775	38873	38922
RC2	1154	2681	2657	5594	5578	11627	11636	17827	17890

Experiment 3: Agent based simulation model and heuristic comparison

In the real world, vehicles in a last mile logistics network have to follow the roads; they cannot travel in a straight line from customer to customer. Most research papers and demos happily ignore this implementation detail. Although using road distances does not affect the NP-hard nature of a VRP much, it does result in a few extra challenges. The data set has ‘Google maps’ like roads with real distances in kilometers between every pair of locations in the dataset. It also contains the time windows and customer demand constraints. We then further developed an agent based simulation model for the last mile logistic dataset. The agent-based model was developed by the AnyLogic OSM server simulation system.

Experiment 3: Dataset II description

Since, public real life VRP datasets with road distances are rare in the operations research community we have sourced our dataset from De Smet (2014) VRP OR library. The VRP OR-library have few benchmarked datasets, but no real GIS data. By using the python library and Google Maps API, sourcing road distances was relatively simple and fast, as long as the entire road network of the data could be loaded into the modelling environment. The following steps were used to generate the dataset:

1. Google Maps API for generating with real distances in miles between every pair of cities or locations in the dataset.
2. Generate dataset in several orders of distance, time and magnitude, to relate scalability.
3. Add practical vehicle capacities, time windows and customer demands, for the constraints in VRPTW.

The procedures ended up generating datasets of Belgium from a Belgium supply chain network, we labelled this as Dataset II for our experimentation, with a location for major cities and from De Smet (2017).

Experiment 3: Results and analysis

The State College newspaper dataset was used for developing the initial simulation model to evaluate against the modified metaheuristic.

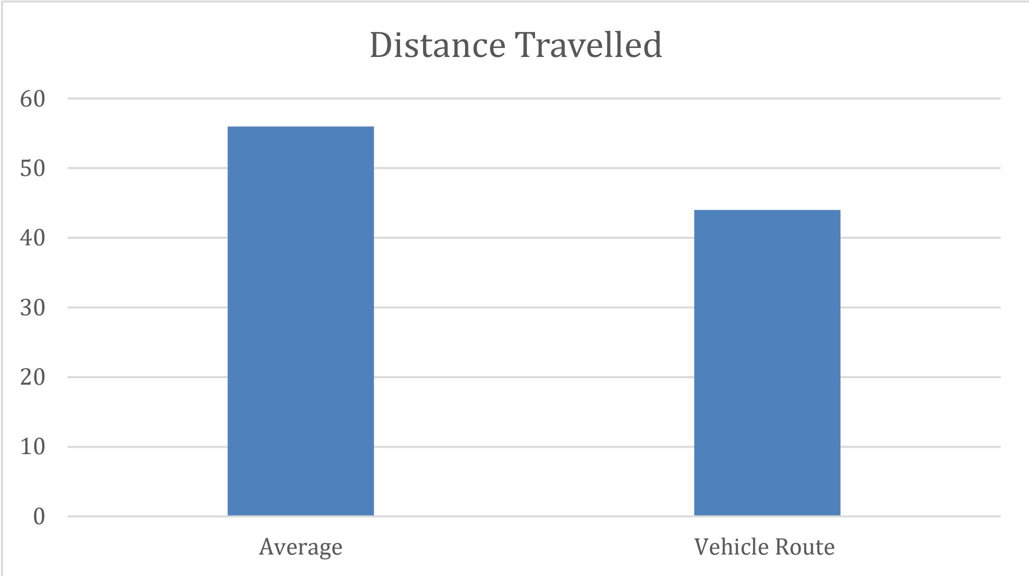

Figure 16: AnyLogic simulation results for distance travelled by vehicles for Dataset I

