DISTRIBUTED CONTROL OF A PARCEL DELIVERY SYSTEM THROUGH CROWDSOURCING

A Thesis in
Industrial Engineering
by
Adithya Nadig

© 2017 Adithya Nadig

Submitted in Partial Fulfillment of the Requirements for the Degree of

Master of Science

August 2017
The thesis of Adithya Nadig was reviewed and approved* by the following:

Vittaladas V. Prabhu  
Professor of Industrial and Manufacturing Engineering  
Thesis Advisor

Soundar Kumara  
Allen E. Pearce and Allen M. Pearce Professor of Industrial Engineering

Janis Terpenny  
Peter and Angela Dal Pezzo Department Head and Professor  
Head of the Department of Industrial Engineering

*Signatures are on file in the Graduate School
ABSTRACT

Last Mile Delivery systems over the years have experienced improvements that have been brought about by technology and supporting infrastructure. With the introduction of regional warehouses and last-mile distribution networks backed by routing algorithms and optimization models, businesses have been able to harness better profits and operate at a higher efficiency. Now, with Internet of Things (IoT) and crowdsourcing, there are new opportunities of improving delivery systems.

This thesis proposes a distributed system for Crowdsourced Parcel Delivery and subsequently presents a Traveling Salesman Problem (TSP) for delivering parcels to their destinations. The system utilizes Driver Reliability, Auctioning, Utility Function, and Binary Bin Packing (considering Vehicle Capacity) to reduce the search space for each sub-problem. The TSP formulated is then solved using either a Genetic Algorithm (GA) or Distributed Control Algorithm in order to determine the best sequence of routes that must be taken.

Computational Experiments show that as the number of parcels and the number of drivers increases, the problem becomes too complex to be solved in real-time using the GA. The DATC is used to obtain quicker solution with trade-offs.

With 100 drivers, 500 parcels and a condition for all parcels delivery, DATC improves the computation time by up to 18% when compared to GA for low capacity examples. However, there exists a non-linear trade-off between distance traveled and computation time between the two methods. The entire solution computation is done off-line at a higher speed of simulation so that a good solution may be obtained in real-time.
# TABLE OF CONTENTS

List of Figures .................................................................................................................. v

List of Tables ..................................................................................................................... vi

Acknowledgements .......................................................................................................... vii

Chapter 1: Introduction ...................................................................................................... 1

1.1 Overview ..................................................................................................................... 1

1.2 Problem Description and Discussion ........................................................................ 2

Chapter 2: Literature Review ............................................................................................. 5

2.1 Last Mile Delivery Systems ....................................................................................... 5

2.2 Auctions ...................................................................................................................... 8

2.3 Utility Function .......................................................................................................... 10

2.4 Algorithms for Traveling Salesman Problem ........................................................... 13

2.4.1 Distributed Control Algorithm ............................................................................. 14

2.4.2 Genetic Algorithm and Ant Colony Optimization ................................................ 17

Chapter 3: Data Description .............................................................................................. 20

Chapter 4: Methodology .................................................................................................... 24

4.1 Parcel and Driver Information ................................................................................. 24

4.2 Route assessment ...................................................................................................... 26

4.3 Auctioning for parcel assignment and bin packing ................................................... 26

4.4 Routing ..................................................................................................................... 29

4.4.1 Genetic Algorithm ............................................................................................... 30

4.1.2 Distributed Algorithm ......................................................................................... 31

Chapter 5: Experimental Results ...................................................................................... 34

5.1 Parameter Variations ............................................................................................... 34

5.2 Limiting Parameters ............................................................................................... 43

5.3 Sequencing using PID ............................................................................................. 45

5.4 Solution Quality ....................................................................................................... 47

5.4.1 Computation Time ............................................................................................... 47

5.4.2 Total Distance Traveled ..................................................................................... 48

5.4.3 Due Time Variation ............................................................................................. 49

Chapter 6: Conclusions and Future Work ....................................................................... 51

6.1 Limitations of the System ....................................................................................... 51

6.2 Future Scope ............................................................................................................ 52

References ....................................................................................................................... 54
LIST OF FIGURES

Figure 1: Difference between traditional outsourcing and crowdsourcing [10]. ..............5

Figure 2: Feedback loop of the DATC[36] .................................................................14

Figure 3: The 3 regions of that an arrival time can fall in .....................16

Figure 4: Flowchart of the Genetic Algorithm [37] ........................................19

Figure 5: Distribution of destinations .................................................................21

Figure 6: Eligibility criteria for parcel assignment ........................................27

Figure 7: Threshold criteria for using traditional delivery ............................28

Figure 8: Best routing sequence (left) and sequence of delivery in different genes (right) for configuration 1 .................................................................35

Figure 9: Lack of assigned parcels in Configuration 4 .......................................39

Figure 10: Best routing sequence (left) and sequence of delivery in different genes (right) for configuration 1 .................................................................39

Figure 11: Comparison between fulfilled and unserved destinations ................40

Figure 12: Parcel assignment limited by driver capacity ..................................42

Figure 13: Case wherein availability of parcels and vehicle capacity is high in a particular route .................................................................42

Figure 14: Final traversed route at 500 generations .......................................44

Figure 15: Final traversed route at 1000 generations ....................................45

Figure 16: Completion time using the DATC for parcel 6 in configuration 1 example. ..........46

Figure 17: Completion time graph from Figure 17 captured separately for odd and even iterations .................................................................46

Figure 18: Comparison between computation times as the number of deliveries increases ....48

Figure 19: Comparison of solution quality of GA and DATC ................................50
LIST OF TABLES

Table 1: Example of driver tuple in the database.................................................................25
Table 2: Example of Parcel tuple in the database.................................................................25
Table 3: Propagation of the Genetic Algorithm.......................................................................30
Table 4: Parameters that affect the final solution in GA........................................................31
Table 5: Bounds of the parameters in the system .................................................................34
Table 6: Various system configurations based on the values of the parameters.....................34
Table 7: Alternating solutions between iterations in DATC..................................................36
Table 8: Parcel sequence variation in DATC and GA............................................................37
Table 9: Solution quality comparison between the two methods. ........................................37
Table 10: Limiting parameters for different configurations. ................................................43
Table 11: Comparison of computation times for obtaining best solutions. ........................48
Table 12: Travel Distance for Configuration 8 example. ......................................................49
I wish to express my heartfelt gratitude to Dr. Vittaldas Prabhu without whom this work would not be possible. Through his consistent patience and unwavering support, Dr. Prabhu nurtured my research curiosity and provided a sense of direction to my work. While encouraging my ideas, he lent me a platform for intellectual and academic development through his insightful lectures and engaging debates.

I also wish to convey my indebtedness to Dr. Soundar Kumara for piquing my academic interest through his valuable suggestions and constant encouragement. His classes and intriguing discussions were a continuous source of motivation.

I’m deeply grateful to my colleagues and peers – Sandeep Shastry, Shyam Vasudevan, Shamir Jose and Naresh Natesan, whose work in their fields resulted in fruitful exchange of information that helped make my work better.

To my parents, for schooling the importance of discipline, responsibility and hard-work in life, and to Nischala for being my tower of strength – thank you.
Chapter 1: Introduction

1.1 Overview

The last-mile of a delivery network corresponds to the final movement of goods or services, usually from a distribution center, to the final destination. From the purview of last-mile parcel delivery, we can broadly classify the systems in existence into two categories: In-house logistics and Third-party logistics (TPL). In this thesis, the parcels are considered already in the hands of logistics providers such as UPS, FedEx, DHL etc. These companies have a choice to utilize their own fleet of trucks to perform the final delivery (In-house) or to approach a third party (TPL).

