Improving the Efficiency of a Decentralised Dynamic Last-mile Parcel Distribution Model and Adding the Circumstances of Mixed Fleets and Congestion

Master thesis submitted to Delft University of Technology in partial fulfilment of the requirements for the degree of
MASTER OF SCIENCE
in Complex Systems Engineering and Management
Faculty of Technology, Policy and Management
By

C.J. van der Wal
Student number: 5410703

To be defended in public on 29/8/2022

Graduation committee:

Chairperson:	Prof. Dr. Ir. L.A. Tavasszy	Transport and Logistics
First Supervisor:	Dr. J.H.R. van Duin	Transport and Logistics
Second Supervisor:	Prof. Dr. M.E. Warnier	Systems Engineering
External Supervisor:	D. Valdivia	Prime Vision B.V.

Acknowledgments

I am pleased to present my master thesis "Improving the efficiency of a decentralised dynamic last-mile parcel distribution model and adding the circumstances of mixed fleets and congestion". I want to express a few words of gratitude. My thanks go to my supervisors at TU Delft and Prime Vision. From the beginning, I was fed with enthusiasm and interesting suggestions. Many thanks to; Ron van Duin for his insights into the subject and motivating words along the way. Diego Valdivia, for the enjoyable meetings and always willing to help with anything. Lóri Tavasszy, always giving the necessary guidance. Finally, Martijn Warnier, who always asked the right questions. Thanks also to my family, friends, classmates and housemates who helped me along the way. It has been a pleasant process for which I am grateful.

Joost van der Wal
Amsterdam, July 2022

Executive Summary

Today, logistics systems are becoming larger and more complex than ever before. Numerous concepts and paradigms have been proposed to deal with this problem. Self-organising principles in logistics systems can be seen as a potential remedy. This study aims to investigate and develop efficiency improvements for a decentralised dynamic last-mile parcel delivery system. A better understanding of the efficiency, and thus the feasibility of this decentralised logistics method, helps to understand the potential of these systems as alternatives to the current centrally organised system. The main research question addressed is: *To what extent can the efficiency of a decentralised dynamic parcel distribution method be improved by applying different heuristics and adapting the simulation environment with congestion and a mixed fleet?* To answer this question a review of existing literature on last-mile logistics, self-organisation in logistics and auction methods in logistics is conducted. The current model is evaluated, benchmarked and validated. The improvements of the model are defined and applied in an experimental setting using a small data set, after which the extended model is applied to a real-scale case data set based on 729 delivery addresses in Delft.

The problem of last-mile delivery is related to various topics of operational optimisation. It has characteristics in common with routing problems, transport planning problems and scheduling problems. In today's last-mile delivery, the allocation and routing of parcels are performed separately. First, the clustering of postal codes is used for allocation and then a routing problem is solved for each individual vehicle. Since the delivery zones are often set for several weeks, daily fluctuations in the number of parcels can lead to suboptimal utilisation and routing of the vehicles within the delivery fleet. A decentralised dynamic parcel distribution system, inspired by self-organisation principles, would allow for combinatorial optimisation and dynamic allocation, thus overcoming the restriction of strict delivery zones.

When self-organising principles are taken into account, the question arises why a self-organising network is created. For several domains (logistics, mobility, energy, etc.) the drivers are the increasing complexity and scale of systems and processes, the need for greater efficiency, the need for flexibility and the need for robustness against failure of system components. Self-organisation is a broad concept with many different interpretations and the literature is mainly conceptual. The method used to enable self-organising features in this study is a single parcel auction method where the parcels are inserted into the route of the bidding vehicles. The auction system defined for this method is distinguished by its focus on allocation within an organisation of individual parcels. This approach is still relatively unexplored as an alternative to the current centralised sorting and routing system. Moreover, quantitative research through simulation and optimisation approaches is also necessary to investigate the performance and viability of this type of system in real cases.

To assess the actual feasibility of this method, three main characteristics are defined that the system must satisfy to be used effectively in real-world systems: acceptable computational complexity, self-organising capabilities, and the ability to handle a dynamic sequence of parcels. The principles of self-organisation in a logistics system are defined by Pan et al. (2016) which are openness, intelligence

and decentralised control. The analysis of the model revealed the limitations of the current method. The difficulty in applying the method results from the lack of built-in heuristics; furthermore, the dependence on input data is a limitation. To benchmark and validate the method. The performance of the model is compared across multiple test instances with the best-known solution and a centralised solver which showed that the considered decentralised base method performs mediocre when considering the distance travelled and computation times. Further validation by use of a parameter variation and an extreme value test showed the irregular behaviour of the number of reassignment iterations and the number of kick-outs parameters. Therefore, reducing the possibility to validate the model and the results, as path dependency could be present.

To address the limitations present, the model is modified with alternative heuristics. The applied modification strategies are random insertion, parcel swapping and k-means clustering. To take into account the behaviour of the model under different conditions, the method is tested for a heterogeneous fleet of vans and cargo bikes and different levels of congestion. Applying the heuristics for the test instance of random insertion and k-means clustering shows better performance in terms of distance travelled and calculation time. Swapping did not yield any benefits in terms of performance and efficiency. Therefore, the combination of random insertion with a search frequency of 50% and the clustering method in a combined approach was chosen. The results of the combined method are better than those of their separate applications in terms of performance for travel distance and calculation time. The combined method was able to achieve significantly better computation times with up to 40% faster run times for the test data. In general, the heuristics were able to improve the performance of the model, but the improvements in travel distance were minimal. When the combination of the best performing heuristics is applied to the case data, a 4.6% improvement in vehicle travel distance can be achieved compared to the original method, but still with high computation times. Comparison with an external centrally organised OR solver shows that although the decentralised method can be improved, it does not perform well, as the OR solver can find a better solution in a fraction of the time needed by the improved decentralised auction method.

The simulation environment is adapted with a heterogeneous fleet, both for the basic method and for the improved method. The mixed fleet changes the simulation environment of the algorithm with respect to the implementation of different vehicle types with lower vehicle speed and capacity. The fleet is adjusted for three types of combinations of different amounts of vans and cargo bikes in the test case and the case data. It was found that for a mixed fleet with similar capacity, the model is no longer able to deliver all parcels with a large number of cargo bikes in the fleet. In general, it was found that for a higher number of vehicles with lower capacity the routing is less efficient. On the other hand, the integration of the cargo bikes shows that for the test case and the case data, the emissions decrease with a higher number of cargo bikes. This indicates a trade-off between efficient routing and greener transport. The simulation environment is also adjusted for congestion. Congestion is a common disturbing factor in the daily last mile delivery. To take congestion into account, three congestion factors are chosen. Congestion affects the time pressure a vehicle has to accept a package. More congestion means less chance for a vehicle to include a parcel in its route. For the original solution method, the model is not affected by the inclusion of congestion. For the modified solution method, congestion causes a slight increase in travel distance. Emissions increase proportionally to congestion, due to the direct relationship between distance travelled and emissions. For the case data, the inclusion of congestion has a larger effect on the travel distance and the associated emissions. The higher the congestion, the less efficient the routing. In contrast to the mixed fleet, the change in the simulation environment by including congestion does not affect the ability of the vehicles to deliver all packages in a feasible way. When taking into account the combination of congestion and a mixed fleet, it can be concluded that the modified method in this study produces more infeasible results for the test case than the original method. Moreover, it can also be observed that for the same fleet combinations with more congestion more parcels are not allocated, both for the test case and for the real case.

The proposed system would represent a radical change from the current way of doing business and handling parcels. Overall, the method is not comparable to other (centralised) approaches in terms of efficiency, which raises the question of whether such a system is desirable. If the method is to compete with the current centralised method, a more efficient model must be developed. The steps for the development of the model would be to structure the research and development by creating a clear vision and scope, defining the requirements, evaluating the limitations, identifying the key partners and establishing step-by-step steps for the actual development of such a system. This study has contributed to the first three steps. It has demonstrated the proof-of-concept and created a more efficient model. In addition, quantitative research has been carried out through simulation and optimisation, and the performance and viability of this type of system have been examined at real scale case data.

The assumption of complexity is evident in real-life logistical processes. Studying a decentralised solution method for parcel distribution helps in understanding these complex systems and points to possible intervention possibilities. It can be argued that the current method of decentralised dynamic parcel distribution is not a good method for the distribution of last mile parcels. The advantages that a decentralised dynamic approach through individual parcel auctions could bring are outweighed by the low efficiency of the model studied in this study. At the academic level, many papers point to the benefits of self-organising principles. It is widely believed in the literature that decentralised control can help improve the robustness and performance of systems. The gap in research not addressed in this study is the relationship between the efficiency of a model and its robustness. Further research should focus on uncovering this relationship to better demonstrate the potential of this type of method.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

In recent decades, we have seen increasing industrial production and corresponding logistic demands. In the European Union alone, more than 1 million companies are registered in the transport and storage sector, employing nearly 11 million people (Winkelhaus and Grosse, 2020). Production and logistics systems have matured, but new customer demands are putting pressure on logistics systems, turning them from complicated to complex (Winkelhaus and Grosse, 2020). One of the drivers is the direct delivery of packages ordered by consumers. Logistics service providers are faced with the challenge of efficiently delivering millions of packages to customers' homes. Numerous companies have gone bankrupt due to operational and logistical problems related to this last mile of transport (Boyer et al., 2009). Last-mile delivery can be defined as freight transport over the last part of the route to the customer using the last means of transport (Macioszek, 2017). Problems related to last-mile delivery usually arise because deliveries consist of individual orders and are spread across destinations (Macioszek, 2017). These inefficiencies make the last-mile delivery the least efficient stage of the supply chain, accounting for 28% of total delivery costs. Moreover, externalities such as congestion and pollution, occur especially in urban areas (Ranieri et al., 2018). Trends of improvements in this area can be classified into the optimisation of transport management and routing, innovative vehicles, collaborative and cooperative logistics, and innovation in infrastructure (Ranieri et al., 2018). One of these innovations in transport management and routing is the application of self-organising principles in logistics. Self-organisation is a term widely used in many research fields (Pan et al., 2016). The application, architecture of control, and level of self-organisation differ greatly. The definition of all distinct concept systems have one thing in common, there is no external control (Hrabia et al., 2018). Moreover, it is overall hypothesised that decentralised control can help to improve the robustness, resilience, and performance of systems (Zhang et al., 2016, Serugendo et al., 2003, Pan et al., 2016, Van Duin et al., 2021, Hrabia et al., 2018, De Wolf and Holvoet, 2007). The principles of self-organisation in logistics systems relate to openness, decentral control, and intelligent of the agents in the system (Pan et al., 2016). For the paper by Van Duin et al. (2021), which evolved from the thesis of Vlot (2019), it is hypothesised that a system in which parcels can choose their own route in last-mile delivery is more robust and could obtain better results in terms of vehicle distance travelled, delivery times, and emission rates than the current centralised system. The method uses a semi-decentralised auction-based model. In this study, possible improvements of this decentralised auction-based solution method are explored. The efficiency and performance are studied for different heuristics and for changes in the simulation environment. This study contributes to the benchmarking and validation of this alternative to centralised sorting, allocation and routing for last mile logistics. It examines the improvements and limitations and thereby better defines the potential of this method as a remedy for the growing complexity in last mile logistics.

1.1 System Scope

The parcel allocation method used in Van Duin et al. (2021), was commissioned by the Self-Organising Logistics in Distribution (SOLiD) consortium. This is a partnership between TNO, DPD, University of Groningen, Delft University of Technology, Eindhoven University of Technology and Erasmus University Rotterdam. The collaboration aims to develop a proof-of-concept for self-organisation in logistics. The algorithm is hereinafter referred to as the SOLiD algorithm. The system concerns the last-mile of logistics in parcel handling in the Netherlands. The focus of this study is on a combination of two parts. First, there is the sorting and assignment of the parcels in distribution centres and second, there is the last-mile delivery of the parcels. Both adhere to conceptual problems known in the literature. For sorting and assignment, there are multiple algorithms and methods to allocate parcels and fill a vehicle, section 2.1.2 describes the current sorting process. For last-mile delivery, different heuristics exist for vehicle routing, an elaborate overview of the vehicle routing problem is given in Appendix A. For this study, the SOLiD algorithm is considered which combines both. The scope is set at the dynamic sorting and routing, where a team of vehicle agents must allocate parcel agents for the last-mile delivery from the last distribution centre to the customer. The term self-organising system is a popular expression, but it is opaque to logistics in the sense that it is unclear what requirements a system must meet in order to unlock the benefits that can result from it. The main reason for applying a self-organising system in general is the improved flexibility and robustness of a system. For this study, the umbrella term self-organisation is still used to explore the literature and the potential for logistics, but a more precise classification of the solution method used in this study is a decentralised dynamic parcel distribution method for parcel sorting and vehicle routing. To assess the actual feasibility of this method, three main characteristics are defined that the system must satisfy in order to be used effectively in real-world systems: *acceptable computational complexity, self-organising capabilities, and the ability to handle a dynamic sequence of parcels.* Self-organising capability refers to the principles of openness (agents can enter and exit the system), intelligence (agents can make decisions), and decentralised control (agents can function without central control). This study investigates the possible improvements in terms of efficiency of the decentralised auction-based method for last-mile parcel delivery and tests it for real-time complexity by incorporating congestion and a mixed fleet. The above-mentioned requirements act as a tool to evaluate, benchmark and validate the solution method. The scope of the study is limited to examining the performance and efficiency of the method. The benefits of robustness and flexibility that could result from incorporating a number of self-organising capabilities are not tested. The model is tested for a range of instances and scenarios, before finally being applied to a real-scale case study consisting of 729 customer location points (postcodes) in the Delft region. For the mixed vehicle fleet, both a homogeneous fleet and a heterogeneous fleet with two types of vehicles are considered, namely a cargo bicycle and a delivery van. Figure 1 gives an overview of the system scope.

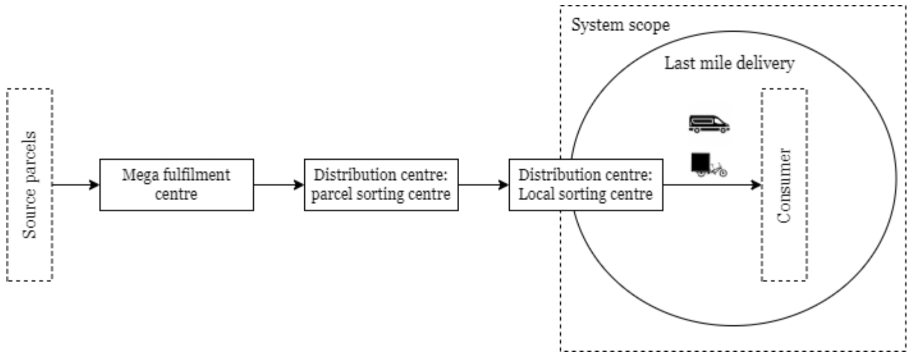


Figure 1: System scope

1.2 Objective and Research Questions

The objective of this study is to evaluate improvements in applying the decentralised single parcel auction method for last-mile logistics. This is done by validating the existing method and proposing an improved version of the solution method which accounts for alternative heuristics, a mixed fleet, and congestion. The insights should provide a critical evaluation of the concept of parcel sorting and vehicle routing by use of the method. The main research question for this study is defined as:

To what extent can the efficiency of a decentralised dynamic parcel distribution method be improved by applying different heuristics and adapting the simulation environment with congestion and a mixed fleet?

The main research question can be answered by dividing it into five sub-questions. The sub-questions are formulated to structurally gather and apply knowledge. The sub-questions are placed in the order of research, namely, analysing the current body of literature on the topic, benchmarking the existing situation, doing experiments to test extension to the model, and identifying the broader implication of the results found.
Sub questions:

1. What is the state of the art of self-organisation and auction methods in last-mile logistics?
2. What is the effect of alternative routing heuristics on the efficiency of the parcel delivery solution method in terms of computation time, and vehicle mileage?
3. What will be the effect of a mixed fleet of vans and cargo bikes for the parcel delivery system in terms of calculation time, vehicle mileage, and emissions?
4. How will the adapted last-mile delivery solution method perform under different conditions of congestion?
5. What are possible other extensions of the decentralised dynamic parcel distribution solution method and what are the steps to implement this technology in last-mile logistics?

The current self-organising logistics system is limited by the assumptions it makes. To validate the current method, moreover, relieve these assumptions, the sub-questions step-by-step examine whether it is possible to include more realistic conditions in the logic of the model. The overview of the state of the art provides the basis for understanding self-organising logistic systems and auctions in logistics systems. This knowledge is then applied to understand the current performance and limitations of the existing model. The model is then extended with alternative heuristics. This is done to see if improvements can be made in terms of calculation time, vehicle kilometres, and emissions. To take into account the behaviour of the model under different conditions, the method is tested for a heterogeneous fleet and different levels of congestion. This gives a better insight into the behaviour of the improved and original model under different conditions of real-time complexity. The extensions to the model should provide further insight into the feasibility of this method and pave the way for further research into this technology.

1.3 Research Approach

The research will be carried out by means of a literature study, modelling and case study approach. The first sub-question 1 will be answered through a literature study. The state of the art in the field of self-organisation in logistics, auction methods and their application possibilities will be studied. The literature study should reveal the gaps in the literature. To ensure the quality of the literature, emphasis should be placed on reviewed papers to obtain information. The required data consists of academic literature that will be consulted using Scopus and Google Scholar. To expand the number of papers, snowballing will be used. The knowledge of the state of the art forms the basis for answering the remaining sub-questions. Moreover, it provides the necessary insights to construct and improve the model for the following sub-questions.

A modelling approach that uses an agent-based model allows for the analysis of the dynamic, multi-actor, multi-objective, and multi-level environment (Van Dam et al., 2012). Therefore, this approach is best suited to study the behaviour of last-mile logistics systems. To create the model, first, the current situation needs to be carefully mapped. The model would consist of a simulation with the interacting vehicle and parcel agents. The main advantage of a model-based approach is that the system can be studied under different conditions (Robinson, 2014). In this way, a comparison can be made between the SOLiD algorithm and extensions to the algorithm. A limitation of the method is that models are always a simplification of reality and should therefore be carefully validated and verified (Robinson, 2014). This approach is used to answer sub-questions 2, 3, and 4. The performance of the model in terms of computing time, vehicle mileage, and emissions will be compared to the performance of the current model to see the effects of the extensions. For each extension, experiments will be carried out that will be compared with the results of the current algorithm working under similar conditions. The modelling approach is applied to a single case study at a company. Prime Vision B.V. would provide access to the data involving last-mile logistics. This provides an opportunity to create a digital environment to test and verify the model based on real historical data. This strategy allows for an in-depth exploration of a complex problem in a real-scale environment (Crowe et al., 2011). In addition, the use of a case study strategy helps to analyse the topic within the boundaries of the organisation (Yin, 2009). The knowledge gained from discussing the topic with industry experts at Prime Vision B.V. is used to answer the last sub-question 5. Here, the possible next steps for the development of this innovation are discussed.