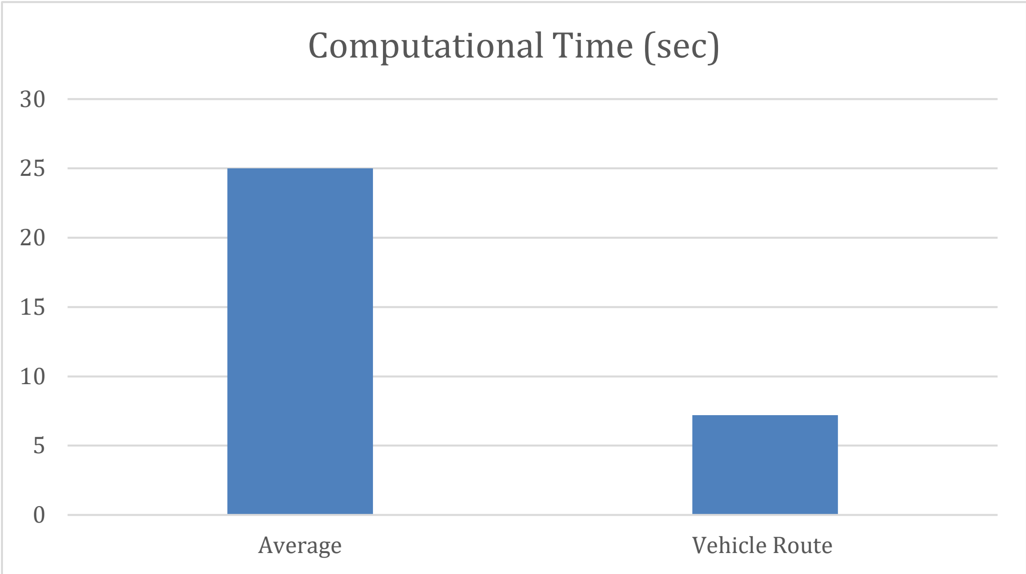

Figure 17: AnyLogic simulation results for computational time for Dataset I

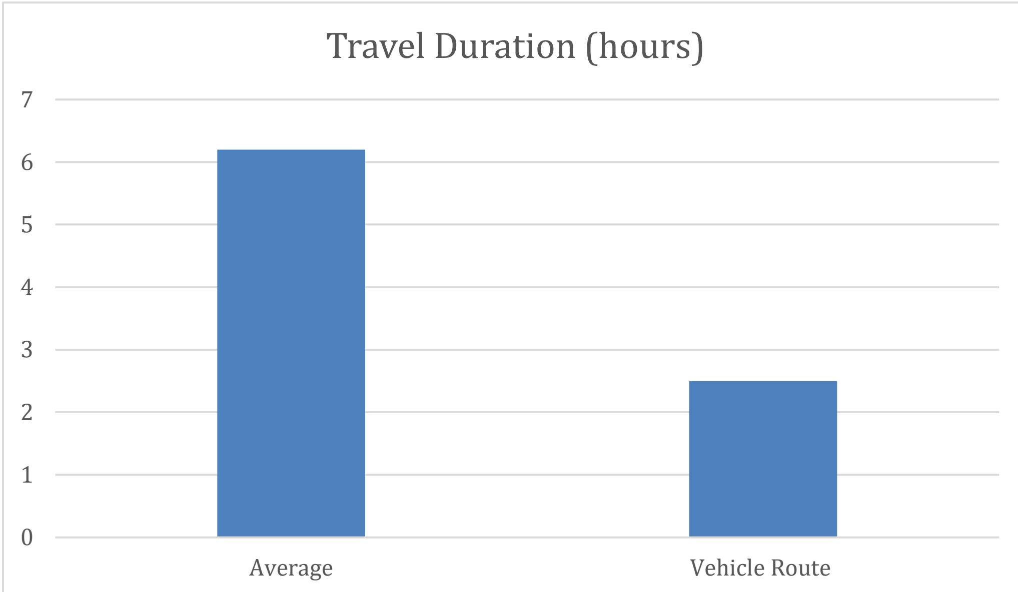

Figure 18: AnyLogic simulation results for travel duration by vehicles for Dataset I

We sourced our dataset from the VRP repository, which included datasets with road distances for Europe. The distribution center is located at Brussels, Belgium. We select 50 cities in Europe for the distribution points as customers. Apart from the 50 distribution points, there are 16 road junctions. We label these junctions for route expression number 1-50 present distribution points separately, while the number 51-66 are junctions. Therefore, we can represent the routes by two digit numbers and “-”. Moreover, the fact should be taken into consideration that in most cases the routes are linked by distribution points which presented by 1-50. However, a junction may exist between two customers. In this case, we need the real distances between every two-distribution points. The results of the agent-based simulation are as follows:

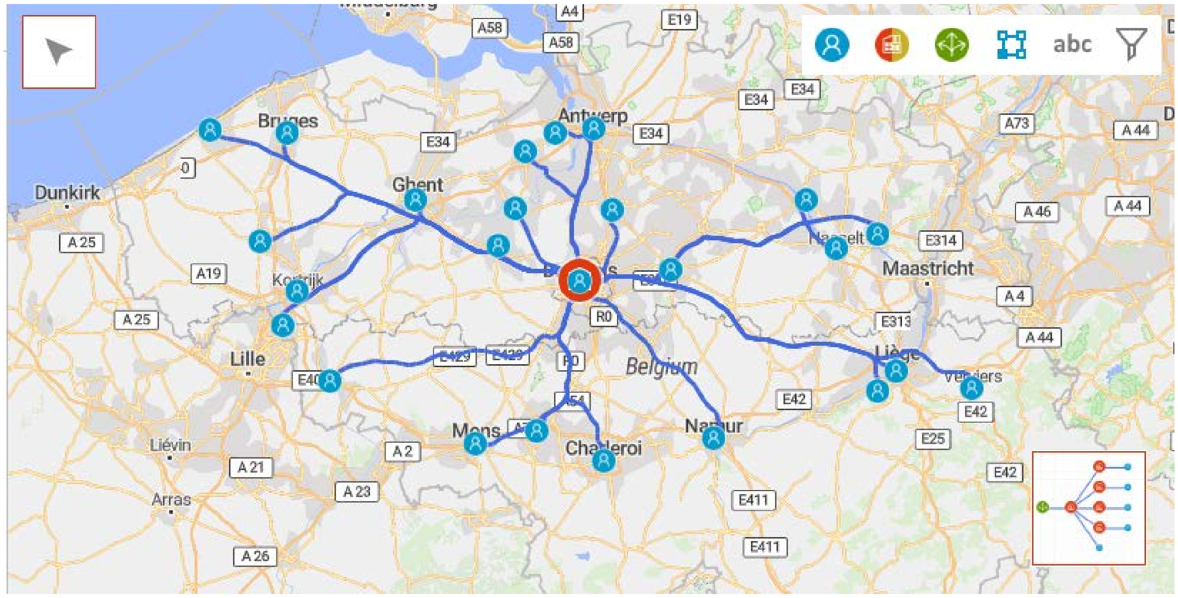

Figure 19: AnyLogic OSM model for 25 customer nodes in GIS space for Dataset II

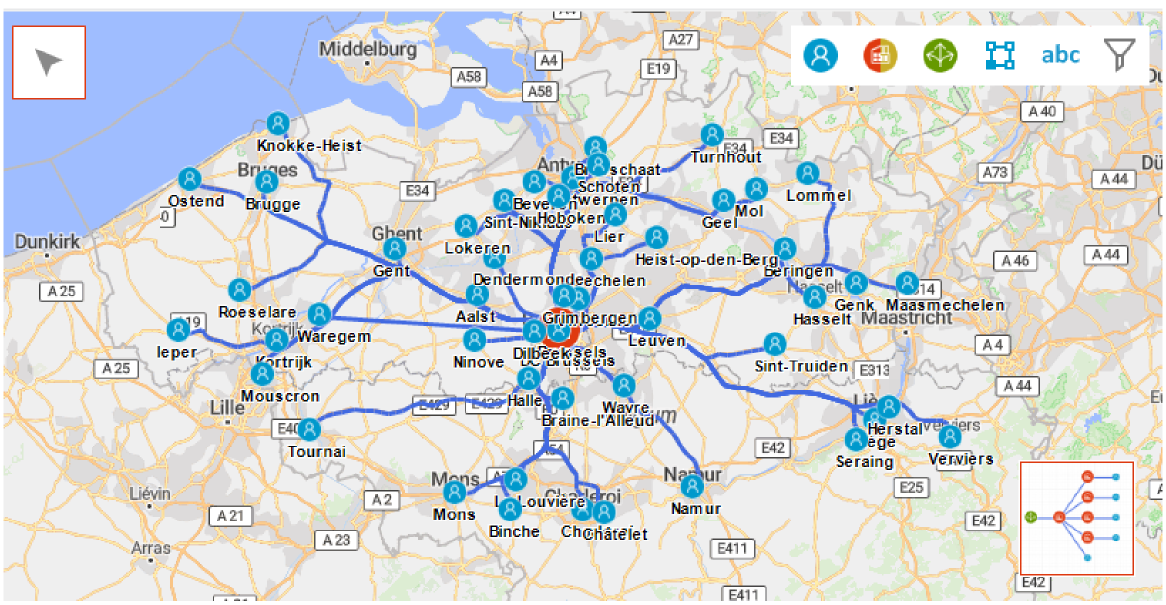

Figure 20: AnyLogic OSM model for 50 customer nodes in GIS space for Dataset II

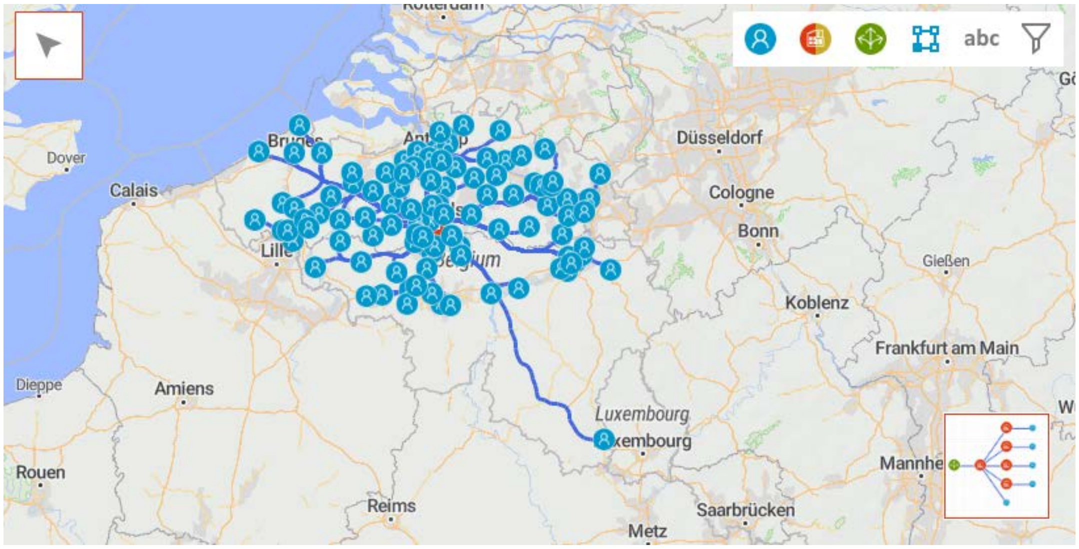

Figure 21: AnyLogic OSM model for 100 customer nodes in GIS space for Dataset II

6.2 Conclusions from experiments

From our model and experiment, we achieved a high level of accuracy from our meta-heuristic model. We modeled both our optimization models using the 'google optimization tool' and 'Pants' libraries in the python IDE. We also were able to source the coordinates from the real addresses present in our dataset. Post development of our models and analysis of our results we analyzed the quality of our heuristic. The results can be found for different cases of addresses /delivery nodes in terms of relative error in the table 7-2 below.

Table 7-2. Meta-heuristic relative error for Dataset II

ACO (Total distance in miles)	Exact method (Total distance in miles)	No. of delivery nodes	Relative error to objective value
2583.66	2727.68	50	5.28%
4782.95	5134.67	100	6.85%
7483.27	8137.52	250	8.04%
9425.15	10594.82	500	11.40%

We can see that the overall relative error is very low in case of the metaheuristic algorithm we used for our problem. However, in the results and discussion we can see that the time complexity obtained for the exact algorithm is higher due to the integral values it assumes. Hence, in conclusion we identify that both the optimization models are accurate in determining the least cost distance for our last mile logistic problem since there is a tradeoff between the two algorithms in terms of solution quality and computational complexity.