Crowdsourced Delivery of parcels has been a field that has attracted a lot of attention in the recent past. This is primarily because of the increased cost incurred by logistics companies in the last mile. It is estimated that the last mile of a supply chain may amount up to 28% of the total transportation expenses [1]. In literature, crowdsourcing has made a case for itself in delivery or, in general, last mile transportation. Though the concept in itself has been in existence, the entry of crowdsourcing as a business model is new and challenging. This is partly because of the conceptualization involved [2] and also because of the reluctance of the existing traditional last mile carriers to delve into an untested innovation [3].
Crowdsourcing as a means of parcel delivery has attracted investment from various startup companies as well as ventures from companies like DHL (MyWays Platform in Stockholm). A recent article reported Wal-Mart’s interest in considering its customers as a means to construct delivery networks for purchases made on its website [4].

A classic crowdsourcing example for transportation is Uber. However, in parcel delivery, unlike Uber, the call for a pick-up is initiated by a single entity and is usually for more than one parcel. Moreover, there is a single pickup location and multiple drop-off locations. These destinations are broken down various routes and participating vehicles are made to choose a preferred route at the beginning. Thus, the system utilizes the choice of the drivers as a means of reducing the search space.

1.2 Problem Description and Discussion

Various heuristics [5] have been developed that can assess a routing problem and provide good solutions. Naturally, since the calculations are computerized, computation time, cost and the closeness of the solution to the actual optimum value become obvious parameters of comparing methods.

While simple case studies have utilized routing heuristics from literature and provided good results, scalability has been an issue for these heuristics [6]. As the problem size increases, the computation time or the deviation from the optimum value increases. From the point of view of vehicle routing, providing answers in real-time becomes harder
as the number of variables in the system increases [7]. The increase in variables creates a stress-point within the system where a bulk of the system computation time is localized. Complexity in the system is primarily a result of these stress-points within the system [8]. One way of coping with this issue is to reduce the amount of information fed into the system during computation and making the various entities in the system highly autonomous [9].

Information regarding a solution, say, with a starting point other than the pickup location, for instance, may be considered as global information in the system. It may also be the information regarding the solution for drivers that are not in the system at that time instance. So, the solution obtained for that driver would be unnecessary waste of the system’s time.

For a Parcel Delivery System, which is traditionally hierarchical in nature, a distributed model is proposed as shown in figure 1. In this model, a crowdsourced parcel delivery service is introduced. The model is tested for a hypothetical small area with multiple destinations and routes. The two methods utilized are then compared for travel distance, delivery time deviation, scalability, and computation time. The resulting distributed nature of the system ensures quick solutions are calculated at discrete time intervals. Since the solutions are obtained quickly, the parameters may be altered to suit the requirement and simulations may be conducted more than once. This can provide the decision-makers with alternate solutions which might not have been possible if the system had had to go through the entire solution space for all the sub-problems.
The architecture may be termed as distributed due to the overall decomposition of the problems in the system. The parcels in this system act as autonomous entities that are served by drivers. The drivers, in turn, decide the set of parcels to serve depending on implicit conditions. The parts are then assigned to drivers based on FCFS rule and capacities of each vehicle. The parcels and the drivers are provided with minimum global information and do not have a centralized controller. Thus, a TSP for a particular driver does not depend on the information from other TSPs.

This thesis is organized as follows. Chapter 2 deals with background literature. A detailed description of the data and its lack thereof is provided in Chapter 3. Data was assumed for an ideal traditional parcel delivery case. All the hypothetical data obtained is distributed uniformly. Different assumptions are discussed and reasons for the same are provided. Chapter 4 discusses the methodology and the general flow of the problem. The utilization of First price sealed- bid auctions, Genetic Algorithm for vehicle routing and the distributed arrival time controller are explained. An explanation of the utility function and the reasons for its implicit use in the system is provided. Computational Results and the impact of changes in parameters on both the algorithms are discussed in Chapters 5 and 6.
Chapter 2: Literature Review

2.1 Last Mile Delivery Systems

TPL at the last-mile network can be further classified into: 1. Outsourcing or 2. Crowdsourcing. Crowdsourcing is a fairly new means of achieving service wherein a network of people, through the internet, are approached to perform a series of tasks. Schenk and Guittard [10] argue that the benefits of crowdsourcing extend to cost savings as a result of micro-payments rather than lump-sum payments.

The major difference between traditional outsourcing and crowdsourcing, as shown in figure 1, is the means of approach of the organization to the logistics provider. While the suppliers are predetermined in outsourcing, the suppliers, or the crowd, respond to an open tender in the case of crowdsourcing.

![Figure 1: Difference between traditional outsourcing and crowdsourcing](image)

Figure 1: Difference between traditional outsourcing and crowdsourcing [10].
Crowdsourcing, in most cases, consists of 3 entities, namely: 1. Providers, 2. Consumers, and 3. Crowdsourcing enablers (Drivers). While the authors discuss various types of crowdsourcing in practice (not just restricted to delivery, but also app development for smartphones among others), they assume the investment in the system purely from the standpoint of a start-up company and not for an already established logistics provider in the parcel delivery industry. The paper juxtaposes the idea of crowdsourcing with other contemporary concepts such as Open Innovation (a two-way process of obtaining as well as providing service) and User Innovation (Beta innovation – where the consumers provide a service to satisfy their needs).

This paper discusses 1. the impact of the general concept of crowdsourcing and makes an argument that the cost is saved overall as a result of micro-payment as opposed to lump-sum payment as in the case of traditional outsourcing 2. Increase in solution quality as a result of volume of beta testers 3. Impacts on risk as a result of lesser dependence on one provider and increase in risk as a result of dependence on the crowdsourcing platform instead.

While the authors argue the pros and cons of crowdsourcing, there is a dearth of mathematical or physical evidence in the paper. An actual example, however, is due to a lack of a real situation wherein both the methods were tested.

Rouges and Montreuil [11] examine crowdsourcing delivery in particular and perform a critical examination of the return for the stakeholders of an enterprise that is venturing in crowdsourcing for delivery. For this purpose, 18 companies that use crowdsourcing platform were chosen. The public released documents including videos,
webpages, and blogs were utilized to collect data. It presents 3 models viz., 1. B2C model (Business to Customer) 2. B2B model (Business to Business) 3. P2P model (Peer to Peer). Among the start-up samples considered, B2C is dominant and are shown to be successful in densely populated urban cities.

The paper examines the benefits of crowdsourced delivery to customers, businesses and the society. While there are a number of advantages of a crowdsourcing system, the paper does not critically examine the system by considering alternatives. A comparison between a traditional system and crowdsourcing is absent. Additionally, the paper elaborates on the qualitative obstacles (trust between the sender and the driver) and quantitative obstacles (critical mass – minimum participation that would make the venture break-even). The paper provides a platform for future research to obtain reliable answers to the claims and means to overcome the obstacles presented.