1.4 Methodology

The research flow diagram is designed based on the structure of the research questions and chapters in the report. The diagram of the methodology is given in Figure 2. The chapters provide direction for the research phases. The smaller boxes indicate the main topics addressed per section.

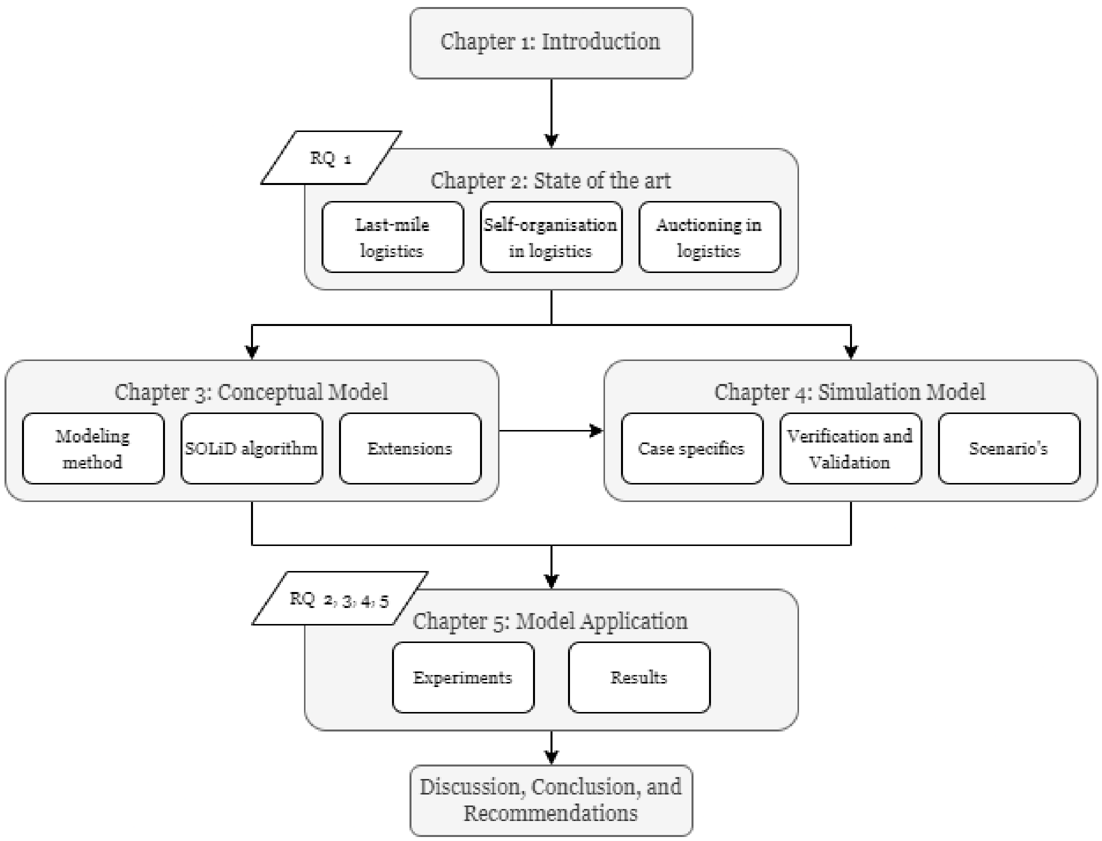


Figure 2: Thesis structure and overview of the research flow

1.5 Previous Work

This thesis is conducted in cooperation with Prime Vision B.V. Prime Vision is a world leader in computer vision integration and robotics for logistics and e-commerce. In its 60 years of existence, it has stimulated innovation in the postal and logistics sector, winning several awards. Prime Vision's goal is to obtain a proof-of-concept for the self-organising capabilities of parcels in logistics. As previous work has been done in collaboration with Prime Vision B.V. in the area of self-organised last-mile logistics, it is important to outline their work and point out the areas that can still be explored and improved. Three of my predecessors have looked at the subject from different angles. Vlot was the first and did extensive research into self-organising parcels and routing. His work resulted in an algorithm for self-organisation that was used in the paper of Van Duin et al. (2021). Valdivia then developed the software architecture for a logistics planning system based on Vlot's algorithm. Finally, Chandrashekar focused on the self-organisation/automation of robots in intralogistics. He thus departed from the concepts of parcel sorting and routing discussed by the first authors.

1.5.1 Distinction of this Study

This study will focus on the continuation of Vlot and Valdivia's work on the application of self-organising principles in last-mile logistics by using the auction-based solution method. This is done by validating and extending the model consisting of Vlot (2019) sorting and routing algorithm. It distinguishes itself from previous work by not classifying the model as fully self-organising, as this might give a wrong impression, but as a solution method that uses decentralised control. This study therefore better delineates the proposed solution method from existing solution methods so that its advantages and limitations can be better compared. The novelty of this study is the exploration of improvements in the decentralised auction method and the analysis of the effects of a mixed fleet, and congestion in the model. A visual overview of the distinction of this work is given in Figure 3.

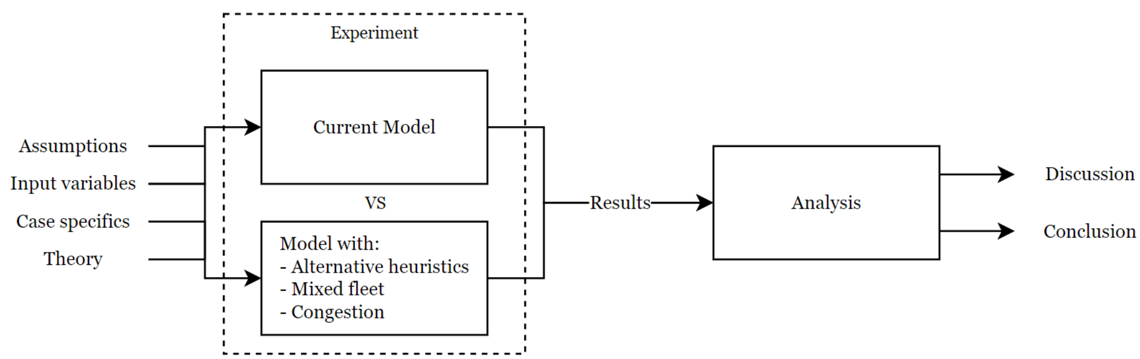


Figure 3: Current solution method compared to the extended solution method

1.6 Research Contribution

The nature of the research problem is the exploration of self-organising principles for the improvement of logistics operations. This is based on the limited knowledge about the effects of decentralised control in logistics and parcel handling. The identified knowledge gap is the lack of understanding of self-organising principles and their ability to improve the performance of parcel sorting and routing. When self-organising principles are considered, the question arises as to why a self-organising network is created. For several domains (logistics, mobility, energy, etc.) the drivers are the increasing complexity and scale of systems and processes, the need for greater efficiency, the need for flexibility and the need for robustness against failure of system components. One consideration that emerges in decentralised systems is between efficiency and greater robustness. Organising all parcels centrally can greatly optimise the delivery system, but it may be less robust against changes in the environment. A decentralised system may be more robust, but less efficient due to the lack of centrally available information. The objectives of design for efficiency and design for robustness are often in conflict (Meepetchdee and Shah, 2007). A better understanding of the efficiency, and thus the feasibility of decentralised logistics, is essential to understanding the potential of these systems. In this way, more insight is gained into the tools for tackling the challenges faced by the constantly evolving logistics sector. This topic relates to the MSc programme Complex Systems Engineering and Management, as the increasing interrelationships and dynamics between different actors in logistics networks lead to highly complex global supply chain processes (Nilsson, 2019). The assumption of complexity is evident in real-life logistics processes. Studying a decentralised solution method for parcel distribution helps in understanding these complex systems and points to possible intervention possibilities. Furthermore, this thesis project deals with the design and analysis of transport systems.

1.7 Thesis Outline

This thesis is structured as follows, this introductory section provides the initial problem analysis and the research framework. Section 2 provides a review of existing literature on last-mile logistics, self-organisation in logistics, this is then translated into a conceptual model in section 3. The implementation of the model is then defined in Section 4. After the model is validated and verified, and the implementation is outlined, the simulation is run for several experiments in Section 5. The conclusion of the research is then given in Section 6, followed by a discussion and recommendation in Section 7.

2 State of the art

This chapter describes the state of the art of last-mile logistics, self-organisation in logistics and auction methods in logistics. The three topics together form the foundation on which this study is based and highlights the research gap present. Furthermore, the current situation is analysed, which will later help in understanding the modelling decisions.

2.1 Last-mile Logistics and Operational Optimisation

At present, traditional brick-and-mortar shops are expanding their online services, pushing the traditional supply chain to new limits with the expectation of excellent service and same-day delivery (Fleischer et al., 2020). The recent times of Covid-19 also show the switch to the more individualistic logistics demand paradigm, forcing logistics networks to become more efficient (Yavas and Ozkan-Ozen, 2020, Ghosh et al., 2021). These changes in behaviour together with growing urbanisation and increased focus on sustainability make last-mile logistics an emerging research area (Olsson et al., 2019). Olsson et al. (2019) divide the literature landscape for last-mile logistics literature in four themes: *emerging trends and technology*, *operational optimisation*, *supply chain structures*, *performance measurement*, and *policy*. The focus of this study can be classified as operational optimisation which focuses on optimising last-mile operations and making better operational decisions. This often employs mathematical modelling and optimisation. The theme of operational optimisation can be subdivided in *routing*, *transport planning*, *scheduling*, and *facility location*. A visualisation of the different themes is given in Figure 4. Routing refers to selecting, planning and finding optimal paths within a network. Transport planning deals with issues such as consolidation, use of spare capacity and load optimisation. Scheduling focuses on planning the sequence of deliveries, and facility location involves finding an optimal location for a facility (Olsson et al., 2019). The problem of decentralised last-mile parcel distribution relates to a combination of routing, transport planning, and scheduling.

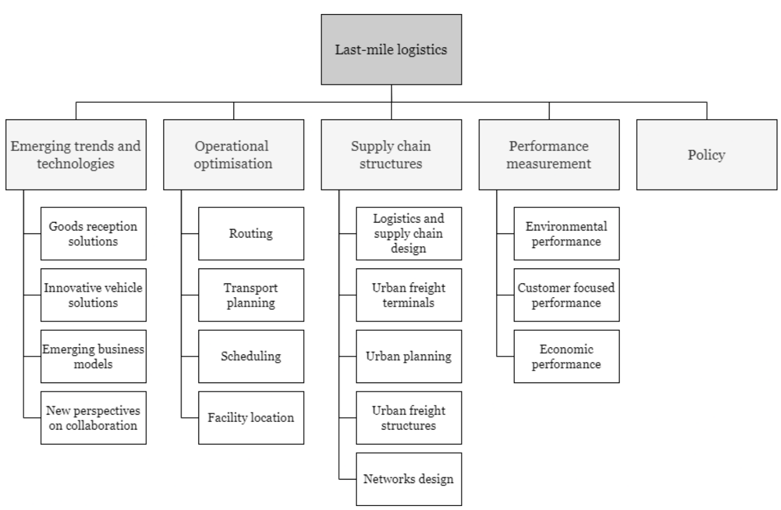


Figure 4: Themes in last-mile logistics literature (adapted from: Olsson et al. (2019))

2.1.1 Computational Complexity

The essence of studying the last mile is the inefficiencies that arise with it. The inefficiencies arise because deliveries consist of individual orders and are spread across destinations (Macioszek, 2017). Operational optimisation for last-mile problems is specifically difficult because many problems that reflect the complexity of today's business environment are NP-hard combinatorial problems (Kang and Lee, 2018). In combinatorial problems, combinations occur in discrete values. One of the best-known combinatorial problems is the travelling salesman problem where the salesman has to visit a series of cities (Buckman, 2018). P-problems is an abbreviation for polynomial problems, which can be solved in polynomial time. NP-problems is an abbreviation for non-deterministic polynomial problems. For a non-polynomial problem, the operational complexity increases exponentially (2^n). Parcel sorting and routing can be seen as a general version of the vehicle routing problem, which has been shown to be an NP-hard problem (Kang and Lee, 2018). When problems become NP-hard, it means that the only way to find the optimal solution is to evaluate all permutations (Buckman, 2018). This number is exponential for the number of parcels and vehicles in the system. Unlike NP-complete problems, it is not possible to validate an optimal solution when it is given. As an aid to finding a solution, heuristics can be applied to guide the search process and do not require brute force search for solutions.

2.1.2 Current Situation for Sorting/Distribution Centres

The parcel delivery industry was worth 500 billion USD in 2019, which grew even more during the Covid pandemic (Ghosh et al., 2021). The parcel delivery system is a complex interconnected and interdependent network. Sorting centres are critical in this infrastructure, as they are the points in the network where incoming parcels are aggregated and then segregated into onward destinations (Ghosh et al., 2021). Logistics service providers are looking for ways to reduce costs and improve customer responsiveness. Cost reduction motivates to centralise inventory, while customer responsiveness motivates to have goods close to the end customer (Nozick and Turnquist, 2001). Distribution centres provide the balance between these two objectives. In the final stage of transportation, it is difficult to combine shipments, this leads to high costs (Macioszek, 2017). Therefore, for sorting the parcels in the distribution centres, it is essential that the parcels are allocated for optimal routing of the delivery fleet. In the e-commerce parcel delivery supply chain Morganti et al. (2014) define five types of facilities, namely, mega fulfilment centres, where goods are stored. Parcel sorting centres, where parcels are sorted before being forwarded to local parcel delivery centres. Local parcel delivery centres for last-mile handling. Local urban logistics depots for fast order fulfilment. And finally, return processing centres. A schematic overview is given in Figure 5. The focus of this work is the delivery from the distribution centre to the customer.

The interview in Appendix B gives insight into the sorting process in the Netherlands. At present, the sorting of parcels and corresponding routing of vehicles is done based on static areas of delivery. To expand on this, each vehicle is assigned to a number of postal codes to which it is set to deliver. Each package has a barcode, which contains the information of the address, delivery time and additional

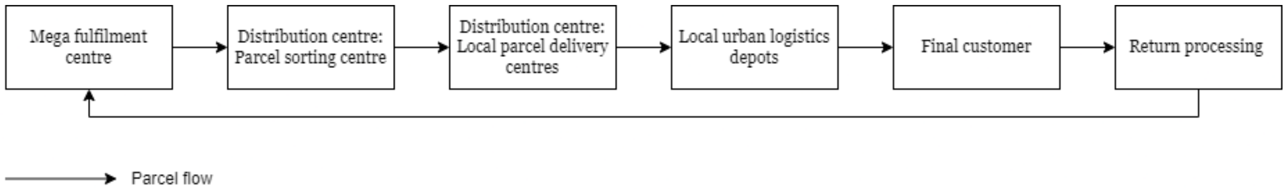


Figure 5: Schematic and simplified overview of the different steps of parcel facilities

services. Most parcels are preregistered, if preregistration is not done, the sorting machine makes a sorting decision by visual inspection of the label. Almost all postal centres in the Netherlands have at least two sorting phases. The first sorting stage is called dispatch sorting and is based on postal code only. The shipment is then sent to a sorting location near the scanned postal code. There, the shipment is sorted again based on the postal code and house number. These postal code groups are often fixed for multiple weeks. The sorting machine scans the packages that pass by on a conveyor belt. The scanner sees the barcode and gives the sorter the correct sorting direction. After being classified, the parcels are placed in roll cages. The maximum number of exists for a sorter is 50, but because current postal services have had to deliver to so many addresses, it is impossible that there is no overlap in postal codes. This is currently being solved by one or more dedicated sorting chutes for the remaining postal codes. Sorting is done in a centralised control system that has all the information of each package at the time it is pre-registered or scanned for the first time. Routes are determined based on this information; this information is then mapped into a centralised system that creates the sorting rules. An overview of the system is given in Figure 6.

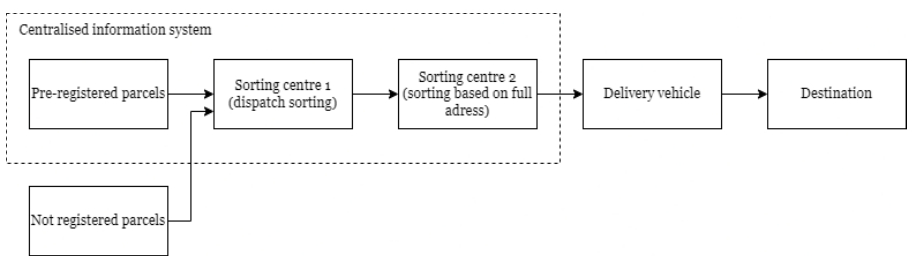


Figure 6: Schematic overview of parcel sorting

Assignment and routing of the parcels are done separately. It first relies on the clustering of postal codes and then a TSP is solved for each vehicle. As delivery zones are often fixed for multiple weeks, daily fluctuations in quantities of parcels can lead to suboptimal utilisation and routing of the vehicles within the delivery fleet.

The method described above is only possible if information is known centrally. This assignment becomes more difficult when not all information is available beforehand. This occurs for parcel handling where the destination of the parcel is not known up to the assignment of the parcel to a delivery vehicle (Phillipson and de Koff, 2020). Considering the dynamic assignment, previous work by Van Duin et al. (2021), uses an auctioning system such that each parcel can decide which vehicle it is assigned to.

Phillipson and de Koff (2020) describe several methods to find an efficient solution to a dynamic allocation problem for vehicle routing. Similar to Van Duin et al. (2021), they propose an approach using minimum cost insertion. The dynamic assignment theory has overlap with auction methods in which assignment is done based on bids of multiple parties. Auctioning methods in logistics and dynamic assignment are further explained in section 2.3.