The agent based simulation model was analyzed and compared to the ant colony optimization metaheuristic. The cumulative number of vehicles and total distance were selected as performance parameters. The agent based simulation model was simulated for 1000 hours to obtain a steady state result. The distances are measured using the Google maps API and OSM server for GIS implementation. The results are as follows in Table 7-3.

Table 7-3. Solution quality comparison against agent based model for Dataset II

ACO (Total distance in miles)	ABM (Total distance in miles)	ACO (CVN)	ABM (CVN)	No. of delivery nodes
2583.66	2560.25	18	22	50
4782.95	4725.39	21	26	100
9425.15	9418.21	61	58	500

From the experimentation runs, both the proposed modified ant colony and the agent based simulation model performed well for up to 500 customer locations. However, for larger problem instances, the solution quality and computation time deteriorates as compared to the traditional exact algorithms.

Chapter 7

Conclusion

The main goal of this thesis was a fair and meaningful comparison of understanding the last mile logistics problem and search for different modified metaheuristics and agent based models applied to the problem. The last mile logistic problem was modelled as a Vehicle Routing Problem with Time Windows (VRPTW) to neatly replicate real life scenarios. The dataset considered for developing the algorithm was based out of GIS space. The performance of the algorithm and the Agent based model simulation as also compared to the solution quality of the exact algorithm and tested against standard VRPTW benchmarks and also through analyzing and researching weaknesses of the standard ant colony algorithm such as slow convergence speed, easy to fall into local optimal points.