Vukovic, Das and Kumara [12] examine systems that utilize crowdsourcing to solve problems. It not only visualizes crowdsourcing as a means to achieve delivery but also as sensors and solvers. As sensors, the authors classify crowdsourced sensing into automatic (for example, driving speed, distances, electricity consumption etc.) and explicit (preferences, opinions, and answers to specific questions). Obtaining crowdsourced data of possible solutions for an existing problem, on the other hand, is a field that has gained attention over the last several years. Utilizing crowd-mapping through games and other interactive methods to get information about crowd density is prevalent as is observed in Google’s local guides and Niantic Labs’ PokemonGO [13].
Mashhadi and Capra [14] suggest that a participant’s mobility pattern and reputation history for the service provided be utilized as a measure of information quality. A credibility weight for a user \((u_i)\) may be determined using:

\[
CW(T_i) = \alpha. Reg(T_i) + (1 - \alpha). Trust(u_i)
\]  

(1)

Where,

- \(T_j\) = Set of spatio-temporal Tuples for user \(j\)
- \(CW\) = Credibility Weight
- \(\alpha\) = weight to adjust between the 2 parameters.
- \(Reg\) = Implicit Location Readings (Mobility)
- \(Trust\) = Reputation score

The implementation as discussed by the authors is through an Android application which would be designed to be competitive among participants. It would also receive inputs from the participants that would be fed into the tuples to calculate the credibility.

### 2.2 Auctions

From the purview of a crowdsourced delivery system, auctions may be utilized as a means of choosing from a set of drivers. For a particular parcel that is to be delivered in a particular route, an auction may be called to obtain a driver or a set of drivers who deliver a parcel. In the latter case, heuristic within the system may be used to rank the drivers in the set.

Luking-Reiley [17] identifies in his paper the prevalence of Vickrey Auctions – An auction wherein the winning bidder pays the price named in the second-best bid – specially in Stamp auctions. However, the author shows literature evidence as to why its use in physical auctions is restricted – due to the fear of cheating by the auctioneers to increase the second price just below the highest bid. The paper, however, does not provide information about the reaction of the bidders when they know that the auction is genuine. The Vickrey auction may also be extended to an online second-price auction wherein the bids are revealed upon request.

English auctions, wherein the price of the commodity increases with bids, are perhaps the most common auctions in practice. Coopinger, Smith and Titus [18] design a set of twelve experiments to examine the price optimality of English auctions and conclude, even mathematically, that the noninfinitesimal increase in the bids causes the winning bid (the last bid) to be slightly above the optimal price. However, a comparison of Pareto efficiency reveals that English auction is devoid of any strategic considerations and involves little to no expertise in bidders, making it the most efficient type of auction. The authors further reveal that a Second-Price Auction consists of minimal learning and hence is the second most efficient type of auction.

When any type of auction is applied to a crowdsourced system, it is essential to understand how the incentives drive participation and bidding. DiPalantino and Vojnovic
provide an experiment to obtain a quantitative solution to this question. The values of the experiments are then compared with available data from an online source. The paper considers a single bid auction which is analogous to how a crowdsourced system would be in an ideal scenario (availability of participants). The “player skills” discussed in the paper may be translated to explicit driver specific information such as vehicle capacity, fuel economy etc. The authors provide an interesting insight regarding the variation of the participation level with changes in incentives. Though an increase would represent an increase in the volume of participants, the quality of service does not undergo much impact.

In a crowdsourced parcel delivery system, upon starting an auction at a time instance, the participating drivers at that instance make their bids based on a utility function that is unique for a driver.

### 2.3 Utility Function

When a driver participates in an auction in the system, his bid to a particular parcel at a time instance depends on a variety of factors such as the distance to be travelled, the time taken for the journey, preference of the route, vehicle capacity, profit made among others. These parameters may be consolidated into a single utility function that determines the price that he or she bids.

Lee, Kang and Prabhu [20] provide a utility function framework from a system’s perspective wherein the function helps to determine as to whether or not the job is to be
accepted. If accepted, the job is added to the Transportation Managements system. The TMS, then, dynamically solves a routing problem for list of jobs present. The section that is of interest for this project is that of the control law, which, in its most basic form, is stated as follows:

\[
\text{Total Cost} = \text{Fuel cost} + \text{Emission cost} + \text{Penalty Cost} \tag{2}
\]

\[
\text{Fuel Cost} + \text{Emission Cost} = (c_f \cdot d_k) \cdot f(e_k, w, \psi) \cdot g(c_e) \tag{3}
\]

\[
\text{Penalty Cost} = c_p(\text{MSD}) \tag{4}
\]

Where,

\[c_f = \text{coefficient of fuel}\]

\[c_e = \text{coefficient of emission}\]

\[c_p = \text{coefficient of penalty}\]

\[e_k = \text{delivery request of } k^{th} \text{ class}\]

\[d_k = \text{Total distance covered}\]

\[w = \text{weight of the vehicle}\]

\[\psi = \text{Average speed of the vehicle}\]

\[g(c_e) = \text{A function of coefficient of emission}\]

\[\text{MSD} = \text{Mean square deviation of earliness and tardiness}\]

Once a job is admitted, a dynamic algorithm, which takes into account the various factors considered above, is used for routing. Through simulation, promising outputs and increase in transportation revenue were observed. The paper provides a scope for future work in the improvement of the algorithm in multiple vehicle and multimodal cases.
Modesti and Sciomachen [21] provide a possible solution wherein a multi-criteria shortest path problem is solved for multimodal transportation networks. It proposes a deterministic utility function that considers the mode of transport and other previously discussed parameters, both qualitative and quantitative, to reduce the overall cost and consider the problem as a regular short path problem.

The authors of the paper model the function as an algebraic sum of coefficients and costs multiplied with binary indicators that indicate various available classes. As a pilot study, the inclusion of qualitative parameters that include participant preferences is promising. Experimental results indicate slight improvement in solution quality when user preferences are taken into consideration, however, the reduction in search space by considering the input from the user is small when compared to the overall information fed to the system. A research into multiple user preference assignment and comparison with change in computation time may be carried out to find the correlation between user assigned variables and computation time.

Almeida [22] proposes the use of ELECTRE method [23] to evaluate a multiple criteria problem. Further, each objective is evaluated using a separate utility function. The author models the individual utility functions as an exponential term with a coefficient to assess the weights. Further, to the integrated utility function, a quality of service evaluation is added through a probabilistic parameter. As a result, the utility function not only considers the various parameters necessary, but also introduces a human error of reading the criteria through a probabilistic structure.
2.4 Algorithms for Traveling Salesman Problem

A traveling salesman problem is a class of vehicle routing problems in which the subject has an enforced constraint of returning to the first node after visiting all nodes in the path.

The Traveling salesman problem can be modeled as follows:

\[ \text{Min } Z = \sum_{i=1}^{n-1} D(p_r(i), p_r(i+1)) + D(p_r(n), p_r(1)) \]  \hspace{1cm} (5)

Where, \( Z \) = tour length

\( n \) = number of points to visit.

\( r \) = a particular sequence of routes in an iteration

\( D(p_r(i), p_r(i+1)) \) = distance between point \( i \) and \( (i+1) \) in a certain route \( r \).

For the purpose of this thesis, we focus only on the symmetric case of the TSP wherein \( D(p_i,p_j) = D(p_j,p_i) \) for \( 1 \leq i,j \leq n \).

There are various heuristics in literature for solving the Traveling Salesman Problem. Many of these heuristics find uses in a variety of problems. However, we consider the variations of these problems to solve the Traveling Salesman Problem. In this survey, 2 methods, viz., the Distributed Control Algorithm and the Genetic Algorithm are reviewed.
2.4.1 Distributed Control Algorithm

In order to cope with complexity, arising, in part, due to the scale of the problem, decentralization of previously centralized problems has been an active field of research in the recent past. The concept of Distributed Arrival Time Control in order to address the problem of discrete-event timing by treating the arrival time as a continuous variable has made it possible, in theory, to reduce the complexity of the problem as size increases [24]. Further, modeling the problem as a heterarchical system by minimizing global information and maximizing autonomy, it is possible to make the problem highly scalable [25].