2.1.3 Transport Modes

For the delivery of parcels along its logistics chain, various modes of transport are used. Long-distance parts of transport are often carried out by trucks, trains or barges (Boudoin et al., 2014). In densely populated areas, such as cities, the road with vans or small trucks is the most traditional mode of transport. However, due to external factors, alternative modes are emerging, such as rail, river/canal boats, and cargo bikes (Cardenas et al., 2017). A visualisation of the possible organisation of such a system with different modes is shown in Figure 7. The emergence of these vehicles is the result of a number of economic and environmental challenges. Urban mobility is responsible for 40% of all CO_2 emissions from road transport and up to 70% of other pollutants (Schliwa et al., 2015). The rapidly growing e-commerce market and the resulting growing number of diesel vans to meet demand do not contribute to this. A possible solution to this is the adoption of cargo bikes in the last-mile logistics fleet. A study conducted by Gruber et al. (2013), shows that 19% to 48% of the mileage of courier logistics could be substituted by electric cargo bikes. Cargo bikes can be classified as 2- or 3-wheeled bikes, with a cargo box, with the possibility of electronic support (Schliwa et al., 2015). One of the main limitations with these vehicles is the limited range they have (Ranieri et al., 2018). Therefore, the main advantage of a van over a cargo bike is its higher capacity and longer range. The viability of implementing different vehicles for last-mile logistics is highly dependent on the geography of the cities and the logistics network. Urban areas with high density and narrow streets in historic city centres are ideal for cargo bikes. Less densely populated and more remote delivery areas argue more for the use of vans (Schliwa et al., 2015).

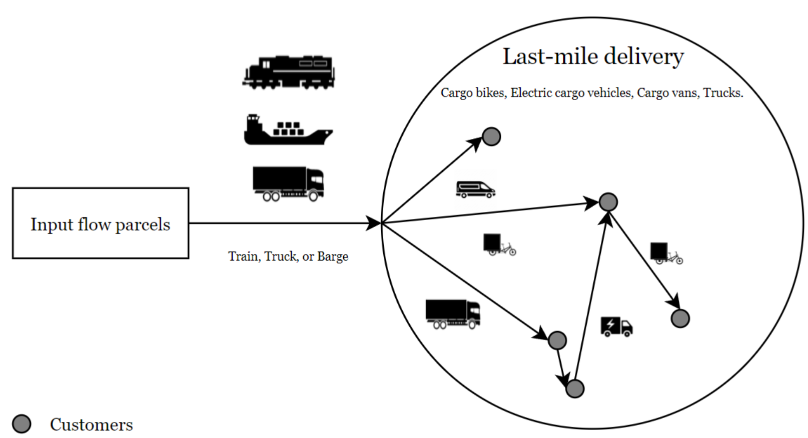


Figure 7: Visualisation of different transport modes for parcels

2.2 Self-organisation in Logistics

Self-organisation is a concept broadly used in many research fields. The term is often used in combination or in overlap with the terms self-adaptation, emergence, and automation. It can be difficult to define the distinction between these three concepts (Hrabia et al., 2018). Appendix C gives an elaborate overview and explains the terms and their dependencies and relationships. When a logistic system is self-organising it can function without significant interventions by managers, engineers, or software control. It functions based on contextual local interactions (Bartholdi III et al., 2010). Today, logistics systems are becoming larger and more complex than ever before. To cope with this problem, numerous concepts and paradigms are proposed as solutions, e.g. intelligent products, holonic and multi-agent systems, cyber-physical systems and physical internet. Self-organising logistics systems can be seen as a joint effort of all these concepts (Pan et al., 2016). In order to achieve self-organising logistics systems, Pan et al. (2016) state that the main characteristics they must fulfil are openness, intelligence and decentralised control.

The concept of self-organisation in logistics is still a developing paradigm, but inspiration is taken from other research domains (Pan et al., 2016). Van Belle et al. (2011), Bartholdi III et al. (2010) and Serugendo et al. (2003) look at a bio-inspired coordination architecture operating system for self-organisation of logistics systems. The principle of self-organisation can be seen in biological processes, where complex organisms can grow from a single cell. A cell can interact with its environment to influence its activities (Van Belle et al., 2011). A commonly used reference in the self-organisation of logistic routing is the network of ants. The self-organising behaviour of the ant population is the result of a network of interactions between the ants and their environment. These can organise the tasks of the ant colony, such as building a nest, finding food, and spreading alarm among members of the community (Serugendo et al., 2003, Van Belle et al., 2011). Ants use pheromones to communicate information along different routes (Van Belle et al., 2011). This principle is used in ant colony optimisation algorithms that can be used, for example, to solve travelling salesman problems (Dorigo et al., 2006) or in cross-docking operations in warehouses (Van Belle et al., 2011). Bartholdi III et al. (2010) apply this approach for the balancing of assembly lines. Furthermore, Serugendo et al. (2003) compare the behaviour of ants routing and food foraging behaviour to a manufacturing control system, in which the different agents communicate with their neighbours to create a system that keeps functioning in a highly dynamic environment. Van Duin et al. (2021) focus on a method of self-organisation specifically for last-mile parcel delivery. Here, self-organisation principles are applied for the allocation of parcels and the routing of vehicles. The self-organisation is facilitated by the integration of an auction algorithm, which allows the parcels to choose their desired path. Another application of self-organisation is by use of a holonic design approach (Bousbia and Trentesaux, 2002, Botti and Giret, 2008). This control architecture is characterised by autonomous and collaborative entities that are often used in manufacturing systems (Botti and Giret, 2008). This concept supports the idea of how self-organising and self-adaption functionalities can deliver cooperative behaviour while providing resilience against the demand and dynamics of the environment (Valckenaers et al., 2008). The application of this architecture is mostly concerned with intelligent manufacturing systems and the allocation and reallocation of resources (Bousbia and Trentesaux, 2002). A trend that gained popularity over the last decade, is the physical internet. This theory applies the analogy of the Internet to structure logistics networks (Pan et al., 2016). Based on the protocols of the digital internet, which is characterised by operational interconnectivity, the physical internet is hypothesised to improve the current logistic models, to solve the challenges of the current global logistic network (Montreuil, 2011).

Most of the literature on both self-organisation and self-organising logistics is conceptual in nature, and relatively little empirical research has been done on it. For practitioners, there is currently no clear direction or course of action for working towards the vision of self-organisation in logistics (Quak et al., 2018). When looking at the decentralised aspects of self-organisation, there is overlap with large multi-agent systems, such as task allocation for autonomous guided vehicles. Decentralised algorithms have made it possible to scale the planners as the teams grow, spreading computation and communication across the robot teams (Buckman, 2018). However, the characteristics of openness (agents joining and leaving the system) are not present in such systems.

2.2.1 Applications of Self-organisation in Different Domains

The concept of self-organisation can be applied in many different contexts. Applications range from energy systems to communication networks. A good example of the application of self-organisation in communication networks is mobile ad hoc networks (MANET). MANETs make use of wireless technologies to connect dynamically without any centralised structure (Hinds et al., 2013). To take into account the dynamic topology of the network, all nodes need to be informed about changes in the network. Conventional information routing methods are inapplicable due to dynamic topology (Gorodetskii, 2012). The main characteristics of these systems are that there is no infrastructure needed as the devices act as network nodes and that the robustness is high. A downside of such a network is the low throughput, as data transfer can only happen at two points. When considering other applications of self-organisation of large-scale systems it is inherently connected to communication structures. Another domain where self-organisation is named is energy systems. An example is decentralised coordination based on self-organisation is used for the improvement of energy resilience. Čaušević et al. (2021) describe an approach to achieve resilience by directing both the physical topology of the grid and changes in supply and demand of individual consumers and prosumers. Additionally, self-organisation can also be observed in computer systems. Serugendo et al. (2003) describe the application of self-organisation in web communities. Authors of web pages place debates on the web with hyperlinks to other pages. The posting of such a web page changes the environment, and in turn, changes the behaviour of other web page authors. These web pages contain specific information for other authors, who will reinforce the strengths between web pages by referring to them. The web communities are thus collectively but independently organised.

2.2.2 Robustness in Self-organising Logistic Systems

A typical objective of a logistics network design is to maximise profits, minimise costs and at the same time satisfy all established constraints. These objectives can be classified as the efficiency of the logistic chain. Logistics networks face contradictory requirements of achieving high operational effectiveness and efficiency while retaining the ability to be robust. Over time, environmental pressures and/or sudden changes may affect the system to such an extent that the entire supply chain may collapse, not to mention have an adverse effect on efficiency. This long-term survival aspect must be taken into account (Meepetchdee and Shah, 2007). The robustness of a system can be defined as the degree to which a system can continue operating despite the existence of errors or malfunctions (Hrabia et al., 2018). The objectives of design for efficiency and design for robustness are often in conflict (Meepetchdee and Shah, 2007). However, systems with higher robustness rather than optimal efficiency are gaining increasing interest (Serugendo et al., 2003). Due to the dynamics of logistic processes, conventional organisational structure can be insufficient in terms of robustness (Berndt, 2011). In the literature on self-organised logistics, it is generally assumed that decentralised control can help to improve robustness (Serugendo et al., 2003, Pan et al., 2016, Van Duin et al., 2021, Hrabia et al., 2018). However, the degree of robustness can vary greatly. For some systems, it is desirable that the system is resistant to failures of parts of the system, while for other systems, delays are the maximum of what can be tolerated.

2.3 Auction Based Logistics

The self-organisation model of the market can be described as an auction in which groups of agents negotiate intending to sell or buy resources. Agents communicate locally by exchanging messages with offers, commitments and payments (Gorodetskii, 2012). An efficient mechanism shares resources between the willing seller with the lowest valuation and the willing buyer with the highest valuation (Ehsanfar and Grogan, 2020). This mechanism design for exchanging resources in a network has attracted interest from a diverse set of communities in literature. In logistics, pricing and auction mechanisms are used for the scheduling of tasks and resources in distributed systems (Wellman et al., 2001, Ehsanfar and Grogan, 2020). For example, Bae et al. (2022) look into auction mechanisms that are used for collaborative freight transport and Gansterer et al. (2020b,a) focus on auctioning of bundles of deliveries. This can also be seen in Van Duin et al. (2007) who developed a sealed multiple auction approach to allocate (peaks of extra) freight to the best offer of a carrier. Moreover, Lee and Kim (2015) look at continuously auctioning single delivery tasks. Giving the carriers the possibility to reevaluate their assigned freight. Furthermore, Van Duin et al. (2021) try to make a connection between the properties of self-organisation and auctioning for the sorting and routing for last-mile delivery. For all, algorithmically, an auction mechanism includes a submission form for participants, an outcome evaluation and a winner selection method (Ehsanfar and Grogan, 2020). Auction mechanisms proposed in the literature include a single-round sealed-bid auction, Vickrey-Clarke-Groves (VCG) mechanisms, market-clearing price, and iterative auctions (Ehsanfar and Grogan, 2020).

Horizontal collaborations in freight logistics have been extensively studied in the literature. Literature for collaborative routing services can be divided into centralised and decentralised planning. In centralised planning, agents have perfect information, whereas, in decentralised planning, limited information is available. For decentralised planning, the method of exchanging requests can be complex; to address this, auctioning is a dominant method in the literature (Karels et al., 2020). Such as combinatorial auctions, which have been shown to be effective mechanisms for establishing collaborations (Gansterer et al., 2020b). For collaborative logistics, combinatorial auctions can be used to exchange transport requests without revealing critical information. This method can yield significantly higher collaborative profits (Gansterer et al., 2020b). The concept of combinatorial auctions is that transport requests are not traded individually, but combined into bundles. A carrier receives the entire bundle of freight if the bid is accepted (Gansterer et al., 2020a). The logic shows similarities with the method of Van Duin et al. (2021), however, they look into individual parcel auctions. These auctions of individual parcels are rarely described in the literature. This while in general clusters affects the performance of single-item auctions in a negative way (Karels et al., 2020).

For Bae et al. (2022), Gansterer et al. (2020b,a), Van Duin et al. (2007), Lee and Kim (2015) the auctioning is used as an optimisation objective function, where different parties compete in (combinatorial) auctions. Meaning that multiple competing carriers try to win the auction. In single parcel auctions, however, the focus is on the benefits of matching shipments for non-competing carriers. Therefore, diverging from existing auctioning literature as all parcels are individually assigned per auction within the same logistic service provider. Thus distinguishing between cooperation and competition. This distinction is also made for the architecture for distributed control for a multi-robot system by Dias and Stentz (2000). The flexibility of a market model within a non-competitive environment allows for the vehicles to cooperate and compete as necessary to accomplish a task, regardless of homogeneity or heterogeneity within the fleet (Dias and Stentz, 2000).

Inspiration can be drawn from other types of auction applications in logistics. A somewhat similar situation is auction-based task allocation for multi-agent systems. In the case of dynamic task allocation, a team of agents is presented with a new, unknown task that must be allocated with their original allocations (Buckman, 2018, Braquet and Bakolas, 2021). This method is focused more on production operations in a job-shop environment (Braquet and Bakolas, 2021). While the individual parcel method of Van Duin et al. (2021) allows for a redistribution of all allocated parcels, Buckman (2018) considers a partial redistribution, in order to allocate new tasks more efficiently. Closely related to task allocation is auctioning in decentralised scheduling where the allocation of resources is auctioned off to competing autonomous agents (Wellman et al., 2001). In contrast to static allocation or scheduling problems, these problems involve dynamic applications. When considering dynamic aspects in vehicle routing, an often-used example is dial-a-ride services that consider transport on demand (Van Duin et al., 2021, Buckman, 2018). The limitations of these problems are that for larger instances, repeating the expensive calculations to obtain routes or allocations becomes a burden on the system and it can no longer return solutions on the time scale of dynamics (Buckman, 2018).

The SOLiD algorithm uses an incremental auction method to (re)allocate each individual parcel. An overview of the sorting situation is shown in Figure 8. As can be seen in the figure, an auction takes place for each unit arriving in the system. The result of the auction depends on the availability of the vehicles and the parcels already in the system. The changing order of the packages makes the system dynamic.

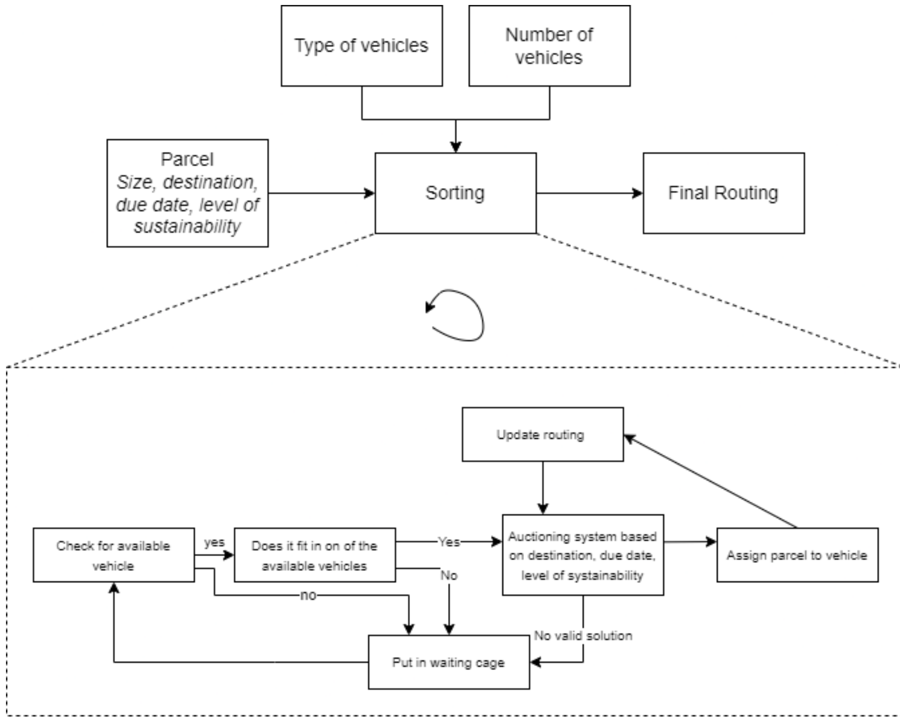


Figure 8: Interpretation of the sorting of self-organising parcels based on the auctioning system proposed by Van Duin et al. (2021)

2.4 Summary

Self-organising logistics systems have only been studied to a limited extent in the literature. The problem of self-organising last-mile delivery is related to several topics of operational optimisation. It shares characteristics with routing problems, transportation planning problems and scheduling problems. The topic of optimisation in this research area is specifically difficult because many problems involve NP-hard combinatorial problems. To overcome this, the real-life sorting and routing system assigns each vehicle to a predetermined static delivery area. As delivery zones are often fixed for multiple weeks, daily fluctuations in quantities of parcels can lead to sub-optimal utilisation and routing of the vehicles within the delivery fleet. Logistics systems that have self-organising characteristics can potentially improve the current system by making it more dynamic and robust. Three main characteristics have been defined for self-organising logistics systems to be self-organising systems: openness, intelligence and decentralised control. Self-organisation in logistics is still a developing field of research, often taking inspiration from other research domains such as biology or communication systems. Most of the literature on both self-organisation and self-organising logistics is conceptual in nature, and relatively little empirical research has been done on them. There is currently no clear direction or course of action for working towards the vision of self-organisation in logistics. The method used in this study also relates to auction-based logistics. Auction-based logistics and horizontal cooperation have been studied extensively. The focus of these studies is on the allocation of bundles of packages to competing logistics providers. The auction system defined for the SOLiD algorithm is distinctive in that it focuses on the allocation within an organisation of individual parcels. This approach, which would allow for decentralised control, is still relatively unexplored as an alternative to the current centralised sorting and routing system.

3 Conceptual Model

Conceptual modelling is the abstraction of a simulation model from the part of the real world it is representing (Robinson, 2011). This section will elaborate on the abstraction and simplifications of the conceptual model used for this study. Figure 9 gives an overview of the artefacts of conceptual modelling as defined by Robinson (2011). The steps are in line with the steps taken in this research. Sections 1 and 2 give the system description, followed by the conceptualisation in this section, and the model design in section 4. This conceptualisation builds upon the model used in Van Duin et al. (2021). The proposed method is validated, improved to be more applicable for real-life instances, and tested for different heuristics, congestion, and a mixed fleet. To better understand the system an initial conceptualisation is done, and then an in-depth analysis of the original algorithm is done in section 3.2, at last, the extensions of the model are explained in section 3.3.