With the advantages of the exact methods such as being intrinsic to the implicit parallelism and the ability of global optimization, the thesis proposed the optimization and modification of the ant colony algorithm to search for a global minimum. Developed a model of the vehicle scheduling system and implied the basic test functions of the model basing on the modified ant colony algorithm. Experimentation results showed that the optimization algorithm can get a better solution in a relatively short time, the algorithm has good stability, fast convergence speed, and can overcome the shortcomings of local optimization premature stagnation, demonstrates the advantages of the optimized algorithm for last mile logistic problem fully.

From this research, it becomes that generating a positive business for the last mile network is highly challenging. Based on their exhaustive reviews of existing last mile logistic literature, Min et al. [1998] and Nagy and Salhi [2007] identify a number of research gaps that require attention that is more detailed. Use of route length approximation formulae Nagy and Salhi [2007] explicitly state that, as routing plays a subservient role in LM logistics, approximation formulae for tour lengths and thus for the tour cost can often be used instead of explicit vehicle routing algorithms within the solution process of the last-mile model to speed up the process. Most of the available LM literature focuses on the development of purely deterministic models. For real-world applications, however, various sources of uncertainty, e.g. related to demand, cost, travel times and other crucial elements of the decision problem, have to be considered. It would thus be of high practical and academic relevance to consider stochasticity, largely with respect to last-mile models. Nagy and Salhi [2007] suggest employing robustness analysis to cater for stochastic (dynamic) LM's. Time windows and multiple optimization objectives are discussed only in very few publications on. Closely related to this is the observation that most existing last-mile models focus on cost as their sole optimization objective. In fact, they also have to ensure that they will meet certain customer requirements, such as on-time delivery, while providing their service at the least possible cost. Thus, the incorporation of multiple optimization objectives and time windows in particular would be a fruitful extension to existing literature.

Almost all last-mile logistics related articles consider static models. It would however offer interesting insights to develop dynamic last-mile models that account for multiple time steps and changing model parameters over time. While topics like closed-loop supply chains and reverse logistics are constantly gaining in relevance and public attention, the majority of the existing last mile logistics literature still considers inbound pickup operations and outbound delivery operations as two separate optimization problems. Thus, multi-stage LM models that consider inbound and outbound item flows simultaneously would offer a great research potential.

Application to real-world decision problems Min et al. [1998] highlight the importance of actually applying last-mile models and solution algorithms to real-world decision problems in order to broaden the spectrum of considered location-routing options but also to provide evidence of their efficacy and practicality. While last mile logistics models are generally a very well researched class of decision problems, the academic contributions to this stream of research are seldom applied and adapted to actual real-world problem settings. Closing the gap between theory and practice offers many appealing research opportunities.

Bibliography

1. Aized, T. & Srai, J. (2014). Hierarchical modelling of Last Mile logistic distribution system. *The International Journal of Advanced Manufacturing Technology*, 70, 1053-1061.
2. Allen J., Thorne G., Browne M. (2007). Good Practice Guide on Urban Freight Transport.
3. Anderson, W. P., Kanaroglou, P. S. & Miller, E. J. (1996). Urban form, energy and the environment: a review of issues, evidence and policy. *Urban Studies*, 33, pp.7-35.
4. Barbati, M., Bruno, G. and Genovese, A. (2012).Applications of agent-based models for optimization problems: A literature review. *Expert Systems with Applications*, Vol. 39 No. 5, pp. 6020-6028.
5. Barnard J.Lalonde (1997). Supply Chain Management: Myth or Reality?.
6. Billari F.C., Fent T., Prskawetz A., Scheffran J. (2006). Agent-Based Computational Modelling: An Introduction. Agent-Based Computational Modelling. *Contributions to Economics*.
7. Bruce L. Golden, Larry Levy. Rakesh Vohra (1987). The orienteering problem. *Naval Research Logistics (NRL)*. Volume 34, Issue 3 June 1987.
8. Bullnheimer, B., Hartl, R. F., & Strauss, C. (1998). Applying the ant system to the vehicle routing problem. *Meta-heuristics: Advances and trends in local search paradigms for optimization* (pp. 109-120).
9. Chen, C., S. Pan, Z. Wang, and R. Y. Zhong. (2016). Using Taxis to Collect Citywide E-commerce Reverse Flows: A Crowdsourcing Solution. *International Journal of Production Research.*