Prabhu [26] presents an algorithm that utilizes cooperative scheduling wherein solutions are generated continuously using local information present at the time. The feedback loop as shown in figure 2 utilizes the control law below to ensure that the schedules generated in the following iterations tries to penalize both earliness and tardiness of the delivery. This further indicates that the changes to the information in the system could result in bad solution in the following iteration.

![Figure 2: Feedback loop of the DATC][36]

The DATC control law is as follows:
\[ a_j(t) = k_j \int_0^t (d_j - c_j(\tau)) \, d\tau + a_j(0) \]  

(6)

Where,

- \( k_j \) = controller gain for part \( j \)
- \( a_j(t) \) = arrival time of part \( j \) at time \( t \)
- \( d_j \) = delivery time
- \( c_j \) = completion time
- \( a_j(0) \) = arbitrary initial condition

As seen in the equation above, the initial condition does not affect the value of the final solution for a sufficiently large value of \( t \). It can also be seen that the arrival time is considered as a continuous variable. When a case with one machine and two parts at a given time (with processing time \( p_i \) and \( p_{i+1} \)) are considered, the solution space of arrival time may be classified into 3 regions viz., discontinuous region, dead-zone region and decoupled region as shown in figure 3.

It is worth noting that even though the arrival times are considered to be continuous, the resulting region may be discontinuous. This is because, in certain conditions, small changes in arrival time may result in changes to the processing sequence.

In a two-part case, when \( a_2 \geq (a_1+p_1) \) or \( a_1 \geq (a_2+p_2) \) there is no waiting time involved for processing the part. This results in processing sequence of (1,2) and (2,1) respectively. This case is indicated by the decoupled region in figure 3.
When \( a_2 \leq (a_1+p_1) \) or \( a_1 \leq (a_2+p_2) \) given that \( a_1 \neq a_2 \), the system is in dead-zone region. Here, when the left-hand side of equations changes within the specified conditions, the completion time of the parts changes but does not affect the processing sequence of the jobs. When \( a_1 = a_2 \), even small changes in the values of the arrival time can result in changes in sequence. This is the discontinuous region.

When the system enters the discontinuous region, there are 2 cases: Feasible due dates and Infeasible due dates. When the due dates are feasible, the solution in the steady state will be in the decoupled region. However, when the due dates are infeasible, the
steady state solution for the parts with infeasible due dates will be in the discontinuous region.

2.4.2 Genetic Algorithm and Ant Colony Optimization

The GA and ACO are simulation techniques used in search algorithms to find a good path to reach a goal state from a start node. Both these techniques involve probabilistic parameters that define the movement of an entity from a node to the next in search of the goal.

For a TSP, the Ant Colony Optimization algorithm has been applied both purely and with improvements. Since the vehicle routing problem is similar to TSP, we can extend the results from a VRP that is solved using ACO to TSP. From the purview of TSP, the GA works analogous to biological evolution that is modified in iterations with a set of probabilities to obtain better solutions.

Bell and Mullen [27] discuss how the ACO algorithm’s meta-heuristic compares with other methods such as Tabu Search and Genetic Algorithms from the standpoint of vehicle routing problems. The authors start by indicating the exponential increase of the feasible solution due to a linear increase in the number of participants. For this NP-hard problem, the authors design an experiment that consists of three VRPs found in literature (of varying sizes of participants). The quality of solutions ranges between 1 to 6.5% of the optimal solution. This indicates that the ACO in its purest form is not a good heuristic for larger problems in vehicle routing.
This paper further provides a techniques of multiple ant colony algorithm, which provides competitive solutions to VRP. Since the problem does not always yield optimal solutions, but is capable of generating good answers in real-time, ACO may be used for small to medium sized VRPs.

Bin, Zhong-Zhen and Baozhen [28] propose an improvement to the ACO algorithm by changing the pheromone update strategy and introducing mutation of routes.

1. The Pheromone Update (Ant-weight strategy) utilizes the information about the contribution of each link towards the solution in that iteration. This weight is inculcated into the pheromone update equation. This would make sure that the pheromone on the link in question has taken into account both the global and the local information, thereby facilitating the searches in the area where the links provided higher contribution.

2. Mutation Operation involves randomly changing the route to a nearby route, thus saving memory and iterations, according to a predetermined probability ($p_m$). The new solution produced varies very little from the original solution. If this variation produces a better result, the pheromone is updated for the new solution instead of the original, thereby reducing the solution space.

Grefenstette et. Al. [29] and Chatterjee et. al. [30] provide a method of application of the Genetic Algorithm in the Traveling Salesman Problem. The authors provide an adjacency representation of routes that can possibly reduce the computational burden on
the system by representing a solution within a cell, thereby eliminating the need for multiple arrays with ordered lists of cities visited.

![Flowchart of the Genetic Algorithm][37]

The authors also discuss crossover heuristic and its impact on solution quality. Further, this paper suggests the utilization of a predetermined start city as it would decrease the search space by a factor of at least two.

While there is literature that separately addresses the traveling salesman problem using the two methods discussed in this review, a comparison between the two methods is relatively novel in this research. Further, a model for the entire crowdsourced delivery system using auctions to choose drivers is a new concept.
Chapter 3 : Data Description

Since the model of a crowdsourced parcel delivery is relatively novel, there is very little readily available data for result comparison. However, data regarding a traditional last-mile parcel delivery system can be simulated.

i. Delivery destination: The delivery destinations are chosen as random points within a limited radius. The distance between 2 destinations is the straight-line distance between them. It is seen that the straight-line distance may be used as a reliable heuristic to obtain sequences. In this thesis, 500 destinations are considered that require delivery.

ii. Routes: In order to account for situations wherein two close destinations (based on straight-line distance between them) are not accessible by road of a proportional length, routes are assigned to destinations. By doing so, an implicit path is generated in the code along which the vehicles traverse.

iii. Parcel size: The parcels are assigned random standard sizes, as done by most logistic providers. For the purpose of this thesis, 4 sizes are assigned to the parcels with allocated sizes between 1 and 4. 4 being the largest size classification of the parcel allowed for crowdsourcing.

iv. Vehicle capacity: The capacity of the vehicle may range from 5 to 50. This information is related to the size of the parcel - indicating that a driver can, at least, hold one parcel.
v. Drivers: The number of drivers participating in the auction may be arbitrary. For this thesis, between 5-10 drivers are used per auction.

vi. Auction response: This is a binary variable that designates whether a driver wants to deliver the said package based on the reward for delivering it.

vii. Driver Credibility: Each driver is assigned a normalized “Mobility” and “Trust” rating to determine past participation and reputation respectively.

As it can be seen from the chosen random data, the system is simulating a city in order to determine the success of the parcel delivery. For the purpose of this thesis, a city bound by the points (10,10), (-10,10), (-10,-10), (10,-10) is chosen within which a set number of parcels are to be delivered.