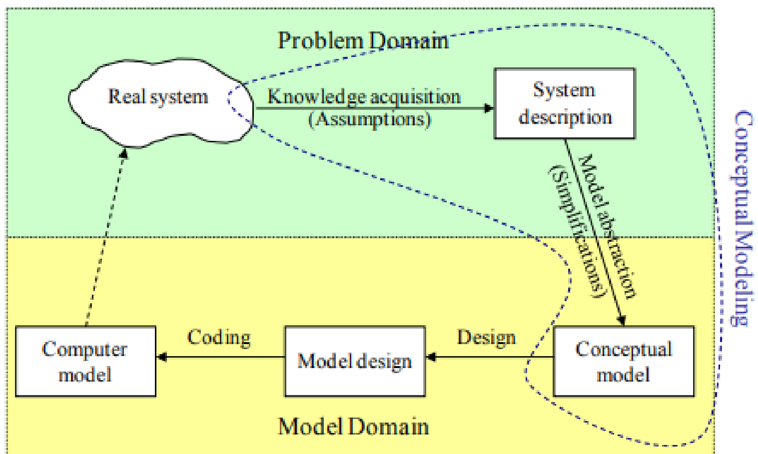


Figure 9: Artefacts of conceptual modeling taken from Robinson (2011)

3.1 General Conceptualisation

Based on the system descriptions, the conceptual model can be made. The conceptual model shows the proposed simulation model to be run in the last mile delivery environment. A visualisation of this is shown in Figure 10. Three categories of input parameters can be defined. The customer demand parameters, the parcel parameters and the vehicle parameters. This is the starting point of the model. Based on these parameters, the model will perform the operational optimisation of the auctioning/matching/sorting/routing problem in question. The optimisation is a combinatorial optimisation problem in which the total utility (based on emission, speed and distance) is maximised. The model itself consists of the interaction between the different agents, the objective function, and the constraints. The objective functions define the optimisation that should take place, and the constraints indicate the limits of this optimisation. The agents can be defined by the parcels, the vehicles and the control platform. The combination of the agents, the objective function and the constraints result in an operational optimisation algorithm. It answers a two-fold question of how the parcels should be assigned to the delivery vehicles and what path should the delivery vehicles follow.

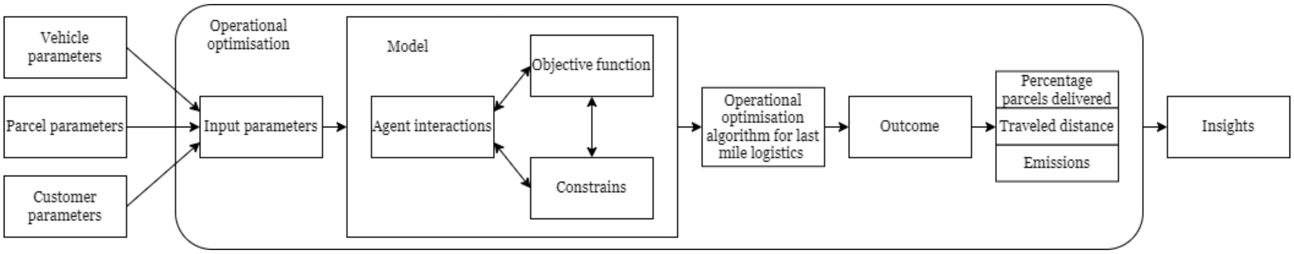


Figure 10: Conceptual overview model

3.1.1 Agents

All agents can be described by the state, rules, and interactions. The state of the agent represents the location of control. A simple example is that an agent can be idle, or busy. The rules of an agent define the actions and the ability to act within the system. The interaction can be explained by the exchange of information between the different agents and their environment. For the model, three agent groups can be defined, namely, the vehicles, the parcels, and the platform. The customer parameters are incorporated into the parcel's characteristics. For example, each parcel has a drop-off location which defines the demand for that parcel at that customer point. The platform agent provides the possibility of matching vehicles and parcels to take place. An overview of the interaction between the agents is given below in Figure 11.

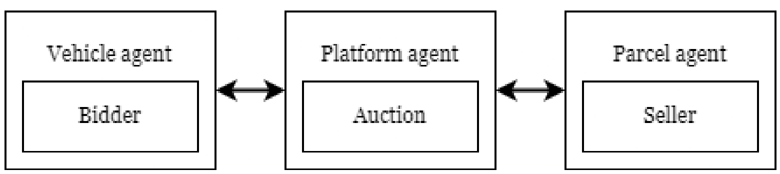
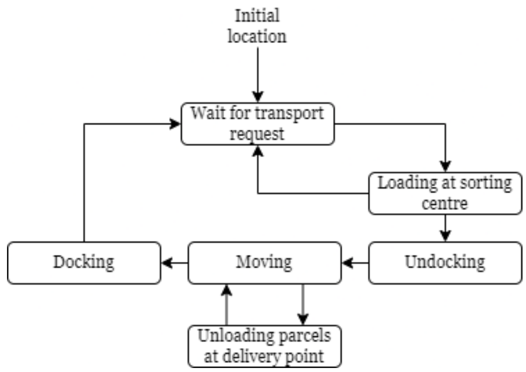


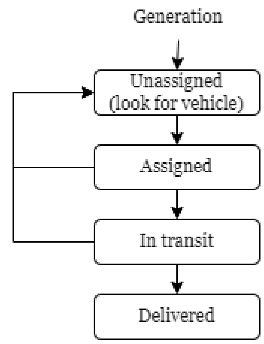
Figure 11: Agent interaction overview

Vehicles

The vehicle agents take the parcel agents from the sorting centre to the desired drop-off locations. All vehicle agents are initially generated at the start of the simulation. Each vehicle is waiting for a transportation request. When a parcel request is accepted the vehicle agent changes its scheduled itinerary. Once the truck's capacity is full, or all parcels are sorted, the vehicle undocks and will go past each delivery point in its itinerary. The state changes from docked to moving. Every time a parcel is dropped off the state changes from moving to unloading. For modelling simplifications, it is assumed that all vehicles of the same type have the same characteristics. Each vehicle has the ability to give an offer for the transport request. Moreover, each vehicle can update its itinerary. A state flow of the vehicle agent is given in Figure 12a.



(a) State flow vehicle agent



(b) State flow parcel agent

Figure 12: State flow of the vehicle (a) and parcel (b) agents

Parcels

A parcel agent represents an unit to be shipped within the capacity of the vehicle agent. The parcels are generated in the sorting centre where they decide upon their vehicle of transportation. The state of the parcels can change from unassigned, to assigned, to in transit, to delivered. In the first three stages, the parcel is able to send out transportation requests to improve its routing possibilities. A state flow of the parcel agent is given in Figure 12b.

Platform

The central platform enables the auction process to take place. It brings together the transport requests of the parcels and the vehicles that are willing to fulfil these requests. The platform is able to place all relevant vehicles in a pool so that the parcel agent can choose the most suitable vehicle. The platform has no states to switch between, it only acts as temporary storage of information. A visualisation of this is given in Figure 13.

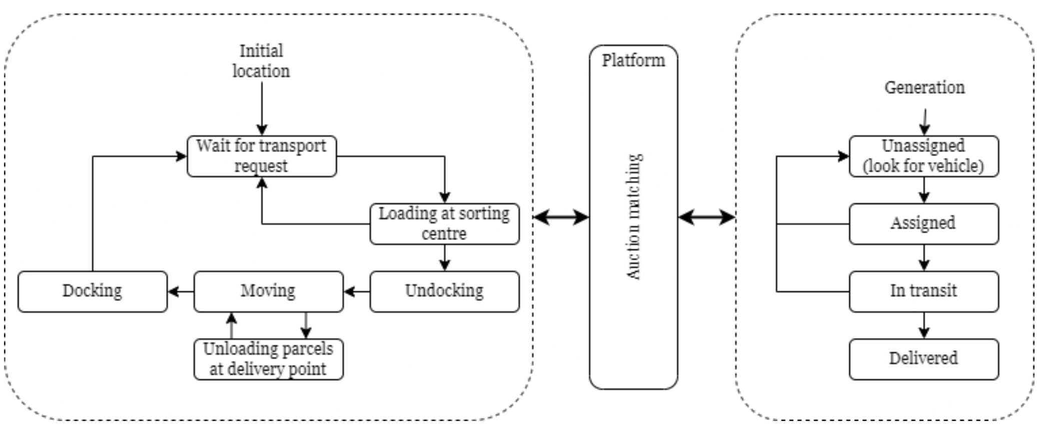


Figure 13: Combined state flows of the the vehicle, parcel and platform agent

3.1.2 Optimisation Strategy

The goal of the proposed method is to optimise the last-mile logistics provider's efficiency while meeting all delivery requests. The optimisation strategy follows an auction process to match the transportation request with the vehicle's capabilities to deliver. To obtain efficiency maximisation an objective function is proposed in the bidding process. In the objective function of each auction session, two utility parameters are considered; delivery speed and emissions, both derived from the travel distance of each vehicle. To reflect the customer's preference (speed or low emission), a general cost function with the value of emission (VOE) and the value of time (VOT) is used (Vlot, 2019). The algorithmic procedure of the auction methods comprises four steps. The first step is to request a transport auction. The second step is that the vehicles bid their best transport bid. In the third step, the bid is evaluated based on the equation described in equation 1. The fourth step is that a vehicle wins the auction and adds the parcel to its route. As the system is decentralised, it relies on local minimisation of the cost function for each bidding round. To describe the objective function first the notations are given.

Notations:

V is the set of vehicles in the auction, indexed by v

R is the transport request being auctioned, indexed by r

d_v^r is the additional distance (m) of a request r for vehicle $v \in V$, $r \in R$

p_{var}^v is the variable operating price (€) per kilometre of vehicle for vehicle $v \in V$

p_f^v is the fixed operating price (€) for vehicle $v \in V$

C_v^r is the operating cost (€) of a vehicle v for a request $r \in R$, $v \in V$

α^{time} is the VOT

T_r^v is the leadtime (sec) (until final delivery) of the parcel request $r \in R$ for vehicle $v \in V$

$\beta^{emission}$ is the VOE

E_r^v the CO_2 emitted (gr) per kilometre for vehicle $v \in V$

K_v capacity of vehicle $v \in V$

K_v^{max} maximum capacity of vehicle $v \in V$

T_v^{max} maximum delivery time (sec)

The cost function per parcel request is given by equation 1:

$$C_r^{Generalised} = C_v^r + \alpha^{time} * T_r^v + \beta^{emission} * E_r^v * d_v^r \tag{1}$$

The operating cost C_v^r of vehicle v for request r, is defined by equation 2. The transport costs consist of a variable cost per kilometre and a fixed operating cost.

$$C_v^r = d_v^r * p_{var}^v + p_f^v \tag{2}$$

Each vehicle bid will be evaluated based on the cost function 1 for each request. In order to chose the winning bid, a simple winner determination program needs to take place. The winner determination program is formulated as minimising the $C_r^{Generalised}$ for all vehicles while $K_v < K_v^{max}$ and $T_r^{lead} < T_v^{max}$.

3.2 The SOLiD Algorithm

The algorithm by Vlot, used in Van Duin et al. (2021), is done in an assignment for the Self-Organising Logistics in Distribution (SOLiD) project. In the proposed algorithm a parcel requests transport from vehicles based on its preferences. The algorithm allows for decentralised decision-making by each parcel through a bidding system, and it also considers the transfer of parcels between vehicles. The parcel requests transport, and then the vehicles send bids, which are then accepted based on the preferences of the parcel. The parcel can send several transport requests along its route, evaluating whether a transfer is desirable. If these requests are accepted, the parcel can change vehicles, making the system hypothetically more robust than current centralised systems. An overview of the system can be given by the UML sequence diagram in Figure 14. In order to understand the system, the following sections will give an overview of its various functions, the input data, the auction method, and possible extensions.

3.2.1 Functions

The algorithm consists of the main file and 19 functions. In the main file, the simulation is performed, which consists of a number of iterations during a certain time frame, representing the handling of the parcels during that time frame. For this purpose, the different defined functions are used. Table 1 contains an overview of each function and an explanation. An overview of the interrelationships of the different functions is visualised in Appendix D in Figure 35.

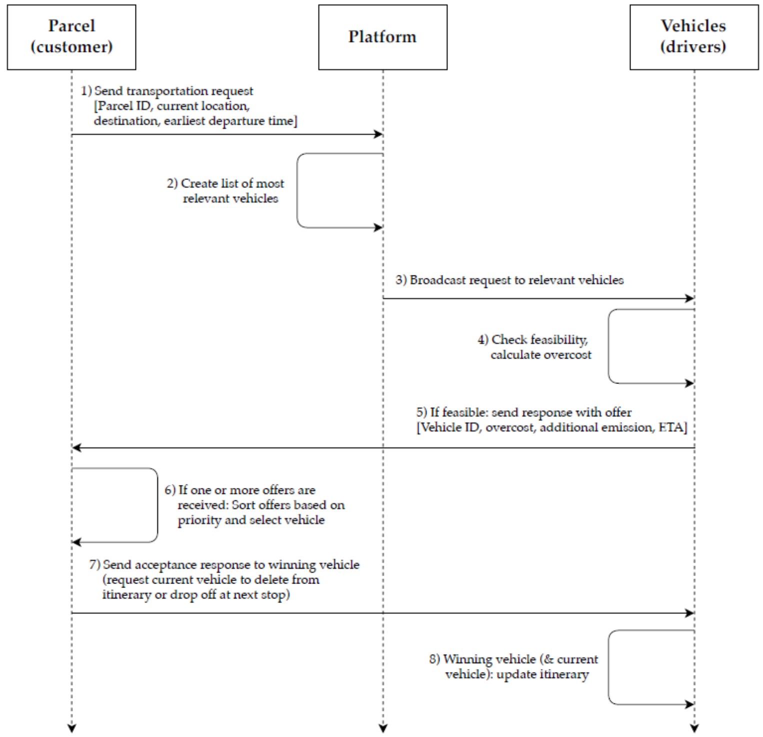


Figure 14: UML sequence diagram of the auction-based sorting algorithm taken from Van Duin et al. (2021)

Table 1: Algorithm Function Overview

Functions	**Explanation**
Main	Model the last mile logistics of the parcels
findDistance	Loads in text file with distance matrix and finds distance between location A and B
removeParcel	Removes the pickup and delivery of parcel in a vehicle's itinerary
insertOffer	Inserting pickup and delivery into vehicle's itinerary.
findRelevantVehicles	Determines which vehicles are relevant to make an offer
makeOffer	Creates offer a vehicle can make to respond to a transportation request
calculateETA	Calculates the time of arrival given an vehicle's offer
checkTimeLimit	Checking if time limit is not exceeded
checkCapacityLimit	Checking if capacity limit is not exceeded
rescheduleDelivery	Removes pickup and delivery of a parcel in vehicle's itinerary
calculateIterationStats	Calculates the KPI of the system after each iteration
calculateKPIs	Calculates the KPI of the system, parcel agents, and vehicle agents
removeParcelKickOut	Removes the pickup and delivery of a parcel in a vehicle's itinerary
removeDummyDestination	Removing a dummy destination from a vehicle's itinerary
kickParcels	Removes worst preforming parcel from the truck based on the distance from the delivery centre
updateStatus	Updates status of the parcels and vehicles
atTransshipmentLocation	Checks if a parcel is located at a transshipment point
assignedParcels	Organising the transportation request auctions for parcels that are assigned
unassignedParcels	Organising the transportation request auctions for parcels that are unassigned
enRouteParcels	Organising the transportation request for parcels that are en route. Calculates the current ETA, emission, and overcost, and than compares it to the new offer.

The UML in Figure 14 helps to understand the basic sequence of the code. It is, however, too simplified in understanding the sequence of all functions. To understand the interrelations and the working of the algorithm an extensive overview is created. This overview is given in Figure 15.

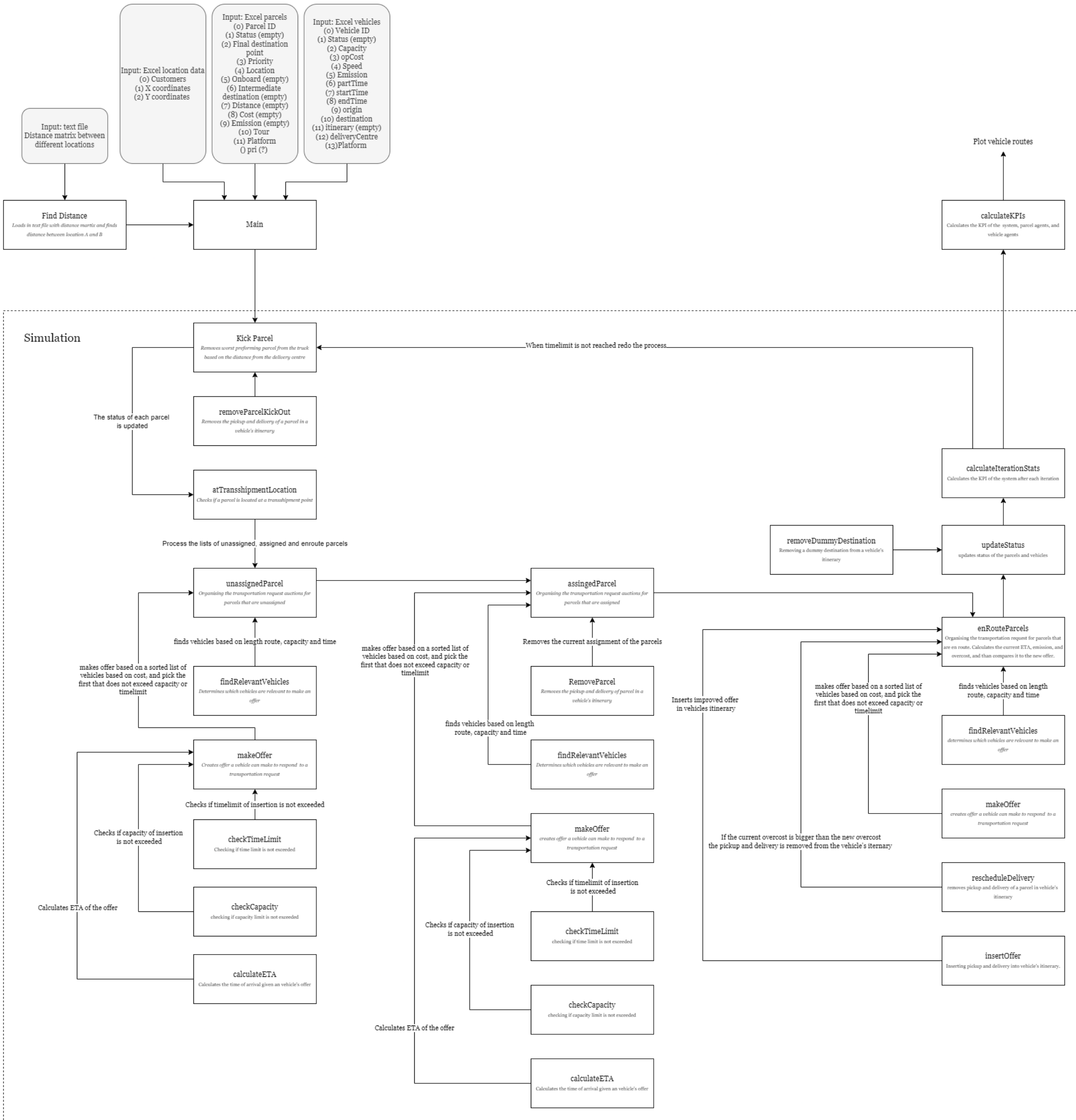


Figure 15: Diagram to illustrate the sequence of the SOLiD algorithm based on the functions used in the algorithm

This overview provides a visualisation of the algorithm. Four input files are used to start the code, which is shown on the top left. When analysing the code, it is important to note that the functions in Figure 15 that are in the dotted simulation box are executed iteratively for each time step. The KPIs are calculated when the stop time is reached. At the beginning of each time step (excluding the first time step), a predetermined number of worst-performing parcels are removed from the vehicle, if they have already been assigned. This is done on the basis of the distance to the delivery centre of that vehicle. These removed delivery and pick-up locations are then removed from the truck's itinerary. Next, a check is made whether parcels are at a transfer point and the lists of unassigned, assigned, and in transit, parcels are updated. The next part of the algorithm can best be described if it is divided into three parts. Namely, depending on the status of the parcel; unassigned, assigned, and en route.