10. Cheng, C-B., & Wang, K-P. (2009). Solving a vehicle routing problem with time windows by a decomposition technique and a genetic algorithm. *Expert Systems with Applications*, 36(4), 7758–7763.
11. Clarke, G. and Wright, J.R. (1964).Scheduling of Vehicle Routing Problem from a Central Depot to a Number of Delivery Points. *Operations Research*, 12, 568-581.
12. Cleophas, C & Ehmke, JF (2014). When Are Deliveries Profitable?: Considering Order Value and Transport Capacity in Demand Fulfillment for Last-Mile Deliveries in Metropolitan Areas. *Business and Information Systems Engineering*. Vol 6, no. 3, pp. 153-163.
13. Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007). Evolutionary algorithms for solving multi-objective problems. *Genetic and evolutionary computation series.* Springer.
14. Cormen, T.H., C.E. Leiserson, R.L. Rivest, C. Stein. (2009). *Introduction to Algorithms*. MIT Press, USA.
15. Crainic, T. G., Ricciardi, N., & Storchi, G. (2009). Models for evaluating and planning City Logistics systems. *Transportation Science.*
16. De Smet (2014).Vehicle routing with real road distances. *OptaPlanner blog.*
17. Dolan, S. (2018, May 21). Crowdsourced delivery explained: making same day shipping cheaper through local couriers. *B.I. Intelligence*. Retrieved from http://www.businessinsider.com/crowdsourced-delivery-shipping-explained.
18. Donati, Alberto, Norman Casagrande, Andrea Rizzoli, and Luca Maria Gambardella (2008). Time Dependent Vehicle Routing Problem with a Multi Ant Colony. *European Journal of Operational Research*, Vol. 185, No. 3, pp. 1174-1191.

19. Dorigo, M. (1992*). Optimization, Learning and Natural Algorithms*. Ph.D. Thesis, Politecnico di Milano, Italian.

20. Ehmke, J. (2012). City Logistics in Integration of Information and Optimization for Routing in City Logistics. *Springer.*

21. El-Sherbeny, N. A. (2010). Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. *Journal of King Saud University (Science),* 22(3), 123–131.

22. G. Laperte, Y. Nobert and M. Desrechers, (1985). Optimal routing under capacity and distance restriction. 1050-1073.

23. G. Laporte and Y. Nobert, Exact algorithms for the vehicle routing problem.

24. Himelstein, L. (1999). Log on Boss. *Business Week*. EB26-EB27.

25. Holland, H., (1975). Adaptation in Natural and Artificial Systems. *The University of Michigan Press*, Ann Arbor, MI.

26. Homberger, J. and H. Gehring. (1999). Two Evolutionary Meta-Heuristics for the Vehicle Routing Problem with Time Windows. *INFORMS* 37, 297–318.

27. Homberger, J. and H. Gehring. (2005). A Two-Phase Hybrid Metaheuristic for the Vehicle Routing Problem with Time Windows. *European Journal of Operational Research* 162, 220–238.

28. Infocomm Development Authority of Singapore (2014). *Factsheet Smart Nation Platform.*

29. Ivanov D, Kopfer H, Haasis HD, Schönberger J (2011). Dynamics and sustainability in international logistics and supply chain management. *Proceedings of the 6th German-Russian logistics and SCM workshop* pp 41-50.

30. Jan A. Persson and Paul Davidssonn (2005). Integrated optimization and multi-agent technology for combined production and transportation planning. *In Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05).*

31. Jiang, L. & Mahmassani, H. S. (2014). City Logistics: Freight Distribution Management with Time-Dependent Travel Times and Disruptive Events. *Transportation Research Record: Journal of the Transportation Research Board*. 85 - 95.

32. Joerss, M. (2016). How customer demands are reshaping last-mile delivery. Retrieved from https://www.mckinsey.com/industries/travel-transport-and-logistics/our-insights/how-customer-demands-are-reshaping-last-mile-delivery.

33. Johann Dréo (2006).The ant colony optimization of the travelling salesman problem. [Photograph].

34. Johann Dréo (2006). Shortest path find by an ant colony. [Photograph].

35. K Małecki, S Iwan, K Kijewska (2016). Analysis of parcel lockers' efficiency as the last mile delivery solution–the results of the research in Poland. *Transportation Research.* Vol 12, 644-655.

36. Kohl, N., Madsen, O. (1997). An optimization algorithm for the vehicle routing problem with time windows based on Lagrangean relaxation. *Operations Research* 45 (3), 395–406.

37. Land A. H. and A. G. Doig. (1960). An Automatic Method of Solving Discrete Programming Problems. *Econometrica*, Vol. 28 No. 3, 1960, pp. 497–520. *JSTOR.*

38. Laudon, K. C., C.G. Traver. (2007). *E-commerce*. Pearson/Addison Wesley, New York.

39. Lindner, J. (2011). Last Mile Logistics Capability: a Multidimensional System Requirements Analysis for a General Modeling and Evaluation Approach. Retrieved from

http://www.pomsmeetings.org/confproceedings/025/FullPapers/FullPaper_files/025-0707.pdf.