Figure 5: Distribution of destinations
The data has been represented in tuples and nested tuples for the driver, driver credibility, parcels and results of auction.

\[
\text{Driver} = D_i = (\rho_i, C_i, \text{cap}_i), \quad (7)
\]

Where, \( \rho_i = \) Route Preference of driver \( i \)

\( C_i = \) Credibility of driver \( i \)

\( \text{Cap}_i = \) Vehicle Capacity of driver \( i \)

\[
\text{Credibility} = C_i = (\text{Mob}_i, \text{Tru}_i) \quad (8)
\]

Where, \( \text{Mob}_i = \) Implicit Locations (Mobility) of driver \( i \)

\( \text{Trust}_i = \) Reputation score of driver \( i \)

\[
\text{Parcel} = P_n = (d_n, p_n, r_n, s_n) \quad (9)
\]

Where, \( d_n = \) Due time of parcel \( n \)

\( p_n = \) Processing time of parcel \( n \) (destination distance from warehouse)

\( r_n = \) Route where parcel \( n \) must be delivered

\( s_n = \) Size of parcel \( n \)

\[
\text{Auction} = A_{in} = (\delta_{ni}, \text{Dr}_{in}, \text{arr}_{in}) \quad (10)
\]

Where, \( \delta_{ni} = \) Binary variable indicating whether parcel \( n \) is assigned to driver \( i \)

\( \text{Dr}_{in} = \) Binary variable to indicate if driver \( i \) chooses parcel \( n \) in the auction

\( \text{arr}_{in} = \) Arrival time of driver \( i \) to pick parcel \( n \)
Moreover, in order to ensure that the solution is not computed on empty vectors, a lower limit, greater than zero, is set for each parameter of the Driver and Parcel tuples during parameterization. Discrete whole number values are assumed for all sizes, times and capacities. From a pure computational perspective, a discrete case helps in reducing the complexity. However, for the GA and DATC algorithm, the integer constraint is relaxed.

A matrix of distances between delivery locations is made using randomly generated Cartesian co-ordinates of delivery locations. The matrix works under the assumption that straight line distances is an admissible measure of the actual distance between the two points. Therefore, a straight-line distance heuristic is implicitly applied to generate the matrix of distances for the Traveling Salesman problem. Additionally, adding routes to each destination adds randomness and can make close graphically close destinations fall under different routes, thus simulating a non-grid system of roads.

In the Genetic Algorithm that follows, the probabilities of crossover, mutation of routes and mutation of nodes are provided with comparable values found in literature [31]. The population size and the number of generations have the highest impact on the solution obtained from GA. These are provided with varying values to check for change in the computation time vs solution quality.

For obtaining the distance using the distributed algorithm scaling factors ranging from 0.05 to .05 are used. The number of iterations are varied to determine when the system reaches steady state.
Chapter 4: Methodology

The model attempts to minimize global information at every step. By doing so, the stress on the system is reduced and as a result, the problem complexity decreases. This heterarchical control in the system makes it highly scalable and can provide solutions in real-time for scenarios with large number of participants.

Further, the model not only makes the entities highly autonomous, but in doing so, reduces the search space in each step, thus reducing the burden on the system. By making a subset of the solution space inadmissible, the system assesses a lesser number of solutions at any given iteration.

The system consists of 3 parts –

i. Parcel and Driver Information

ii. Route Assignment

iii. Auctioning for parcel assignment

iv. Routing.

4.1 Parcel and Driver Information

Every parcel that enters a system at a given point in time contains information about its due time, destination and size. The route of the parcel is analyzed by the system based on destination. This is done for only those parcels that enter the system.
Each driver participating at a given point in time has his vehicle capacity and credibility information present in the database. However, this information does not enter the system until the driver decides to participate by selecting a route.

<table>
<thead>
<tr>
<th>Driver Number</th>
<th>Route</th>
<th>Credibility Weight</th>
<th>Vehicle Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>0.6278</td>
<td>24</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0.6905</td>
<td>42</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>0.6546</td>
<td>48</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0.4735</td>
<td>55</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>0.4359</td>
<td>37</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>0.5574</td>
<td>23</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>0.8899</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>0.5688</td>
<td>78</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>0.4996</td>
<td>51</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>0.7424</td>
<td>40</td>
</tr>
</tbody>
</table>

Table 1: Example of driver tuple in the database.

<table>
<thead>
<tr>
<th>Parcel Number</th>
<th>Route</th>
<th>Distance from origin</th>
<th>Size of Parcel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>4.1231</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>10.1980</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>9.4868</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>8.9443</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>6.4031</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>9.2195</td>
<td>4</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>7.6158</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>10</td>
<td>12.8062</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>9.2195</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>5.5120</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 2: Example of Parcel tuple in the database.
4.2 Route assessment

An arbitrary number of routes are made available to the participants before the auction of parcels. The drivers decide their route preferences and they are considered for only those parcels that have destinations in those routes. A driver may change his route preference after he chooses his route the first time. However, if the auction has begun before he changes his route, he will not be considered as a participant for that auction. This step ensures that the system need not make changes to available information once the calculations start. Since the system only considers the participating drivers’ and the parcels’ information, the complexity of the system is reduced. Thus, for n participating drivers, n vectors of parcels are created. A driver may only bid for those within his vector of parcels.

4.3 Auctioning for parcel assignment and bin packing

The drivers start bidding for those parcels that are in their vector based on the reward offered for the parcels. At this point, the system has requests from various drivers for each parcel. Thus, each parcel has a vector of j different drivers that may bid for it. The bidding variable $A_{ij}$ is a binary variable. If a driver wants to deliver a parcel, the state of the Auction variable for parcel i and driver j is $A_{ij} = 0$. Otherwise, $A_{ij} = 1$.

Here, the system may use a FCFS rule in order to reduce complexity. However, a limit is placed on the number of parcels a driver can take at each auction. This is done using the vehicle capacity and parcel size information. If FCFS rule is utilized in the
system, the problem of integer bin packing reduces to a limiting variable. For instance, a
driver with a vehicle capacity of 10, who has already chosen two parcels of size 4 each
can only bid for one size 2 parcel or two size 1 parcels. The flowchart of parcel
assignment is shown in figure 6. Note that a parcel is deemed ineligible for an iteration if
the conditions within those iterations are not met. The parcel may enter the system again
depending on the threshold condition as shown in figure 7.

![Flowchart of parcel assignment](image)

**Figure 6: Eligibility criteria for parcel assignment**

Once the parcels are assigned to the drivers, 2 steps are necessary:
1. Parcel processing sequencing using an algorithm

2. Assessing the parcels that were not assigned to a driver in the auction.

When a parcel is not assigned to a driver, the system must decide whether the parcel needs to be auctioned in a new iteration or if it should be delivered using existing traditional methods. An increase in reward for delivering the said parcel may also be an option.

Figure 7: Threshold criteria for using traditional delivery
From (7), (8), (9) and (10), the assignment of a parcel to a driver may be assessed as follows:

For a parcel \( n \) and driver \( i \), if

1. \( \rho_i = r_n \)
   (Part has a destination in the same route as the driver wants to travel in)

2. And, \( \text{cap}_i \geq \left[ \sum_{z=1}^{t(n-1)} s_{(z-1)}(\delta_{(z-1)i}) \right] + s_n \)
   (The capacity of the vehicle is at least equal to the total size of assigned parcels)

3. And, \( \sum_{k=1}^{(i-1)} \delta_{nk} = 0 \)
   (Parcel \( n \) has not been assigned to any other driver in the queue)

4. And, \( C_i \geq \) (\textit{minimum predetermined credibility threshold})

5. \( Dr_{in} = 1 \)
   (Driver \( i \) chooses parcel \( n \) in the auction, based on the offered reward)

Then, \( \delta_{ni} = 1 \)

This indicates that the driver \( i \) may be assigned to parcel \( n \).