- For an unassigned parcel, first all relevant vehicles are selected based on the length of the route, the capacity of the vehicles, and the time frame. Then an offer is made, this offer consists of a sorted list of available vehicles based on the delivery costs. The first vehicle that does not exceed the capacity and time limit is selected.
- For an assigned parcel, a similar process is carried out, and the whole process of the unassigned parcel is carried out again. So finding relevant vehicles and making an offer based on the available vehicles, cost, capacity and time limit. To understand why assigned parcels go through a similar process as unassigned parcels, it is important to know that the system depends on the order in which the parcels arrive. Each arriving parcel is assigned to a vehicle, but this may no longer be optimal when there are more packages in the system.
- For an en-route parcel, a different process is performed. First, the relevant vehicles are identified. Then, a sorted list is drawn up based on cost, time limit and capacity. Then the overcosts for the changeover is calculated and if the current overcosts are higher than the new overcosts, the pick-and-deliver is removed from the vehicle route. The improved offer is then included in the vehicle route.

After the bidding process and allocation is done for each parcel for each status (unassigned, assigned, en route) for every iteration. The algorithm provides the routes of the vehicles and the KPIs of the system for the given input values. The method of the current assignment of parcels is similar to a Dynamic Assignment Vehicle Routing Problem. In this VRP the parcels arrive at a location in a certain order, each parcel reveals its destination on arrival and has to be assigned immediately to one of the vehicles (Phillipson and de Koff, 2020). This happens in the first allocation round when the parcels are not yet allocated. The algorithm assumes that in the following steps there is still room to reassign a parcel after the first assignment. This means that parcels can be moved after they have been assigned to a vehicle.

3.2.2 Input Data

The input data used consists of three excel sheets and a text file. An Excel sheet with parcel data, an Excel sheet with vehicle data, an Excel sheet with predefined delivery point data, and a text file with the distance matrix between all delivery points. The origin-destination matrix is calculated in advance on the basis of Euclidean distances. A detour factor of 1.5 is used for all distances (Vlot, 2019).

The parcel file exists of the parcelID, the status, final destination of each parcel, priority, start location, onboard (empty), intermediate Destination (empty), distance (empty), cost (empty), emission (empty), tour, and platform. The vehicle file exists of the Vehicle ID, the status (empty), capacity, operation cost, speed, emission, partTime, start time, end time, origin, destination, itinerary (empty), delivery centre, and platform. The empty variables are added to an list in the algorithm and constantly updated during code execution. An overview of the variables used as input and an overview of the variables that are predefined or updated can be found in Table 14 in the Appendix E.

As mentioned above, the delivery centre for each vehicle is determined in advance. In the method of Vlot used in Van Duin et al. (2021), the preferred delivery centre is chosen randomly. The delivery centres play a role in assigning a parcel to a vehicle. Bidding is based on the distance that a vehicle must travel. By assigning a delivery area to each vehicle, the algorithm mimics the current situation in which vehicles are assigned to a fixed set of postal codes. However, the random selection of delivery areas may lead to a situation where vehicles' routes overlap. Phillipson and de Koff (2020) use a K-means clustering method to determine the delivery centres in the case of the dynamic assignment vehicle routing problem. A similar approach could be used to improve the current algorithm.

3.2.3 Auctioning Method

The auction process of the SOLiD algorithm is best described by the make offer function and the unassignedParcel function of the algorithm. The pseudocode is given in Algorithm 1 and 2. The offer part of the auction in Algorithm 1 consists of an offer for insertion in the itinerary of an available vehicle. Each transport request from the parcels is considered for inclusion in the routes. After which a list is made based on possible insertions which are then sorted based on cost (the cost function is explained in more detail in Section 3.1.2). It is then checked whether the time or capacity limit is exceeded. The best insertion in the travel route is selected from the sorted list (based on cost). This offer is then sent to the parcel that sent the transport request.

Algorithm 1 Auction (makeOffer) in SOLiD algorithm (modified from: Vlot (2019))

```
Require: Offer for parcel transport request
  for Each point of pickup insertion do
    for Each point of delivery insertion do
      if Pickup time <= delivery time and pickup time >= departure Time then
        Calculate cost of pick up and delivery
        Create list of possible insertion combinations
        Sort possible insertion combinations based on summed cost
      end if
    end for
  end for
  for Each insertion combination do
    if If arrival time at destination > endtime delivery window then
      Send no offer
    end if
    if Timelimit not exceeded then
      if Maximum vehicle capacity is not exceeded then
        Select best insertion
        Create offer for parcel
      else if Maximum vehicle capacity is exceeded then
        Send no offer
      end if
    end if
  end for
```

The next step in the auction process is to accept the best bid. This is done in the algorithm 2. First, the availability of vehicles is checked. All vehicles available make a bid using the MakeOffer function described above. For each bid in the list, the generalised cost is then calculated according to the priority of the package (speed or emission). The list is then sorted by cost, speed, and then emissions. The best choice is selected, and the parcel is assigned to that vehicle.

Algorithm 2 Auction (unassignedParcel) in the SOLiD algorithm

```
Require: Organising the transportation request auction for unassigned parcels
  for All vehicles do
    if None available then
      Pass to next transportation request
    else if Vehicles available then
      for All available vehicles do
        Make offer                                  ▷ Function described in algorithm 1
        if Offerlist is empty then
          Pass to the next transportation request
        end if
        for Each offer in the list do
          Calculate the generalised cost based on the priority and corresponding value of time
  (VOT) and value of emission (VOE) (see equation 1)
          Sort offer list based on cost, then speed, then emission
        end for
        for Sorted offer-list do
          Pick the first on the list
          for Selected offer do
            Insert selected offer in itinerary
            Add distance to vehicle's KPIs
            Set the status of the parcel to assigned
          end for
        end for
      end for
    end if
  end for
```

3.2.4 Model Limitations

The research done by Vlot (2019) is extensive, but there is still room for improvement. To improve the current model, it is important to identify the areas of improvement. Vlot (2019) named several model extension in his study which are described in Appendix F. In addition to the limitations named by Vlot (2019), more areas of improvement were found while analysing the algorithm:

- *Applicability on real-life instances:* One of the main limitations of the algorithm is the calculation time, even for a small dataset of 80 parcels with 11 vehicles for 660 iterations (7 hours in simulation) it takes more than an hour to execute the code (Processor: Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz, RAM: 16.0 GB). This happens because all calculations are performed sequentially. As the number of parcels increases, the model's calculation time increases significantly. To clarify, for each parcel that is (re)allocated, the allocation of all other parcels is recalculated. A vehicle routing problem is an NP-hard problem, which means that the solution time required increases significantly with the size of the problem. Heuristics are therefore of great importance. In his study, Vlot (2019) mentions that, for the model, a random insertion heuristic is first performed to create the initial solution. Then a two-opt improvement heuristic is used to randomly swap route segments (1000 iterations per vehicle), guided by a simulated annealing metaheuristic. Finally, a random insertion heuristic is used to remove the last quirks in the vehicles' routes. The given algorithm does not incorporate these heuristics. The pseudocode 1 shows that the insertion for the given vehicle is not done for a random set of insertions, but for the complete route of each available vehicle. Furthermore, the two-option improvement has to do with the assigned parcels changing vehicles, but also for this all possibilities are calculated instead of a random number of parcels. The simulated annealing meta heuristic described can not be found in the functions creating the algorithm, as also can be seen in the overview in Figure 15. To summarise, the current algorithm runs a full enumeration. An improvement could be the implementation of heuristics to reduce the computation time and applicability on real-life instances.

- *Dependence on input data:* All parcel information, coordinates, distance matrix, and vehicle characteristics have to be predefined for the model to run. All the start and end times of the vehicles must be predefined in the input data, which means that the model can run only during the time frame that corresponds to the input data. This has to be logically aligned beforehand, to have the algorithm running smoothly. An interesting extension would be to have agents that are to some extent autonomously join the system.

- *Dynamic demand:* Currently, all customer points are predefined and static. It would be interesting to include a full range of drop-off points where demand varies over a longer period.

- *Replenishment of delivery vehicle:* In the current model it is not possible to have vehicles restocked at the distribution centre. An interesting extension would be the ability to restock the vehicles. Making simulation over longer periods possible.

- *Congestion:* For the parcels to be exchanged between the vehicles, it is essential that both vehicles are at the same point to transfer the parcels. Moreover, the delivery routes are dependent on the actual delivery times of the vehicles. To test the behaviour of the system under real-time complexity, congestion can be included in the model. Both to validate the method under different conditions and to make it more realistic.

- *Clustering:* For the current method, according to the report of Vlot (2019), random delivery centres are chosen for each vehicle in the validation phase and delivery zones are chosen in the case experiments. However, when testing for different random delivery centres, overlapping routes occur. Furthermore, the performance of the system based on the calculated KPIs decreases. An improvement could be to add k-means clustering. However, this reduces the dynamic behaviour of the model, since all parcel locations must be known in advance.

- *Heterogeneous fleet:* The current model only considers the use of vans in the base fleet. An interesting extension would be the integration of different vehicles to look at the effects and behaviour between agents with different characteristics.

- *Iterations to simulate time steps:* For the current code, the entire code is looped for each timestep, which is set to one minute. This means that 660 iterations represent 7 hours (1 workday). The current algorithm is unable to distinguish a minimum time frame for the delivery of the parcels. To clarify, when the code is executed during one time step, the KPIs for distance travelled and parcels delivered still give results as if a full run (1 workday) had been performed. In other words, time steps are currently only used to limit the number of insertions (exceeding the pick-up and delivery time) and to announce the availability of a vehicle.

- *Incorporation of more stochastic elements:* Currently, the algorithm applies only one random seed used to shuffle the list of parcels on arrival. It would be interesting to include more stochastic elements in the model.

Given these limitations, it is important to define that part of the novelty of this study lies in a better understanding of this type of model and its applicability to real cases. One of the contributions is the validation of this existing method and the evaluation of this decision-making tool. The next section will extend this by testing the method for a range of instances. In addition, the model is extended to address some of the current shortcomings, potentially bringing the method one step closer to implementation.

3.3 Model Integration

Now that the general conceptualisation of the model and the analysis of the existing algorithm have been carried out, the model integration can be defined. The current model has proven that the applied method can give better results for total vehicle distance, operational costs and carbon emissions (Van Duin et al., 2021). However, analysis of the model has shown that the application of the model in practice has its limitations. It was discussed in Chapter 1.1 that in order to be used efficiently in real-world systems, the model must have acceptable computational complexity, self-organising capability, and the ability to handle a dynamic sequence of parcels. In order to realise these capabilities, the model is adapted with different heuristics, heterogeneous fleet and congestion. The different heuristics help to reduce the computational complexity but may affect the performance of the model. The changes in the simulation environment by adding a heterogeneous fleet and congestion test the behavior of the model under real-time complexity.

3.3.1 Heuristics

The current method runs an enumeration of all possibilities of (re)assignment of the parcels within the constraints of the capacity of the vehicle and the time frame of delivery. It can be said that it uses brute force to come up with a feasible solution. In order to effectively use the algorithm, heuristics need to be incorporated. The algorithm does to some extent apply a greedy algorithm as it chooses the best option within the available list of offers. However, the number of available offers that are calculated can be limited. The heuristics to be applied and tested are a random insertion heuristic, a two-opt swap heuristics, and a k-means clustering heuristic to find the delivery centres. A visualisation of this can be seen in Figure 16.

3.3.2 Heterogeneous Fleet

The heterogeneous vehicle fleet should provide insight into the effects of different vehicle characteristics. Literature shows that the use of cargo bikes is a growing trend for last-mile delivery. To include cargo bikes in the basic fleet, adjustments have to be made for vehicle speeds, capacity and fleet capacity. In addition, elements such as maximum distance travelled need to be taken into account. This will result in additional constraints on vehicle availability (e.g. maximum distance based on a time constraint) and a change in input variables.

3.3.3 Congestion

Congestion is a common phenomenon in last-mile logistics. It adds a layer of complexity to the system. The implementation of congestion mimics the existence of system failures. Testing different congestion factors help to understand how the system behaves under these environmental changes. The algorithm uses time iterations of minutes. For each pick-up, delivery and redistribution of a parcel, this time iteration is used to determine and limit the possibility of redistribution of parcels. The assignment and routing of parcels can thus be influenced by congestion. The current system claims to perform better than a system with fixed clusters. It would be interesting to see how congestion affects the self-organisation of the parcels when congestion occurs. Moreover, it could show the advantages of using a heterogeneous fleet, as congestion affects vans more than cargo bikes. Congestion can initially be modelled by changing vehicle speeds. This is the simplest way is to simulate the congestion of each vehicle.

3.4 Summary

The model consists of a combinatorial optimisation problem where distance and emission are minimised. The simulation itself consists of an interaction of agents that exchange information about the allocation and routing of a parcel. Three agents can be defined: vehicle agents that make transport offers, parcel agents that request transport, and a platform agent that reconciles the offer and the request. The analysis of the algorithm led to a comprehensive overview that can be seen in Figure 15. It also revealed the limitations of the current method, including the difficulty of applying the model to real

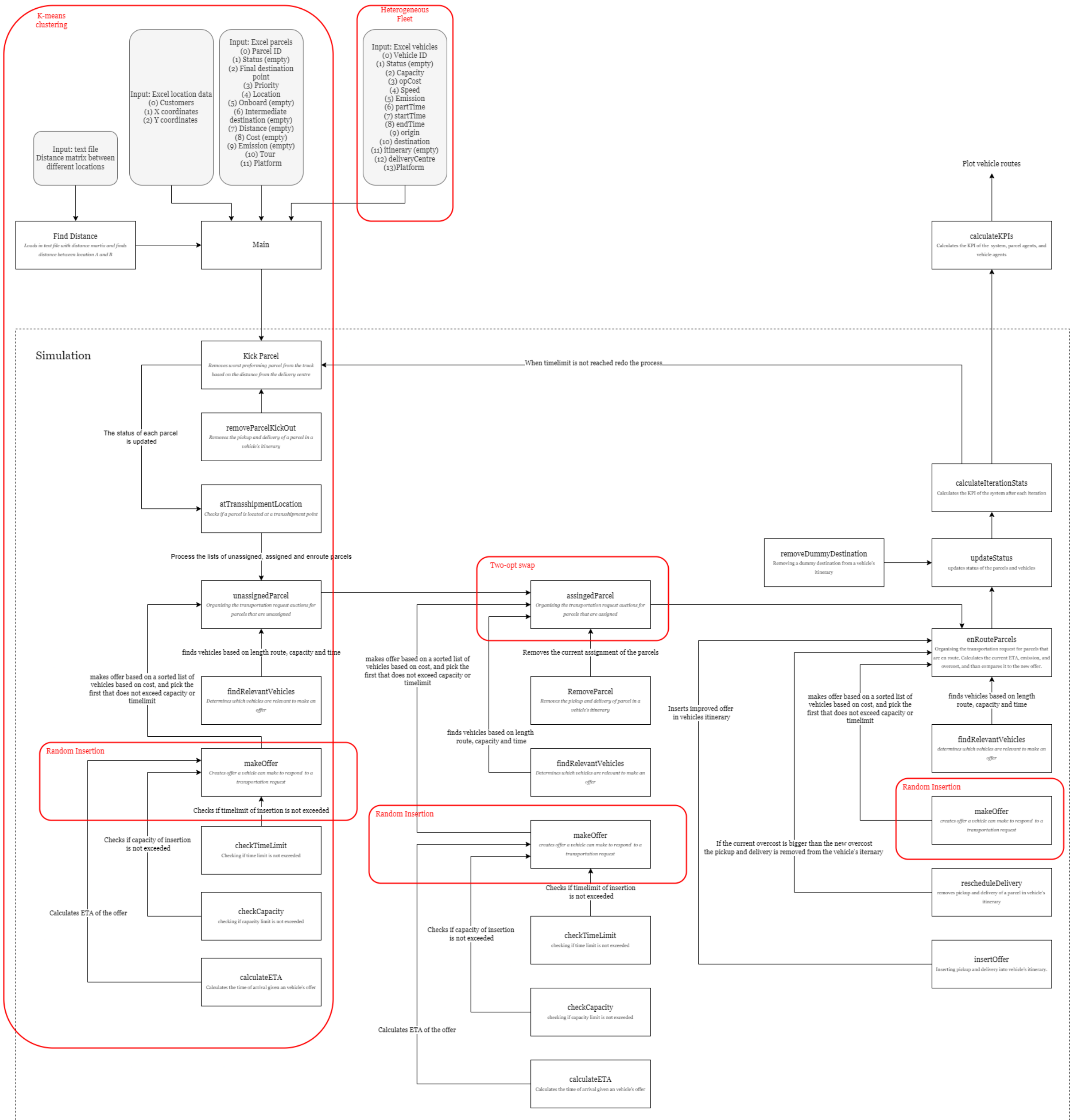


Figure 16: Diagram to illustrate the sequence of the SOLiD algorithm with alterations to the heuristics

cases, which is due to the computational complexity resulting from the lack of built-in heuristics. In addition, the dependence of the model on input data is a considerable limitation. For the integration of the model, the application of three heuristics is considered, namely random insertion, two-opt swap and k-means clustering. The simulation environment is adapted for a heterogeneous fleet with cargo bikes and vans and congestion to observe the behaviour of the model under real-time complexity in the simulation environment.