40. London, T. F. (2012). Delivering a road freight legacy.
41. M´elanie Bourgeois, Gilbert Laporte, and Fr´ed´ericSemet. (2003). Heuristics for the black and white traveling salesman problem. *Comput. Oper. Res.*, 30(1):75–85, 2003.
42. Marc Pirlot. (1996). *European Journal of Operational Research* vol. 92, issue 3, 493-511.
43. McCrea, Bridget. (2016, June 6). From DC to Final Destination: Last Mile Dilemma.
44. Michael Hu and Sean T. Monahan "US E-Commerce Trends and the Impact on Logistics" *2016 AT Kearny.*
45. Min, H., Jayaraman, V., & Srivastara, R. (1998). Combined location-routing problem: A synthesis and future research directions. *European Journal of Operational Research*, 108(1), 1-15.
46. Mitrovic-Minic, S., Krishnamurti R, and Laporte G. (2004) Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transportation Research* B, 38, 669-685.
47. Morganti, E., Gonzalez-Feliu, J. (2014). City logistics for perishable products. The case of the Parma Food Hub. *Technical Report LET, Case Studies in Transport Policy.*
48. Morris, A. G. (2009). The Last Mile: Developing Efficient Freight Operations for Manhattan's Buildings. *Steven L. Newman Real Estate Institute*, Baruch College, CUNY.
49. M. Duran Toksari (2006). Ant colony optimization for finding the global minimum. *Applied Mathematics and Computation.* 176 (2006) 308–316.
50. Nagy, G., &Salhi, S. (2007). Location Routing: Issues, Models, and Methods. *European Journal of Operational Research*, 177, 649-672.

51. Niklas Kohl and Oli B. G. Madsen (1997). An Optimization Algorithm for the Vehicle Routing Problem with Time Windows Based on Lagrangian Relaxation. *Operations Research* 45:3 , 395-406.

52. Ombuki, B., B.J. Ross, F. Hanshar. (2006). Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence* 24(1): 17-30.

53. Osman. I (1993) Meta strategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Ann Oper Res* (1993) 41: 421.

54. Paloheimo, H, Lettenmeier, M &Waris, H (2016), Transport reduction by crowdsourced deliveries – a library case in Finland. *Journal of Cleaner Production*, vol 132, pp. 240-251.

55. Perspektiven der Wirtschaftspolitik, Volume 12, Issue 3, Pages 242–257.

56. Ploos Van Amstel, W. 2014. *Citylogistiek.*

57. Punakivi, M, Yrjölä, H &Holmström, J (2001). Solving the last mile issue: reception box or delivery box?. *International Journal of Physical Distribution & Logistics Management*, vol 31, no. 6, pp. 427-439.

58. Qiu, W., Yang, F. & Yang, D. (2005). City Logistics in China: An Overview. *Recent Advances in City Logistics*, Elsevier, New York, 417–427.

59. Roel Gevaers, Eddy Van de Voorde, Thierry Vanelslander (2014). Cost modelling and simulation of last-mile characteristics in an innovative B2C supply chain environment with implications on urban areas. *Procedia-Social and Behavioral Sciences.*

60. Roy, E. (2018, January 16). Who’s knocking at your door? Analyzing the future of last mile delivery. Trade ready, Retrieved from http://www.tradeready.ca/2018/topics/supply-chain-management/whos-knocking-door-analyzing-future-last-mile-delivery/.

61. Solomon, M.M. (1987). Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research* 35, 254–265.

62. Solomon, M.M. and J. Desrosiers. (1988). Time Window Constrained Routing and Scheduling Problems. *Transportation Science* 22, 1–13.

63. Sprenger, R., &Mönch, L. (2012). A methodology to solve large-scale cooperative transportation planning problems. *European Journal of Operational Research*, 223, 626–636.

64. Stanford University (2016). *One Hundred Year Study on Artificial Intelligence*.

65. Talukdar S.N., and Ramesh V.C. (1993). Cooperative Methods for Security Planning. *4th International Conference on Expert Systems Application to Power Systems*. Melbourne, Australia, Jan 4-8.

66. Talukdar S.N., and Souza P.S. (1992). Scale Efficient Organizations. *In Proceedings of the 1992 IEEE International Conference on Systems, Man, and Cybernetics*. Chicago, Illinois, Oct. 18-21.