### 4.4 Routing

Once the parcels are assigned to the drivers, there is a vehicle routing problem that must be solved for each driver. This is done using two different models:

1. Genetic Algorithm
2. Distributed Algorithm
4.4.1 Genetic Algorithm

This algorithm works on the principle of natural selection wherein solutions are measured based on a fitness function, in this case – total length of the path. The algorithm proceeds over the iterations searching for a better solution. Roulette wheel method – wherein the chances of choosing parents (earlier sequences) is directly proportional to their fitness value – is used to randomly select the first generation with a higher probability of obtaining a gene with higher fitness value. Each iteration is carried out using crossovers and mutation of genes as shown in the 5 gene 3 iteration example shown in table 3.

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Sequence</th>
<th>Path Length</th>
<th>Crossover</th>
<th>Mutation</th>
<th>New Path Length</th>
<th>Change in sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1,2,3,4,5</td>
<td>27.6</td>
<td>cell 3 to cell 5</td>
<td>No</td>
<td>29.8</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>1,2,3,4,5</td>
<td>27.6</td>
<td>cell 2 to cell 1</td>
<td>No</td>
<td>21.4</td>
<td>Yes</td>
</tr>
<tr>
<td>3</td>
<td>2,1,3,4,5</td>
<td>21.4</td>
<td>cell 3 to cell 2</td>
<td>cell 4 to cell 1</td>
<td>20.8</td>
<td>Yes</td>
</tr>
<tr>
<td>End</td>
<td>4,3,1,2,5</td>
<td>20.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Propagation of the Genetic Algorithm

Here, it can be seen that the value of the total distance reduced from 29.8 units to 20.8 units.

Since the ideal solution is not known, the stopping criterion is the number of generations specified. Similarly, elitism of a gene is calculated as a comparative estimate between $i^{\text{th}}$ and $(i-1)^{\text{th}}$ generation. The new route is only chosen in the $(i+1)^{\text{th}}$ generation if the new fitness function was greater than the previous fitness function.
The Genetic Algorithm parameters that determine the solution for each Traveling salesman problem are shown in table 4.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>prob_mutn</td>
<td>Node Exchange Mutation Probability ($p_{mn}$)</td>
</tr>
<tr>
<td>prob_mutp</td>
<td>Path Exchange Mutation Probability ($p_{mp}$)</td>
</tr>
<tr>
<td>prob_mutc</td>
<td>Probability of Crossover ($p_c$)</td>
</tr>
<tr>
<td>pop_size</td>
<td>Population Size</td>
</tr>
<tr>
<td>num_generations</td>
<td>Number of Generations</td>
</tr>
</tbody>
</table>

Table 4: Parameters that affect the final solution in GA

4.1.2 Distributed Algorithm

In a distributed arrival time controller, the earliness and the tardiness of a delivery is equally punished. Therefore, by providing equal due times for all parcels that are being delivered by one driver, the system is forced to regard all parcels as equally important in terms of delivery time. However, numerically, the distance between two destinations can be expressed in terms of time. Therefore, the cumulative distance is used as the completion time in the equation.

A due time of zero is set in order to ensure that there is equal competition among parcels for being delivered first. Thus, the importance of ordering only depends on the distance between them that is captured in the variable $p_n$ (processing time of parcel n).

Thus, the transformed distributed arrival time controller equation becomes:

$$D_j(t) = k_j \int_0^t (-c_j(\tau)) \, d\tau + D_j(0)$$
Where,

\[ D_j(t) = \text{Overall path distance till parcel } j \text{ in the sequence.} \]

\[ k_j = \text{Scaling Factor for parcel } j. \]

\[ c_j(\tau) = \text{Overall path distance till parcel } j \text{ in the sequence at iteration } \tau. \]

\[ D_j(0) = \text{Arbitrary initial path distance to get to parcel } j. \]

Thus, the arrival time of each parcel is adjusted to ensure that there is least overall deviation of due time. Thus, the model can be expressed as a goal programming problem as shown below:

**Minimize** \( D^+ + D^- \)

**Subject to:**

\[ D_j(t) + k_j \sum_{\tau=1}^{t} c_j(\tau) = g_j + D^+ - D^- \]

\[ c_j(\tau) = \sum_{i=2}^{j} d_{i,(i-1)}(\tau) \]

\[ D^+, D^- \geq 0 \]

\[ d_{i,(i-1)}(\tau) > 0 \]

Where,

\( d_{i,(i-1)}(\tau) \) is the distance between destination \( i \) and \( (i-1) \) in the \( \tau^{\text{th}} \) iteration.

\( g_j \) is goal cumulative distance for deliveries in sequence before parcel \( j. \)

\( D^+ \) and \( D^- \) are the positive and negative deviations from the due time.
In the model above, we can see that the values of distances are strictly positive. This indicates that, if the “g” variable is set to zero for all deliveries, then only $D^+$ is positive and $D^-$ is zero. Thus, the objective function would just minimize $D^+$, and therefore acts as a heuristic for finding the sequence of deliveries that have the least overall deviation from the due time.

Thus, there exists a trade-off between the DATC and GA models in terms of path distance, deviation of due time and computation time.
Chapter 5 : Experimental Results

5.1 Parameter Variations

The solution quality mainly depends on 3 parameters: Number of routes, number of drivers and number of parcels. In order to check the solution quality across the established limits of the parameters, the model must be simulated across the minimum and the maximum values of each of the three parameters. Thus, there are 8 possible experimental conditions within which the simulation may be done as shown in Table 6.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>High</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of drivers</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>Number of routes</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>Number of parcels</td>
<td>500</td>
<td>10</td>
</tr>
<tr>
<td>Vehicle capacity</td>
<td>50</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5: Bounds of the parameters in the system

<table>
<thead>
<tr>
<th>Configuration number</th>
<th>Configuration</th>
<th>Number of Parcels</th>
<th>Number of Drivers</th>
<th>Number of Routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LLL</td>
<td>10</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>LLH</td>
<td>10</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>LHL</td>
<td>10</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>LHH</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>HLL</td>
<td>500</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>HLH</td>
<td>500</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>HHL</td>
<td>500</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>HHH</td>
<td>500</td>
<td>10</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 6: Various system configurations based on the values of the parameters.

The color maps for various configurations or routes indicate the sequence of delivery for each of the 50 genes considered for the genetic algorithm problem in the
system. The value of the solution for each gene progressively gets better with each iteration. Best solution value is populated based on the last iteration’s solution of these 50 genes.

**Configuration 1**

For all the parameters set to the lower limit, since the system is designed to allot a route to each parcel, we see that all parcels have destinations in route 1. Therefore, this configuration tests the Auctioning process. A normalized parameter indicating the likelihood of the driver choosing the parcel can be established in order to simulate real conditions of response rates. Additionally, the auction may be repeated for unallocated parcels based on a certain condition such as a threshold time beyond which delivery by traditional methods is more profitable than through crowdsourcing.

![Image](image_url)

**Figure 8: Best routing sequence (left) and sequence of delivery in different genes (right) for configuration 1**
Figure 8 shows the routing of parcels recommended by GA. The matrix indicates the sequencing of parcels in different genes. The genes with the higher fitness function (lower path distance) have a higher probability to be picked for successive generations. As we can from the figure above, all available parcels were picked during the auction for delivery. This is because of weighted response rates and multiple iterations of the auction with increasing reward functions.