4 Simulation Model

This chapter describes the case specifics, the verification and validation, and the scenarios of the modified algorithm for the distribution of last-mile parcels. In the previous chapter, the conceptualisation of the model was explained; during the development and modification of the model, it is important to produce an accurate and credible model. For this purpose, the original model is benchmarked with respect to a series of instances and compared with an OR solver. The model is then calibrated and path dependency is assessed. Subsequently, the modification strategies are explained and the scenarios for the experiments are elaborated.

4.1 Case Specifics

The case is based on a dataset provided by a major Dutch delivery company, containing data of its delivery operations in the Netherlands on Tuesday 18 September 2018. The dataset is the same as the one used in Van Duin et al. (2021). It contains per delivered parcel the delivery address, depot number and trip number. A subset was selected consisting of 11 vehicle trips and 729 parcels to be delivered in the city of Delft and its surroundings. An overview of the delivery addresses is given in Figure 17. This data will be used to run the experiments in chapter 5. However, this chapter first focuses on a series of cases known in the literature, which will be explained in the next section.

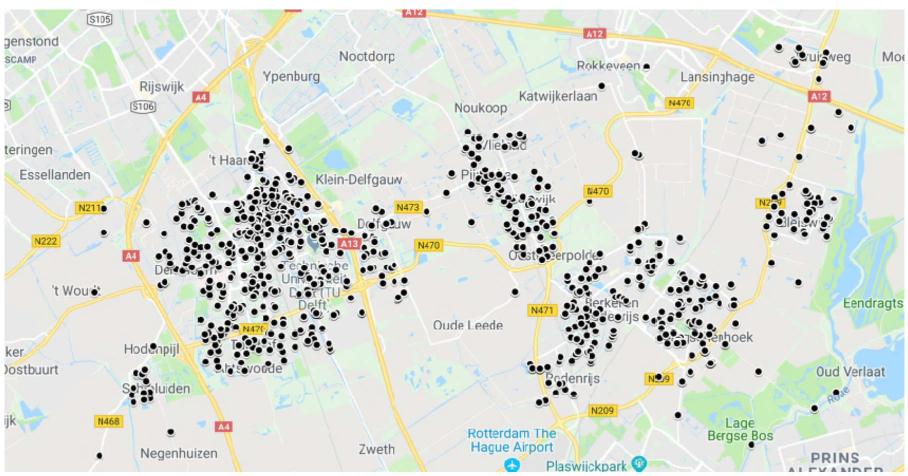


Figure 17: Overview delivery addresses taken from Vlot (2019)

4.2 Comparison for Different Instances

Since this method is not applied to real-world operations, the input-output transformation cannot be compared to real cases. When real-world data is not available, it can be particularly useful to make a comparison with other models. The original model is tested for a range of known instances to benchmark and validate the existing model. For the comparison and benchmarking of the algorithm, instances belonging to the TSPLIB library are used[1]. A graphical overview of the instances is given in Appendix G Figure 36. This set of cases was chosen for comparison, as the graphical layout is different for each case; in addition, Yousefikhoshbakht et al. (2013) results provide insight into a similar test case with one depot, a demand of one (the parcel) at each node and a fleet consisting of several vehicles. To test the effectiveness of the SOLiD method, it is compared to an OR solver and a best solution found in the literature for modified genetic algorithm (Tang et al., 2000), sweep algorithm and elite ant system (Yousefikhoshbakht and Sedighpour, 2012), and modified ant colony algorithm (Junjie and Dingwei, 2006). The OR solver uses an initial solution strategy of the cheapest arc inclusion and then further optimises the set of routes based on the set search parameters and the objective function with relates to minimisation of the travel distance. For the SOLiD method, input parameters are taken that are the same as the proposed input parameter of Vlot (2019) given in Appendix H. The fleet size and capacity are equal for all test runs. For each instance, the best solution (BS) obtained from Yousefikhoshbakht et al. (2013) is given. The difference from the BS of each method is described by the difference calculated by equation 3:

$$Gap = \frac{S^{new} - S^{bs}}{S^{bs}} * 100 \tag{3}$$

An overview of the performance of the algorithm based on the given cases is given in Table 2. The instances used are pr76, pr152, pr226, pr299, and pr439. The number corresponds to the number of nodes (delivery addresses) of each instance. The fleet size and vehicle capacity are set to the same values for the BS. In addition to the distance travelled, for the OR solver and the SOLiD method, the computation times are also given. The average values of the SOLiD methods are based on iterations over 3 different random seeds.

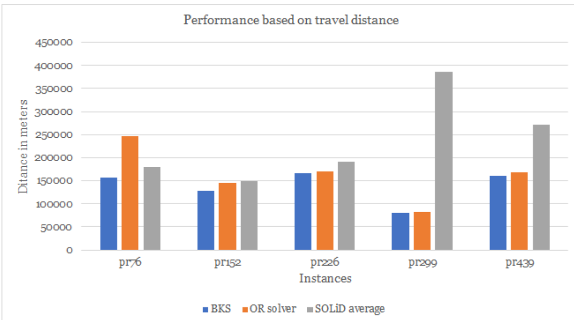
Table 2: Overview table of the performance of the the OR solver, the SOLiD method and the best-known solution taken from Yousefikhoshbakht et al. (2013)

				BS	**OR Solver**			**SOLiD average**		
	n	vehicles	capacity	Distance (m)	Distance (m)	Gap	RT (sec)	Distance (m)	Gap	RT (sec)
pr76	76	5	20	157413	246104	56,34%	6,00	180393,91	14,59%	37,64
pr152	152	5	40	127755	146078	14,34%	9,02	150162,86	17,54%	169,53
pr226	226	5	50	166827	169647	1,69%	30,03	191313,81	14,68%	753,01
pr299	299	5	70	81261	82140	1,08%	60,00	385640,14	374,57%	406,77
pr439	439	5	100	160298	168412	5,06%	734,23	272222,67	69,82%	25034,53

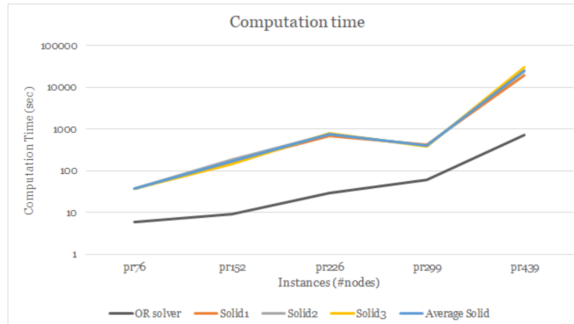
A graphical representation of the performance of the different methods is given in Figure 18a. A visualisation of the computation time on a logarithmic scale is given in Figure 18b. The instance package derived from Yousefikhoshbakht et al. (2013) also included Pr1002, corresponding to a matrix with 1002 nodes. Due to computation time, it was not possible to obtain results for the SOLiD

[1] further information on the instance library can be found on: `http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/`.

method, so it has been excluded from the overview. The first tests of the algorithm show that the random assignment of the delivery centres of the vehicles is not desirable. For the optimisation method the worst parcel is removed from the routes of the vehicles. This is based on the static, predefined delivery centres. Assigning completely random delivery centres (as proposed for the SOLiD method) could lead to infeasible or inefficient solutions. Therefore, a quasi-random allocation of delivery centres is done based on the zones in the delivery matrix.



(a) Performance based on travel distance



(b) Computation time

Figure 18: Performance and computation time of the different methods for different instances

For the five cases, the SOLiD algorithm is able to find a feasible solution. In the comparison, the SOLiD algorithm (and the OR solver) was not able to find better results for the BS of the five examples. The OR solver outperforms the SOLiD method in almost all cases, only for the pr76 instance, the SOLiD method is able to find a better solution. The OR solver is also more consistent with the solution gap and the computation time required. For the pr299 instance, an outlier can be observed for the SOLiD method, which has a large performance difference compared to the other methods. This can be explained by the layout of the nodes; since the SOLiD method depends on successive improvements using kick-outs based on distance from delivery centres, it tends to perform less for compact layouts with few outliers. As a result, the method sticks to a local minimum. Overall, the comparison shows that the SOLiD method is inconsistent en requires long computational efforts. For some instances, it can achieve good results.

4.3 Stochastic and Deterministic Parameters

In stochastic modelling, the simulation has a number of randomly determined parameter values. In the original SOLiD simulation, one stochastic variable is chosen. A random seed and a random shuffle are used for the arrival order of the parcels. For the extended model, the stochastic elements of random insertion and random swapping of parcels are used. Both elements are defined by a random seed and a random sample element. To account for the stochasticity the model has to be tested for multiple random seeds, which results in a spread of solutions. To obtain insights into this spread of the solution space a confidence interval is calculated over multiple random seeds. An overview of the stochastic elements is given in Table 3. For the deterministic part of the model, all parameters are predefined and fixed throughout the simulation. The values of the parameters are given in Appendix H and E.

Table 3: Stochastic elements of the model

Stochastic parameters			
	Stochastic parameters	Distribution	Unit
SOLiD	Parcel arrival order	Random shuffle	Parcels
SOLiD extended	Parcel insertion	Random sample	Parcels
SOLiD extended	Parcel swap	Random sample	Parcels

4.3.1 Calibration

The model has various input parameters that can be adjusted to the specific characteristics of the case. Variation of the parameters produces different results. The SOLiD method searches iteratively for a feasible solution. The aim is to efficiently generate good solutions, but it does not guarantee the optimality of the solutions found. In order to get closer to the optimality of the solution, two important parameters can be identified with respect to combinatorial optimisation, namely the parameter of reallocation iteration and the parameter of the number of kicked parcels. The first parameter limits the number of time steps in which parcels are reallocated based on the removal (parameter "kicked parcel") of the worst parcel (furthest away from the delivery centre) in each truck itinerary. To see the behaviour of the SOLiD method and the convergence towards optimality, these parameters have been varied. The results can be found in Table 16 in Appendix I. The parameter variation shows that the current algorithm can produce feasible solutions for most of the variation of input parameters. For the pr152 instance for a combination of 5 reassignments and 5 kick-outs, 8 reassignments and 1 kick out, and 20 reassignments and 20 kickouts not all parcels are delivered. The variation of the input parameters does not give a logical insight into the best-performing parameter combination. Running it for the pr76 instance suggests that a higher number of kick-outs results in better performance, but this does not hold for the pr152 instance. A possible explanation is that the algorithm gets stuck in local minima. The implementation of the random insertion of parcels in the itinerary could aid in escaping these local minima. It could also be that the algorithm is path-dependent. For convenience, the combinations of 5 parcel kick-outs and 8 reassignment iterations were chosen, as this is similar to the methodology of Van Duin et al. (2021), and would allow a comparison with the case data.

4.3.2 Path Dependence

A problem may be that the result of the model is not robust. The order of the parcels or the input parameters define the solution space for the allocation and routing. This problem is called path dependence; regardless of the model's input data, randomised or not, the starting position determines the outcomes. Path dependence can be a structural element of a problem; it cannot be solved with more or less randomness. So it makes no difference to add heuristics with random reallocation of insertions. To test the path dependency for the order of parcels that arrive in the system, a 95%

confidence interval is taken for a sample of 20 random seeds for the first three instances. To analyse the results, the confidence level for the distance travelled is graphed for every instance shown in Figure 19. The confidence level is derived from the standard deviation and significance level of 0.05.

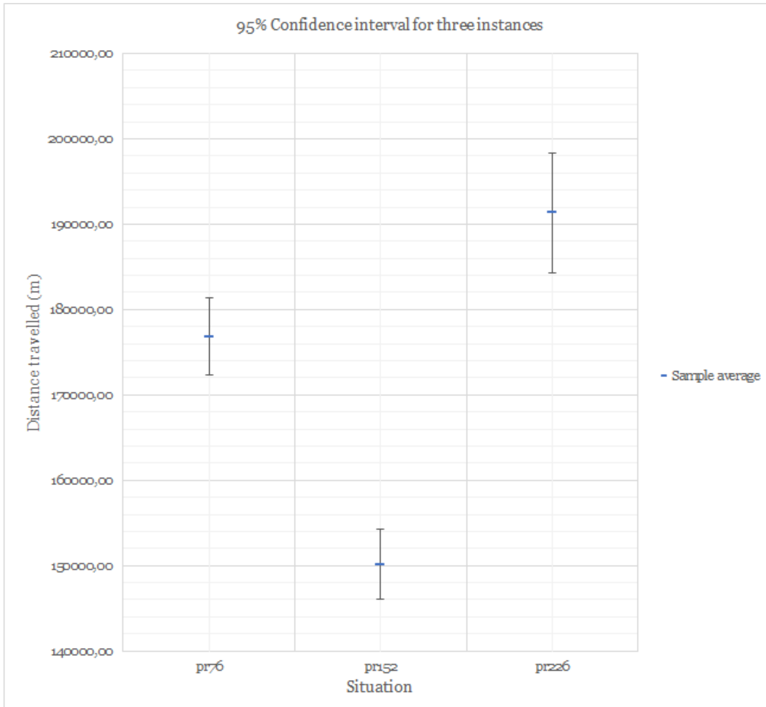


Figure 19: 95% confidence interval for instances pr76, pr152, and pr226

The confidence interval and the data from which it is derived (see Appendix J) show that for a varying random seed where the order of parcels entering the system is shuffled, each iteration gives a different result. However, the intervals are relatively small. This shows that the solution space is not path-dependent on the order of incoming parcels. This contrasts with the dispersion in the parameter variation mentioned earlier. This is an important limitation to take into account, as it makes the validation of the model more difficult.

4.4 Modifications Strategies

The modification of the algorithm exist three different alterations. The implementation of the random insertion heuristic can be obtained by limiting the calculations of the makeOffer function (described in Algorithm 1). For each combination of insertion points, an offer is calculated. This can be constrained by taking a random sample of the collection of insertions. In this way, the best offer from the sample can be chosen. The size of the sample can be determined by taking a percentage (q) of the total number of inserts. The modified algorithm is given in algorithm 3.

Algorithm 3 New auction (makeOffer) of SOLiD algorithm

Require: Offer for parcel transport request
  for The length of the itinerary **do**
    Take a random sample based on search percentage (q) of itinerary of pick up points
    Take a random sample based on search percentage (q) of itinerary of delivery points
    for For a random sample of pickup insertion points **do**
      for For a random sample of delivery insertion points **do**
        if Pickup time <= delivery time and pickup time >= departure Time **then**
          Calculate cost of pick up and delivery
          Create list of possible insertion combinations
          Sort possible insertion combinations based on summed cost
        end if
      end for
    end for
  end for
  for Each insertion combination **do**
    if If arrival time at destination > endtime delivery window **then**
      Send no offer
    end if
    if Timelimit not exceeded **then**
      if Maximum vehicle capacity is not exceeded **then**
        Select best insertion
        Create offer for parcel
      else if Maximum vehicle capacity is exceeded **then**
        Send no offer
      end if
    end if
  end for

The implementation of the two-option exchange can be done by taking a sample of a set of allocated parcels and reassigning those parcels to the existing travel routes. The current reallocation is done based on the kicked parcels based on the distance to the delivery centre. By randomly selecting a number of parcels to be switched (e.g. kicking from multiple vehicles), next to the kicked parcels, the model can incorporate the random swap heuristic. The adjusted algorithm for this is given in algorithm 4. This method can even be extended by only removing assigned parcels randomly, which means that for each iteration a random set of parcels is reassigned, which can either be placed back in its existing itinerary or placed in a better itinerary position.

Algorithm 4 Removal of the parcels

Require: Removal assigned parcels

for Each parcel in list of assigned parcels **do**

  Calculate the distance from the delivery centre of the corresponding vehicle

  Create sorted list based on distance from delivery centre

  for Length of the 'number of kickouts' for each vehicle **do**

    Remove worst performing parcels pickup and delivery in a vehicle's itinerary

  end for

  for Length of the 'random number of kickouts' for each vehicle **do**

    Remove random parcels pickup and delivery in vehicle's itinerary

  end for

end for

The real method for sorting parcels in last-mile delivery revolves around clustering delivery addresses based on postal codes. This method can be partially adopted by relaxing the assumption that information only becomes available dynamically during the period of the simulation. Assuming that the list of parcels is known, clustering can be performed beforehand to obtain the correct delivery centres, rather than assigning them randomly. In this way, the algorithm may be able to find an optimal solution faster. The clustering can be performed by k-means clustering in python and the centroid location is added to the vehicle agent's input data. To find the clusters, random centroids are first assigned, which are updated by minimising the distance of each node to each centroid. A graphical overview of the clustering for the pr76 instance is shown in Figure 20.

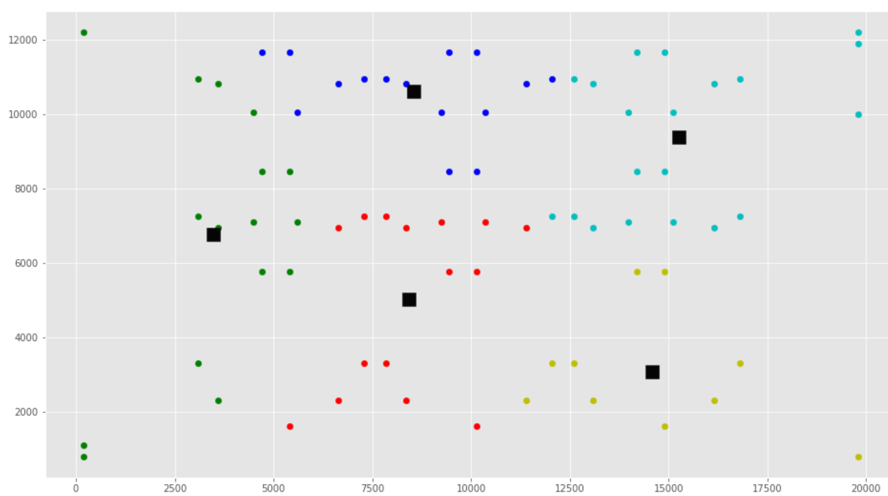


Figure 20: K-means clustering with the clusters visualised by the different colours and the centriodes depicted by the black boxes

The implementation of the mixed fleet is done by adjusting the size of the fleet and single-vehicle characteristics. Two vehicles are considered in this study, namely, delivery vans and cargo bikes.