67. Talukdar S.N., Pyo S.S., and Mehrotra R. (1983). Distributed Processors for Numerically Intense Problems. *Final Report for EPRI Project*. RP 1983-1764-3.

68. Talukdar, S.N., Baerentzen, L., Gove, A., and Souza, P. (1996). Cooperation Schemes for Autonomous Agents. *Times Assincronos para Problemas Industriais*, Sao Paulo.

69. Talukdar, S.N., Souza, P. de, and Murthy S. (1993). Organizations for Computer-Based Agents. *Engineering Intelligent Systems*, 1(2).

70. Taniguchi, E. & Thompson, M. (2002). Modeling City Logistics. *Journal of the Transportation Research Board*, 45 - 51.

71. Teodor Gabriel Crainic, Nicoletta Ricciardi and Giovanni Storchi. (2009). *Transportation Science Vol. 43*, No. 4 (2009), pp. 432-454.

72. The Logistics Institute Asia Pacific. (2013). Synchronized Last-Mile Logistics for Sustainable, Efficient Urban Delivery. *TLI - Asia Pacific White Papers Series.*

73. Gust Blauwens, Peter De Baere, Eddy Van De Voorde. *Transport economics.*

74. Wang X, Kopfer H .(2011). Increasing efficiency of freight carriers through collaborative transport planning: chances and challenges.

75. Wilson, R. (2007). Annual State of Logistics Report. *CSCMP.*

76. Zhou Q, LI Z. (2011). Evaluation of properties to determine the importance of individual roads for map generalization. *AdvCartogrGisci, Lecture Notes in Geoinformation and Cartography*, Part 51:459–475.

Appendix

Ant-colony pseudo code

```
import math
import argparse
from datetime import timedelta
from pants import World, Edge
from pants import Solver


ADD_COORDS_38 = [
[40.808499,-77.895752],
[40.803390,-77.883049],
[40.792134,-77.867813],
[40.800097,-77.867198],
[40.831305,-77.843047],
[40.823346,-77.874715],
[40.814783,-77.854642],
[40.808623,-77.855536],
[40.812196,-77.856102],
[40.807439,-77.856156],
[40.803221,-77.861796],
[40.803845,-77.865218],
[40.797601,-77.866382],
[40.793705,-77.868122],
[40.792065,-77.863437],
[40.793436,-77.860714],
[40.793360,-77.859769],
[40.798125,-77.855862],
[40.798464,-77.855829],
[40.794870,-77.860170],
[40.794759,-77.860740],
[40.793323,-77.862671],
[40.779583,-77.879468],
```

```
[40.773129,-77.856234],
[40.778888,-77.853733],
[40.780296,-77.850761],
[40.783378,-77.837776],
[40.783824,-77.827286],
[40.785654,-77.834790],
[40.798522,-77.860419],
[40.804293,-77.854620],
[40.801510,-77.861751],
[40.799695,-77.869574],
[40.798525,-77.870421],
[40.797015,-77.870886],
[40.797324,-77.868361],
[40.813198,-77.906811],
[40.832941,-77.800758],
]


def distance(x, y):
    return math.sqrt((x[1] - y[1]) ** 2 + (x[0] - y[0]) ** 2)


defrun_args(dnodes, *args, **kwargs):
    world = World(dnodes, distance)
    solver = Solver(**kwargs)


solver_format = "\n".join([
        "solver:",
        "lim={w.limit}",
        "r={w.r}, Q={w.q}",
        "a={w.a}, b={w.b}",
        "e={w.e}"
        ])
```

```
    print(solver_format.format(w=solver))


    col = "{!s:<26}\t{:<26}"
    div = "-" * (26 + 26)
    head = col.format("Time taken", "Distance covered")
    col = col.replace('<', '>', 1)


    print()
    print(head)
    print(div)


    fast = None
start_t= time.time()
    for x, ant in enumerate(solver.solutions(world)):
        fast= ant
fast_time = timedelta(seconds=(time.time() - start_t))
        print(col.format(fast_time, ant.distance))
total_t = timedelta(seconds=(time.time() - start_t))


    print(div)
    print("best sol:")
    for x, n in zip(fast.visited, fast.tour):
        print("  {:>9} = {}".format(x, n))


    print("Sol value: {}".format(fast.distance))
print("found at  {} out of {} sec.".format(fast_time, total_t))
```

```
if __name__ == '__main__':
    epilog = "\n".join([
        '  * 0.5 <= a <= 1',
        ])


dnodes = {
        38: ADD_COORDS_38
    }[args.dataset]
```