<table>
<thead>
<tr>
<th>Parcel Number</th>
<th>Iteration i</th>
<th>Iteration (i+1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>38.172</td>
<td>-38.172</td>
</tr>
<tr>
<td>2</td>
<td>43.498</td>
<td>-43.498</td>
</tr>
<tr>
<td>3</td>
<td>-4.706</td>
<td>4.706</td>
</tr>
<tr>
<td>4</td>
<td>-18.975</td>
<td>18.975</td>
</tr>
<tr>
<td>5</td>
<td>13.113</td>
<td>-13.113</td>
</tr>
<tr>
<td>6</td>
<td>1.696</td>
<td>-1.696</td>
</tr>
<tr>
<td>7</td>
<td>14.205</td>
<td>-14.205</td>
</tr>
<tr>
<td>8</td>
<td>35.542</td>
<td>-35.542</td>
</tr>
<tr>
<td>9</td>
<td>17.318</td>
<td>-17.318</td>
</tr>
<tr>
<td>10</td>
<td>25.645</td>
<td>-25.645</td>
</tr>
</tbody>
</table>

Table 7: Alternating solutions between iterations in DATC

<table>
<thead>
<tr>
<th>Genetic Algorithm</th>
<th>DATC</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>2</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>
Table 8: Parcel sequence variation in DATC and GA

For the due date requirements as the priority, using the DATC, we obtain an alternating value of completion times between iterations. This is because of infeasible due dates. The sequences of parcel delivery for the two methods are shown in table 8. The solution using the two models are shown in table 9:

<table>
<thead>
<tr>
<th></th>
<th>Path distance</th>
<th>Computation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>GA</td>
<td>48.102</td>
<td>0.289</td>
</tr>
<tr>
<td>DATC</td>
<td>62.472</td>
<td>0.117</td>
</tr>
</tbody>
</table>

Table 9: Solution quality comparison between the two methods.

It is observed that there is a 30% increase in path distance using the DATC when compared to the GA. However, since earliness and tardiness are punished in the DATC, the parcels are sequenced in such a way that the average earliness and tardiness are minimized. However, the trade-off is the computation time. Using the GA increased the computation time by 1.5 times when compared to that of DATC.

Configuration 2

In this configuration, the number of routes is increased to its upper limit while there is only one driver participating in the auction. The parcels are, therefore, assigned randomly to each available route. Obtaining a solution in this configuration is only possible when there are more than three parcels assigned to a route and has the driver picking the same route to deliver in. Hence, the limiting parameter in this configuration is
the number of parcels. As the number of parcels increases, the probability of obtaining more than three parcels in a given route increases.

**Configuration 3**

The limitation in this configuration is the auctioning of parcels. Since there are only a few parcels available, the probability of having more than 3 parcels per driver is low. Since there is a one-to-one relation between the parcels and routes, the system works like configuration 1 for all drivers who are delivering more than 3 parcels. However, overall, the system will not have enough destinations to provide sequences for all drivers.

**Configuration 4**

For the purpose of finding a solution, this configuration is similar to configuration 2 and works only for sufficiently large number of parcels. Hence, the number of parcels act as a limiting parameter for this configuration.

**Configuration 5**

Due to minimized global information, the drivers are oblivious to the increase in the number of parcels. It is observed that the drivers choose the parcels that appeal to them in terms of the reward offered and choice of destination regardless of the total number of parcels available for delivery. Therefore, this configuration works similar to configuration 1 and is only limited by the vehicle capacity.
Figure 9: Lack of assigned parcels in Configuration 4

Figure 10: Best routing sequence (left) and sequence of delivery in different genes (right) for configuration 1
**Configuration 6**

This configuration represents configuration 2 in which the upper bound on the number of parcels is relaxed to 500. Therefore, this configuration is just limited by the vehicle capacity. Figure 11 shows the number of parcels served by the driver compared to the number of parcels that are available to be delivered in the same route.

![Diagram](image)

*Figure 11: Comparison between fulfilled and unserved destinations.*

**Configuration 7**

This is similar to configuration 6 in that all routes that have a participating driver will serve the parcels in that route. Since the number of parcels is high, there is a higher possibility of a route having more than 3 parcels in it. Therefore, for a configuration
wherein the number of parcels is much greater than the number of routes and drivers, we can generalize and say that the limiting parameter is the vehicle capacity.

**Configuration 8**

In this configuration, all parameters are set to their upper limits. Since the number of parcels is still sufficiently larger than the number of routes and the number of drivers, we can see that this is an optimum setting for the model. Hence, it can be generalized that the model works best for large number of participants and parcels.

The model is limited by both the auctioning process and the vehicle capacity. Increasing the number of auction iterations or the vehicle capacity will ensure that more parcels are delivered by each driver. In this experiment, the number of iterations of the auctioning process in this configuration is increased sufficiently to ensure that the only limiting parameter is the vehicle capacity.

Figure 12 shows a case in configuration 8 wherein the number of parcels in a route is small due to the capacity of the vehicle. Figure 13 shows the case where the number of parcels available in the route is large and the number of parcels assigned to a driver in the same route is also large. This is because the driver’s vehicle capacity was large enough to accommodate a large number of parcels – further indicating that the capacity is the limiting factor.
The results for each of the available routes are shown in table 12. The trade-off can be observed between the two methods in terms of the distance travelled by the drivers and the computation time.

Figure 12: Parcel assignment limited by driver capacity.

Figure 13: Case wherein availability of parcels and vehicle capacity is high in a particular route
5.2 Limiting Parameters

With current values of the parameters, table 10 shows the limiting parameters for the configurations discussed in 5.1. It can be observed that the system performs better in conditions 1, 3 and 4.

<table>
<thead>
<tr>
<th>Configuration #</th>
<th>Limiting Parameters</th>
<th>Average service level</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Drivers/Vehicle capacity</td>
<td>80%</td>
</tr>
<tr>
<td>2</td>
<td>Drivers/Number of parcels</td>
<td>10%</td>
</tr>
<tr>
<td>3</td>
<td>Number of parcels</td>
<td>100%</td>
</tr>
<tr>
<td>4</td>
<td>Number of parcels</td>
<td>98%</td>
</tr>
<tr>
<td>5</td>
<td>Drivers/Vehicle Capacity</td>
<td>8%</td>
</tr>
<tr>
<td>6</td>
<td>Drivers/Vehicle Capacity</td>
<td>8%</td>
</tr>
<tr>
<td>7</td>
<td>Drivers/Vehicle Capacity</td>
<td>10%</td>
</tr>
<tr>
<td>8</td>
<td>Drivers/Vehicle Capacity</td>
<td>80%</td>
</tr>
</tbody>
</table>

Table 10: Limiting parameters for different configurations.

Assuming that the number of parcels that are to be delivered cannot be varied, the system performance can be improved in those conditions wherein the number of drivers or the vehicle capacity limits the performance of the system. If the vehicle capacity is increased, or in other words, more parcels are delivered by the same driver, the system performance increases. Doing so increases the collective capacity of vehicles traversing in a particular route. This increases the complexity of the routing.

On the other hand, if the number of drivers is increased, the number routing problems solved by the system increases. Both the methods increase the overall complexity of the system.
Figures 14 and 15 show the Genetic Algorithm solution at 500 generations and 1000 generations respectively for configuration 5 with a random vehicle capacity between 75 and 100. With the new vehicle capacity, the number of parcels serviced is between 19 and 100 per driver. It can be seen that with the new values of vehicle capacity the system does not reach the optimal value using the GA at 500 generations on most occasions.

![Figure 14: Final traversed route at 500 generations](image)

If the real-time threshold is set at a value at which GA does not provide optimal solutions, then there is a need to use a method that generates a good solution faster. Thus, in larger problems, the gap of the distance trade-off between DATC and GA reduces.
5.3 Sequencing using PID

Since all due times are set to a single value, in this case zero, there are infeasible. Therefore, the system tries to adjust itself in such a way that the overall deviation from the due date is minimized. In doing so, a few deliveries in the sequence will have negative completion time. This also presents a situation wherein a steady state value of arrival time or completion time. On the other hand, an alternating optimal value is achieved as shown for configuration 1 in table 7 and in figure 17 below.