Cargo bikes fit in the roadmap toward zero-emission city logistics, reducing the negative externalities of combustion-powered delivery vans. A downside of using cargo bikes is the limited capacity and the slower vehicle speed (Arnold et al., 2018). For short-range deliveries, however, cargo bikes are a good substitution for vans. Gruber et al. (2013) showing that cargo bikes could potentially take over 20% - 50% of the vehicle mileage in cities. The vehicle characteristics are based on capacity, operation cost, speed, and emission. These performance indicators for the delivery van (Van Duin et al., 2021) and the cargo bike (Arnold et al., 2018) are given in Table 4.

Table 4: Vehicle characteristics delivery van and cargo-bike

	Capacity	Operation cost (Euro/km)	Speed (km/h)	Emission (Gr/km)
Delivery van	113	0.44	26	230
Cargo-bike	10	0.20	12	0

The current method assumes delivery at free flow. Thus, it does not take into account the congestion that may be present. To take congestion into account, three vehicle speeds are considered: free-flow (26 km/h), light congestion (17 km/h) and heavy congestion (13 km/h) (Arnold et al., 2018). In real life, congestion depends on the area and time of day. For this study, only a congestion factor (based on vehicle speed) is used with this simplified method. As cargo bikes are allowed on cycle tracks, it is assumed that they do not experience congestion.

4.5 Validation and Verification

Verification is the process of ensuring that the model design has been translated into a simulation model of sufficient accuracy. Validation is the process of verifying that the model is sufficiently accurate for the process in question (Robinson, 1997). This study concerns the modifications of an already validated simulation model. Nevertheless, validation is a huge challenge for agent-based simulation modelling. By benchmarking using multiple instances and the parameter variation for calibration of the existing method, this work contributes to further validation of this decentral dynamic parcel distribution method. Since this method is not applied to real world operations, the input-output transformation cannot be compared to real world cases. When real world data is not available, further validation of this model may be particularly useful. As the comparison of the model with the different cases shows, the SOLiD model performs moderately in finding optimal results. Moreover, the parameter variation shows inconsistent results for different variations. This is an important consideration when evaluating the results of the experiments. By extending and modifying the model, the current method can be validated by testing it under different conditions.

The verification and validation of the model are carried out according to the scheme in Figure 21. The conceptual model validity exists of determining the theories and assumptions underlying the conceptual model are correct. The SOLiD algorithm considers a method of minimum cost insertion for assigning the parcels in the travel route. Heuristics regarding random insertions and swaps are a common method in collaborative auctions (Gansterer et al., 2020b) and dynamic allocation in VRPs

(Phillipson and de Koff, 2020, Phillipson et al., 2020). Furthermore, K-means clustering has proven to be a successful heuristic for dynamic allocation of parcels (Phillipson et al., 2020, Phillipson and de Koff, 2020).The implementation of a mixed fleet approach is inspired by Van Duin et al. (2021), Kang and Lee (2018) who use different vehicles by changing the vehicle characteristics related to capacity and speed. The use of a congestion factor is inspired by Arnold et al. (2018) who simplify congestion by using a congestion factor in a simulation study.

Variation in parameters has shown that the results of the SOLiD model can be inconsistent. Therefore, to avoid testing and modifying an invalid model, a test under extreme conditions is performed to increase the verification of the model. For the test under extreme conditions, the input parameters with extremely high and low values are taken, while the other parameters remain constant. The parameters tested in the extreme conditions test are listed in Table 5.

Table 5: Input parameters with normal, minimum and maximum values for testing under extreme conditions

Parameter	Description	Normal	min	max
reassignmentIterations	Number of assigned parcel auctions	8	0	660
numberParcelRemovals	Number of parcels removed from vehicles itineraries	5	0	20
deliveryDuration	Duration of a delivery stop	3	0	20
maxRelevantVehicles	Vehicles selected for auction	3	0	5
Capacity	Capacity of delivery vehicle	20	0	1000
Speed	Vehicle speed	26	0	10000

A summary table of the results of the extreme parameter variation is given in Appendix K. For the reassignment iterations, one would expect that more reassignment rounds would lead to better routing because the algorithm is allowed to search longer for better allocations of the parcels. For iterations with few reassignment rounds, one would expect the results to be worse because the algorithm is not allowed to redistribute any more parcels after the first iteration. The test under extreme conditions shows that for zero redistribution iterations, the parcel's performance is indeed mediocre. However, for a reassignment iteration over the full simulation run time, many invalid solutions are found, and long computation times are required. This shows that the input parameter for the redistribution iteration is not suitable for the purpose for which it is used in the model. For the experiments in the last sections, it should be taken into account that the model may be path-dependent concerning the reassignment parameter. When the kick-outs of the parcels are taken into account, it can be seen that the model performs better than without kick-outs; this is questionable because the kick-outs were included to improve the model. This means that the heuristics applied to kick out the worst-performing parcels may not be efficient. With the maximum number of kick-outs (set to the maximum capacity of the vehicles), it can be seen that most parcels are not delivered. This makes sense, as the method assumes that the parcels will be ejected and inserted into new routes. If there are no routes to insert them, unfeasible results arise. The tests for delivery times show that for longer delivery times, the travel distance increases, which is understandable because less efficient routing is possible due to the time constraint. For the number of relevant vehicles, with zero relevant vehicles, no parcels are delivered, which behaves as expected. For a higher number of relevant vehicles, the results become slightly better as there are more options to choose from in the auctions. Also for capacity, the results are as expected. With a capacity of zero, no parcels are delivered and with a high capacity, routing becomes more efficient as fewer vehicles are needed. The same can be said for speed, with a speed of zero no parcels can be delivered. With a high vehicle speed, the model does not improve much, which is logical because the model is aimed at minimising the distance.

For the operational validity of the modifications, the input from the two interviews is used to assess whether the logic is reasonable. The interviews in Appendix B and R agree with the idea of a self-organising system that can potentially outperform the current centralised system. Both mention the limitation of a simulation model as a tool to demonstrate this. In the interview with the sorting expert, it was mentioned that it is doubtful whether the decision system actually works in practice. Also in the interview with the business expert, this concern was expressed by raising the question of whether and how the physical system that could facilitate such a system would be designed. Both agree with the idea of applying the model in the simulation model to obtain more insights on applicability, but all limitations present should be considered when evaluating the results and generalising conclusions for the system in real terms.

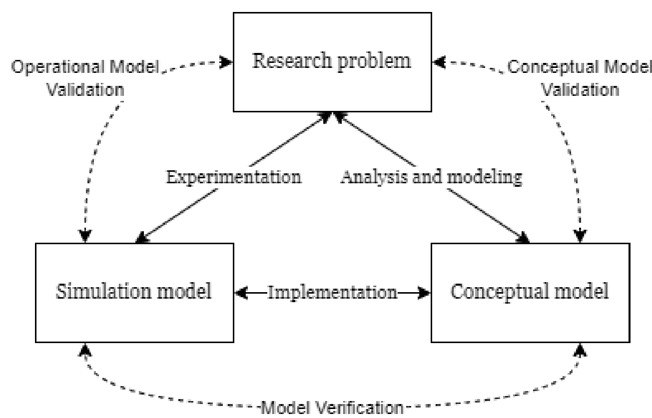


Figure 21: Validation and verification overview adapted from Sargent (2010)

4.6 Scenarios

This study aims to come up with possible improvements of the existing algorithm. Experiments will be conducted step by step to test the proposed changes and understand the changes in the performance of each method. A combination of the best-performing methods will then be tested for a mixed fleet and congestion. The test case is the pr76 instance. This instance has been chosen because the limited number of nodes allows for faster calculations and results. The last two experiments concern the complete data set of a logistic service provider in the Netherlands, as described in section 4.1. The results will be compared with the earlier work of Vlot (2019). The combination of these possibilities results in a list of experiments to be performed, which is presented below.

Table 6: Experiment overview

Experiment	Description
1	Pr76 instance tested for the original solution method
2	Pr76 instance tested for the modified solution method with random insertion
3	Pr76 instance tested for the modified solution method with swap heuristic
4	Pr76 instance tested for the modified solution method with predefined clusters
5	Pr76 instance tested for a combination of best performing aforementioned modifications
6	Pr76 instance tested for the base case method with congestion and mixed fleet
7	Pr76 instance tested for the base case method for a combination of congestion and mixed fleet
8	Pr76 instance tested for a combination of best-performing modifications with a mixed fleet and congestion
9	Pr76 instance tested for a combination of best-performing modifications for a combination of congestion and mixed fleet
10	The case data set tested for a combination of best-performing heuristics
11	The case data set tested for a combination of best-performing heuristics with a mixed fleet and congestion
12	The case data set tested for a combination of best-performing heuristics for a combination of congestion and mixed fleet

Each change is tested iteratively for 10 random seeds to understand the distribution of results when comparing the outcomes. The key performance indicator used to compare each method is the distance travelled by each vehicle. Other important factors are emissions and travel time. As these performance indicators are directly linked to the distance travelled in the model, they are not always considered.

4.7 Summary

For the development and modification of the new model, it is good to benchmark and validate the current model. The SOLiD method is able to find feasible results for different instances, but seems to have limitations as the results are mediocre and larger instances require considerable calculation time. The calibration of the number of reassignment iterations and the number of kick-outs also shows that the model produces different results for different inputs. Testing the model for path dependence for the sequence of input parcels shows that this is not the case for the sequence of parcels. The model is further validated by comparing the conceptual changes with known studies and by a test on extreme parameter conditions. This test again reveals limitations of the SOLiD method, as the reassignment iteration and the number of kick-out parameters do not behave as expected. Showing that that the number of kick-outs and iterations do not fit their purpose of constraining or improving the model. These limitations are important when evaluating the results in the following chapters. The modification strategy for the random insertion method consists of limiting the search sample based on a search rate. For the swap, the strategy is to remove a random number of parcels from the vehicle's route so that they can search again for placement in the available fleet. The clustering is done using k-means clustering and the delivery centres are adjusted accordingly. In terms of changes in the environment, the mixed fleet is incorporated by changing the fleet size and the characteristics of the vehicles in the fleet. Congestion is added by adding a congestion factor to the delivery vehicles. This leads to a list of experiments to be carried out. To structure these experiments, first the test cases are used to obtain preliminary results that can be used to structure the experiments for the case data, which will require considerably more calculation time.

5 Model Application

This section will give an overview of the results of the different experiments that have been carried out. The first part will consist of a test case in which the different modifications are tested and evaluated. Based on this first round of experiments, the model will be tested using real-life case data. This section will conclude with an evaluation of the implementation of this type of decision-making method in logistics. The experiments in the python simulation consist of multiple variations of the heuristics and changes in the simulation environment. The influences of the changes are assessed based on the distance travelled, emission, and computation time. Each experiment is set to run for 660 iterations, corresponding to 660 minutes, simulating a working day from 08:00 to 19:00. The first iterations are considered to be the start-up of the model, where the parcels are iteratively allocated and reallocated, after which the shipments start and the parcels can only be reallocated by trans-shipments. Each set of experiments is performed for different random seeds, in order to gain insight into the distribution of the solution space. Unless mentioned otherwise, the parameter input is set to the input values described in Appendix H Table 15. For the first experiments, the pr76 instance is used, as a test bed for all modifications. The last experiments include the case data set.

5.1 Base case

The first experiment is performed as a reference case. In the reference case, the pr76 instance is compared to the original method[2]. The results of the base case are shown in Table 7. These results are similar to the confidence interval described in Figure 19. A feasible solution was found for all random seeds and no parcel was not allocated.

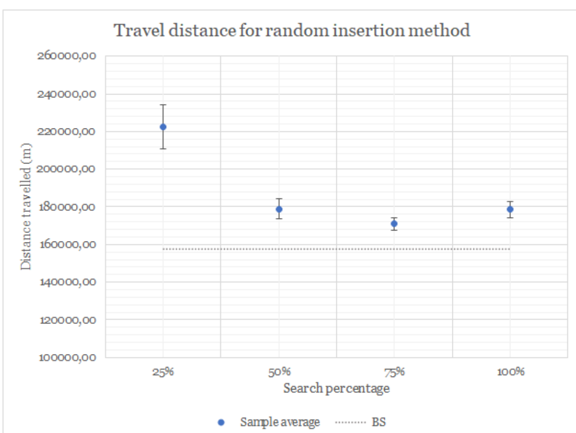
Table 7: Reference case pr76 original SOLiD method

Pr76	
Randomseed	Distance travelled (m)
1	183230,94
2	182202,24
3	175748,57
4	182687,54
5	163053,44
6	185631,51
7	180425,63
8	177480,88
9	171657,06
10	183976,87
Average	178609,47
Best performance	163053,44

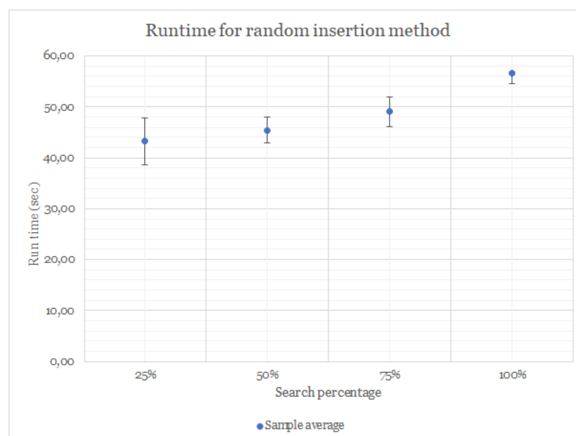
[2]The original algorithm has been debugged and it has been adapted to fit into Spyder's integrated development environment (`https://www.spyder-ide.org/`).

5.1.1 Random Insertion

For random insertions, the algorithm is adjusted to limit the number of insertion points, and therefore the number of calculations, for each parcel. The logic behind the adaptation uses a search rate. In this series of experiments, the distance travelled, the run time and the number of unassigned parcels are derived for a set of 10 random seeds. This is done for a search rate of 25%, 50%, 75%, and 100% (same as the reference case). An overview of the results is given in Appendix L with a table of the results (Table 20) and a graphical representation of the routing for the different search rates are given in Figure 37. At the search rate of 25%, the algorithm was unable to assign all the parcels in some random seeds, and thus these runs had no feasible results. For the other search intervals, all parcels were assigned and thus provided a feasible solution. For the travel distance and the run time, an overview is given in Figure 22 for a confidence interval with a significance level of 5%. The figure shows that the average travel distance for the fleet decreases with a higher search percentage. It is interesting to note that the travel distance for a 75% search sample gives a better result than the 100% search with a similar deviation. In general, the confidence interval gets smaller with a larger search sample for insertion. When looking at the calculation time, a similar trend behaviour can be seen. For smaller search percentages the dispersion is larger than for the higher search rate samples. The calculation time of the algorithm becomes larger with a larger search rate and corresponding insertion sample range.



(a) Travel distance of the random insertion method

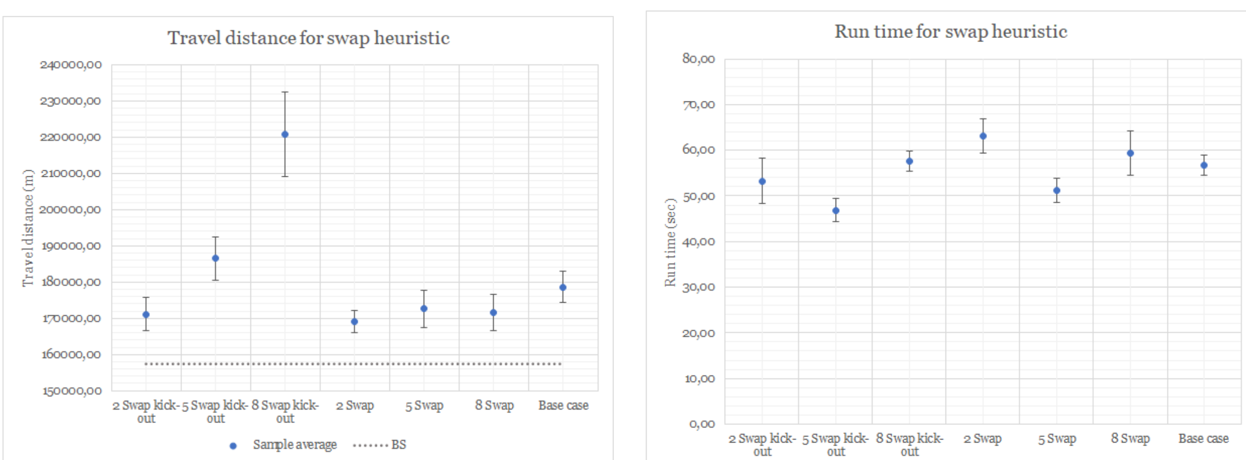


(b) Runtime of the random insertion method

Figure 22: 95% confidence interval for the distance and run time of the random insertion method with the 100% search rate being the same as the base model

5.1.2 Two-opt Swap

For the swap, the algorithm is modified to randomly reassign a predefined number of parcels. The method is evaluated in combination with the current kick-out method which kicks out outliers in the route of the vehicles. The number of kick-outs is kept constant at a predefined value of 5. The algorithm is analysed for a swap of 2, 5 and 8 parcels. The distance travelled, the run time and the number of unassigned parcels are derived for a set of 10 random seeds for each experiment. Earlier parameter analyses in section 4 showed irregularities with regard to the kick-out method. Therefore, besides testing the random swapping of a group of parcels per vehicle in combination with the original kick-out method, a situation where only swapping is used is also evaluated. This is done to see how the algorithm performs under swapping alone. An overview of the results can be seen in Appendix M, with a results overview table of the swap in combination with the kick-outs (Table 21) and only the swaps (Table 22) and a graphical representation of the routing for the different swap cases in Figure 38. For the travel distance and the run time, an overview is given in Figure 23 for a confidence interval with a significance level of 5%. When looking at the travel distance of a combination of the swap and kick-out method, it can be seen that for 2 swaps the algorithm achieves slightly better results than for the base case. For larger numbers of swaps, the performance decreases and the distribution of the solutions becomes less consistent. For the swap-only method, the performance is fairly stable and slightly better than in the base case. The spread becomes slightly larger for larger numbers of swaps. The run time of the experiments shows that all methods perform around the same time interval. It is interesting to note that the method with only random swapping takes longer than a method with both swapping and kick-outs, which would have more computational steps. The method of swapping can improve the results, but overall, the influence of the swapping heuristic does not yield significant benefits when both computation time and performance are considered.