If the due time is increased by a certain value, the completion time and the arrival time for all the deliveries are increased by the same value. Thus, the due date may be increased by the most negative value of the completion time, such that the new lowest
completion time value is equal to zero. In doing so, while the due date still remains infeasible, the completion times become admissible as they are all positive.

Figure 16: Completion time using the DATC for parcel 6 in configuration 1 example.

Figure 17: Completion time graph from Figure 17 captured separately for odd and even iterations.
5.4 Solution Quality

In order to provide a reliable method of comparison between the two methods used in this thesis, the quality of the solutions obtained using them must be compared. The four categories used for comparison are:

1. Computation Time
2. Total Distance Traveled
3. Due-time deviation

5.4.1 Computation Time

For configuration 8, the number of parcels may be increased further to test the scalability of the model. The number of parcels delivered is gradually increased from 50 to 500, with the unlimited vehicle capacity for one driver. Table 11 and figure 16 indicate the computation time for obtaining the best solutions using the two methods. While it can be observed from the graph that best solution computation time increases exponentially for both DATC and GA methods, the DATC still provides answers in real-time whereas GA fails for higher number of parcels.

It is also noticed from figure 16 that the gradient of increase in computation time is similar in both methods. However, from a standpoint of obtaining real-time solutions, the DATC works significantly better for larger problems.
<table>
<thead>
<tr>
<th>Parcels Delivered</th>
<th>Time for best solution</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DATC</td>
<td>GA</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>1.417</td>
<td>12.431</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>2.638</td>
<td>22.653</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>3.924</td>
<td>51.325</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>5.137</td>
<td>117.141</td>
<td></td>
</tr>
<tr>
<td>250</td>
<td>7.921</td>
<td>321.912</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>9.243</td>
<td>793.758</td>
<td></td>
</tr>
<tr>
<td>350</td>
<td>13.266</td>
<td>1212.317</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>18.81</td>
<td>2067.154</td>
<td></td>
</tr>
<tr>
<td>450</td>
<td>24.267</td>
<td>3327.662</td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>35.747</td>
<td>5310.741</td>
<td></td>
</tr>
</tbody>
</table>

Table 11: Comparison of computation times for obtaining best solutions.

Figure 18: Comparison between computation times as the number of deliveries increases.

5.4.2 Total Distance Traveled

The total distance traveled by a participating driver is one of the measures of the solution quality of a routing problem. The GA solves for minimum travel distance, and hence will have perform better for this measure.
The distance trade-off is presented in table 12. The values generated are for optimal solutions/best solutions obtained in the Configuration 8 example. Though it is seen that the GA provides better optimal completion times, for larger problems, it fails to generate solutions in real time as was seen in section 5.4.1. Hence, this measure of solution quality would be relevant to the system for smaller problems as the GA fails to provide real-time solutions to larger problems.

<table>
<thead>
<tr>
<th>Route Number</th>
<th>Travel Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GA</td>
</tr>
<tr>
<td>1</td>
<td>63.0805</td>
</tr>
<tr>
<td>2</td>
<td>77.1393</td>
</tr>
<tr>
<td>3</td>
<td>86.2073</td>
</tr>
<tr>
<td>4</td>
<td>64.3698</td>
</tr>
<tr>
<td>5</td>
<td>89.8347</td>
</tr>
<tr>
<td>6</td>
<td>58.6156</td>
</tr>
<tr>
<td>7</td>
<td>127.1585</td>
</tr>
<tr>
<td>8</td>
<td>72.2786</td>
</tr>
<tr>
<td>9</td>
<td>75.914</td>
</tr>
</tbody>
</table>

Table 12: Travel Distance for Configuration 8 example.

5.4.3 Due Time Variation

The DATC solves for minimum Due Time deviation, therefore, the completion times are adjusted to have the least possible overall earliness or tardiness. In other words, the parcels are delivered just-in-time (JIT). Obtaining such a solution is advantageous when there are limited time-windows of delivery. The GA, however, does not provide a good solution for due time deviation.
Measure of solution quality using due time deviation and total distance traveled are inversely related. While the DATC provides good solutions in terms of JIT, GA performs better in terms of total distance traveled. Therefore, a trade-off graph can be constructed with computation time and either the distance traveled or the due time deviation.

Figure 19 provides a comparison between the percentage difference of computation times and the percentage difference of distance traveled using the two methods with respect to DATC. In this graph, a 100:1 weightage is placed on the distance traveled when compared to the computation time. It can be observed that even with a very high weightage on due-time deviation, the GA is not a preferred method beyond 220 deliveries.

Figure 19: Comparison of solution quality of GA and DATC.
Chapter 6 : Conclusions and Future Work

There is growing interest in crowdsourcing as a business proposition and this thesis explores a method of conducting the activity while addressing the issue of complexity that arises in parcel assignment and routing. It describes a new crowdsourcing parcel delivery system that utilizes an auction mechanism to assign parcels to participating drivers and proposes two different models to find the sequence of delivery. Further, the model may be utilized for larger systems with higher number of participants and parcels.

The various configurations discussed in chapter 5 indicate the parameters that affect the service level. The simulations indicate that with increase in the number of participants and the frequency of auctioning, larger number of parcels are delivered thereby, increasing the service level. Hence, it is suited for high number of drivers delivering a large number of parcels – ideally in a big city.

Additionally, the results provide information about scalability of the model and indicate the effectiveness of the two methods as the number of parcels and the number of drivers increase.

6.1 Limitations of the System

While the system provides solutions, there are certain parameters and constraints that it fails to consider. These limitations, while restricting the usage of the system to certain conditions, can be addressed separately and incorporated in the future work.
1. The overall profitability is not considered in the model. While it is intuitive that as the size of the system increases, higher profits are expected, it is not expressed in the system as a separate threshold point. From figure 7, it can be seen that due time acts as the only means for choosing traditional delivery. However, the system does not mathematically address the threshold break-even point.

2. In general, the service level of the system decreases as the number of participating drivers reduces. Having lesser number of drivers not only reduces the number of parcels available to the driver and fails to generate a sequence when there are less than 3 parcels, but also makes the system less profitable.

3. When the ratio of the number of parcels to the number of routes is low, lesser number of parcels have the probability of traveling in a particular route. This makes crowdsourcing less profitable.

4. It can be observed from the results that GA is not very scalable. Thus, obtaining a real-time delivery sequences becomes harder as the number of parcels or drivers increases.

6.2 Future Scope

1. The lower service level of the system in Configurations 2, 5, and 6 can be avoided by having an explicit service level threshold in the system beyond
which traditional delivery methods will be employed. This ensures the overall service level of the system.

2. While the Distributed Algorithm based method provides solution to the due time deviation problem, it may be transformed to provide the sequence with the least travel distance. The same can be done with the Genetic Algorithm wherein the due time problem is solved. This would ensure that the two methods may be compared directly based on the requirement of the service provider and the method with lesser computation time is chosen. Thus, the overall complexity of the system reduces.

3. The threshold point of profitability may be included in the model so that the system does not compute the sequences for situations wherein delivery is not profitable.

4. Parcel value estimation: The model assesses the parcel sizes and assigns them to a driver through bin packing and auction. However, a variable indicating the value of the parcel may also be introduced – thus creating situations wherein a parcel is prioritized in the same auction.
References