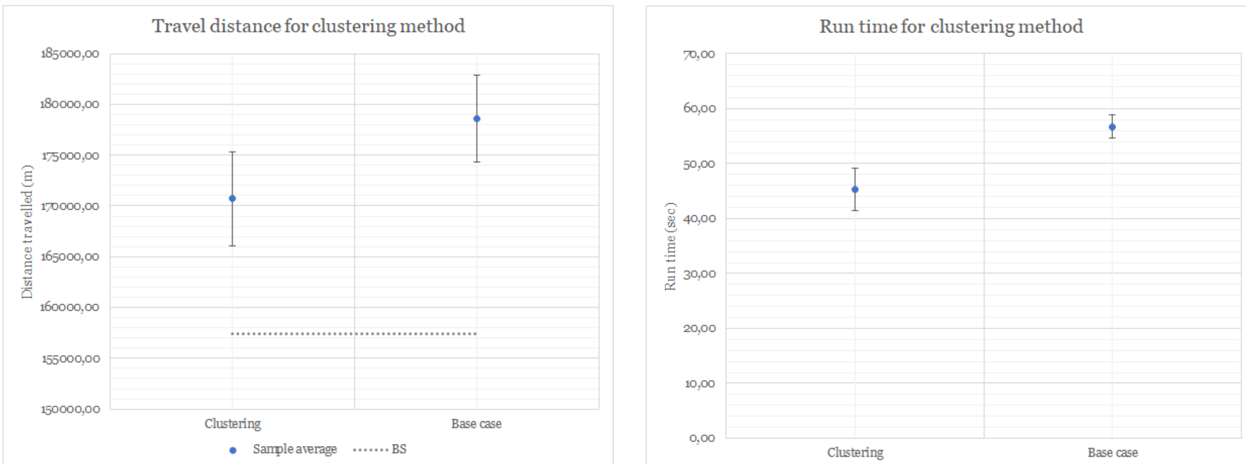
(a) Travel distance of the swap method

(b) Runtime of the swap method

Figure 23: 95% confidence interval for the distance and run time of the swap method

5.1.3 Clustering

Clustering is done separately before the algorithm is executed. By defining delivery centres the algorithm mimics the current situation in which all parcels are known before the final assignment. Having predefined clusters, therefore, limits the dynamic characteristics of the model. The clusters are set according to the k-means clustering done in Figure 20. A result table of the cluster method can be seen in Appendix N. Clustering is a step towards better results. As the as can be seen in the results in Figure 24. The performance of the model improves for a faster computation time. An important note to make here is that the computation time of the clustering is not taken into account. Besides, in the base case, the delivery centres are chosen quasi-randomly, i.e. the centres are placed in a specific part of the matrix. In a situation with completely random centres, the results regarding the performance difference could be more significant.

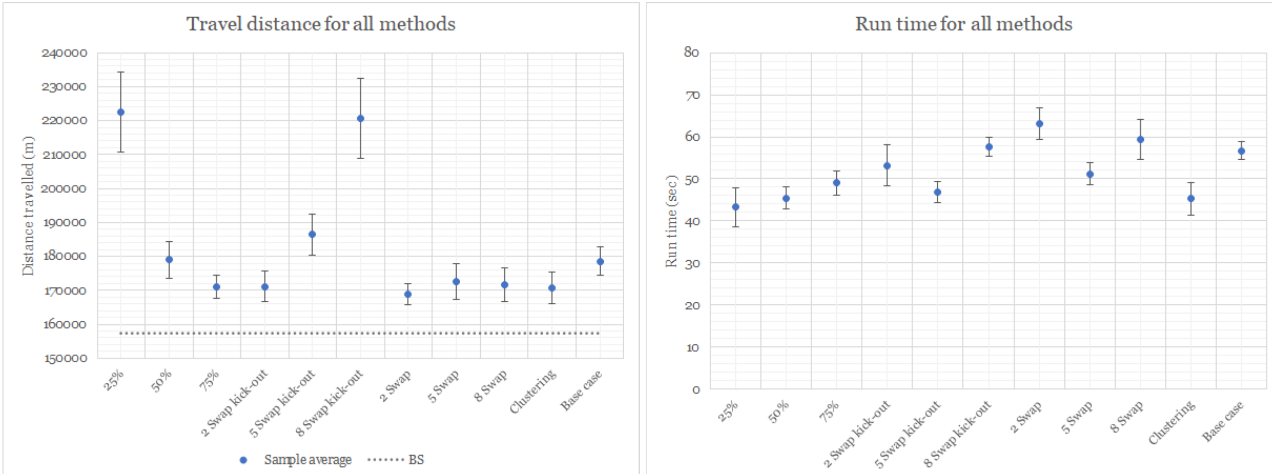


(a) Travel distance of the cluster method **(b)** Runtime of the cluster method

Figure 24: 95% confidence interval for the distance and run time of the cluster method

5.1.4 Combination of Heuristics

When considering a combination of the different heuristics, it is good to give an overview of the performance of the different heuristics, which can be seen in Figure 25. In general, it can be observed that the adjustments to the model only bring about small changes in the results. The overview and the previous results show that the swap heuristic performs well for some combinations for travel distance and for some combinations for run times, but for a combination of both, it does not give favourable results. Therefore, when combining heuristics, the emphasis is on combining the random insertion method and the clustering method. Both the 50% and 75% search samples performed well considering travel distance and computation times. Choosing to cluster for the combination of heuristics means that the dynamic capacity of the model is limited by the pre-allocated clustering. In addition, the decision-making process becomes more centralised, as the system assigns the parcels centrally in advance. Some of the dynamic behaviour is still intact, as decision-making during the simulation is still dynamic and decentralised. Therefore, it was decided to include clustering in the combination of heuristics to be able to compete better with a fully centralised approach.

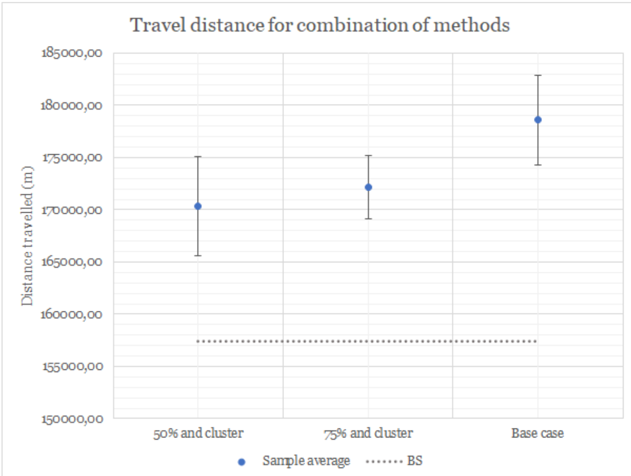


(a) Travel distance of all methods

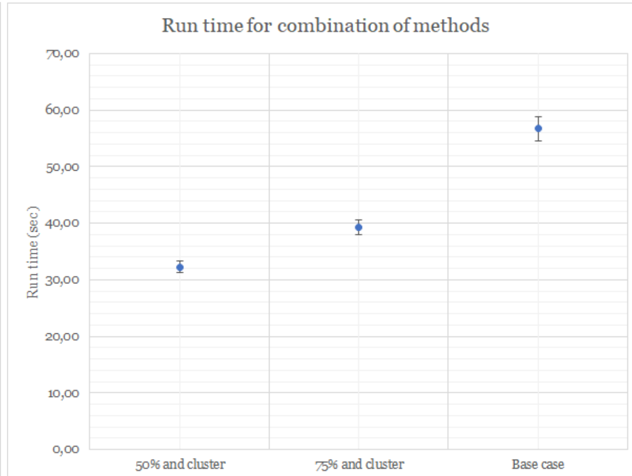
(b) Run time of all methods

Figure 25: 95% confidence interval for the distance and run time of all methods

For the combination of the 50% and 75% search rate and clustering, the results are shown for a 95% confidence interval in Figure 27. The combination of methods performs better than the base case, both for distance travelled and computation time. When comparing it to the clustering-only case the combination of methods does not improve based on performance, but for the 50% rate, it is able to half the computation time. The spread of the solution is similar for the methods. It is interesting to note that the combination with the 50% search frequency performs slightly better than the 75% combination, as this was not observed for the separate tests. Regarding the computation time, a logical increase can be observed for the larger search sample. Given these results, the combination of a 50% search interval and clustering is chosen for application to the case data.



(a) Travel distance for a combination of methods

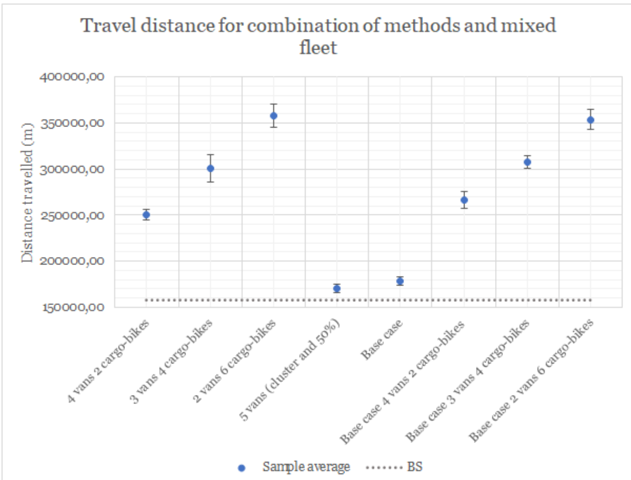


(b) Run time for a combination of methods

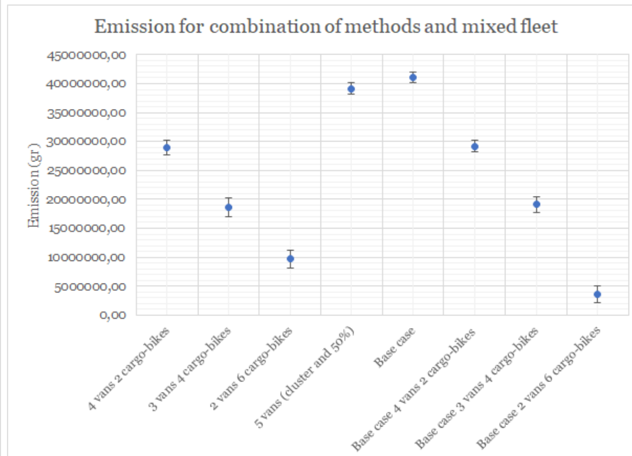
Figure 26: 95% confidence interval for the distance and run time for a combination of methods

5.1.5 Mixed Fleet and Congestion

For this set of experiments, the simulation environment is modified for a heterogeneous fleet for the modified version of the model with the combination of the best-performing heuristics and the base case. The mixed fleet is included by changing the input characteristics of the vehicles according to the parameters defined in section 4.4. For the experiments, three situations are considered in which cargo bikes replace vans. The total capacity of all vehicles is kept the same. The experiments are a combination of 4 vans and 2 cargo bikes, 3 vans and 4 cargo bikes, and 2 vans and 6 cargo bikes. For this set of experiments, the distance travelled, the run time and the number of unassigned parcels are derived for a set of 10 random seeds. In addition, CO_2 emissions are also taken into account. In previous experiments, it was assumed that all vehicles emit the same amount, which can be easily calculated by multiplying the distance travelled by the emission factor. In these experiments, the cargo bikes cause no emissions. The tests are first performed for the base case, after which they are applied to the improved model of the best performing heuristic, this allows to see the effects of real-time changes in the simulation environment on the performance of the improved and base model. Appendix P gives an overview of the results and example graphs for the different fleets. The results in Figure 41 show that for a combination of 4 vans and 2 cargo bikes and 3 vans and 4 cargo bikes the algorithm is able to obtain feasible results for all random seeds. For the combination of 2 vans and 6 cargo bikes, the fleet is not able to deliver all parcels for some random seeds. This can be observed for both the improved method with a combination of heuristics and for the base case. Looking at the travel distance of all methods, it can be seen that the heterogeneous fleet performs less well than the homogeneous fleet. The travel distance is significantly higher for all cases. Moreover, the travel distance increases as the number of cargo bikes increases. On the other hand, the experiment shows that emissions decrease as the number of cargo bicycles increases. This shows that there is a trade-off between lower distance travelled with a more efficient van-only fleet and lower emissions with a less efficient mixed fleet. The runtimes are similar for the different fleet combinations. For the improved method, it can be seen that the runtimes are lower than for the base method. The comparison between the improved method and the base case shows that the performance of the model is not significantly affected by the changes in the heuristics. Both methods perform within the same range for both travel distance and emissions. The improved model achieves only slightly better results.



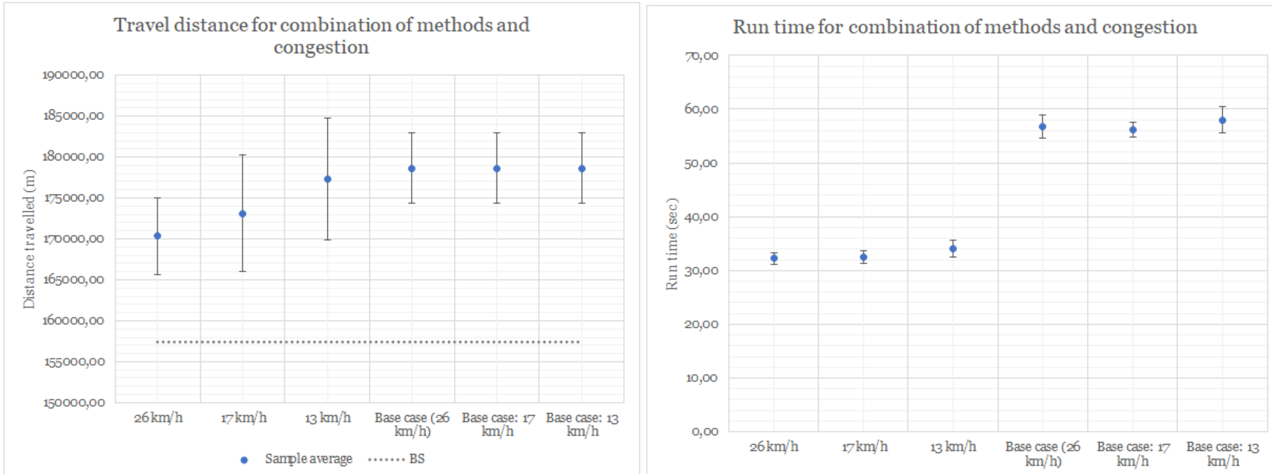
(a) Travel distance for a combination of methods and mixed fleet



(b) Emission for a combination of methods and mixed fleet

Figure 27: 95% confidence interval for the distance and emission for a combination of methods and mixed fleet

To account for congestion in the simulation environment, three vehicle speeds in the fleet of vans are considered for the modified version of the model with the combination of the best-performing heuristics and the base case. Free flow with the current speed of 26 km/h, light congestion with a speed of 17 km/h, and heavy congestion with a speed of 13 km/h. For this set of experiments, the distance travelled, the run time and the number of unassigned parcels are derived for a set of 10 random seeds. The congestion rates are also applied to the base case to see the behaviour of the improved model against the old. An overview of the results is given in Appendix Q. A graphical representation is given in Figure 28. For all three levels of congestion, the model is able to deliver all parcels within the working day in the old and the new model. Interesting to see is that the performance for all three congestion rates for the original method is the same as the base case. Meaning that the model is not affected by the slower vehicle speeds. For the improved method the distance travelled goes up slightly for higher congestion rates, but the changes are minimal. The computation time is better for the improved method when comparing it to the base method and stays around the same time interval.



(a) Travel distance for a combination of methods and congestion

(b) Emission for a combination of methods and congestion

Figure 28: 95% confidence interval for the distance and emission for a combination of methods and congestion

The integration of real-time complexity for a mixed fleet showed that the solution method for sorting and routing parcels is not able to deliver all parcels with a high number of cargo bikes. One of the reasons for including cargo bikes in the model is to look at the effects of low-emission vehicles. Furthermore, it is assumed that cargo bikes are more manoeuvrable in transport by using bicycle lanes, which makes them not affected by congestion. To see the effects in the simulation model for a combination of mixed fleet and congestion, a mixed fleet with 4 vans and 2 cargo bicycles and 2 vans and 6 cargo bicycles is combined with congestion speeds of 13 km/h and 17 km/h. The average results of 10 different seeds are shown in table 8. This is done for both the improved solution method and the base case. The results show that for this test example the effect of more or less congestion is minimal, as for both models the performance in the trip distance is comparable, if not equal, for the same fleet combination with different congestion rates. It is clear that the model is not able to deliver all parcels for the combination of congestion and a mixed fleet for the modified solution method, as several parcels are not delivered. This shows that the new model is less able to cope with this kind of complexity. The run time for this modified model is significantly less than for the base model. For the emission, it could again be observed that for more cargo bikes, less CO_2 is emitted.

Table 8: Results for a combination of mixed fleet and congestion for different fleet sizes (van/cargo bike) and congestion rates for the Pr76 instance

Pr76		4 vans and 2 cargo bikes and congestion 13 km/h	2 vans and 6 cargo bikes and congestion 13 km/h	4 vans and 2 cargo bikes and congestion 17 km/h	2 vans and 6 cargo bikes and congestion 17 km/h	Free flow and all vans fleet
Original solution method	Number of unassigned parcels	0,00	0,60	0,00	0,60	0
	Distance travelled	265791,54	350306,80	266237,57	342710,70	178609,4664
	Run time	54,34	45,26	52,95	44,31	56,73694945
	Emissions	29804577,10	7387455,22	29203776,60	6693386,67	41080177,27
Modified solution method	Number of unassigned parcels	1,60	0,10	1,60	0,10	0
	Distance travelled	233790,70	300840,50	233790,70	300840,50	178609,4664
	Run time	24,12	20,31	28,29	21,07	56,73694945
	Emissions	36396703,60	24042819,80	36396703,60	24042819,80	41080177,27

5.1.6 Interpretation of Results

Overall, it can be concluded that the implementation of the various heuristics improved the model, but the gain in performance was relatively minimal. The solution space for all results in the previous experiments varies. All methods obtain different results. This shows that there is a wide range of feasible solutions for the problem case, but also that the algorithm does not get stuck on similar local minima. It may also mean that this decision model is unable to find these local minima by applying the current method. The random insertion method showed that it is not necessary to consider all insertion points when determining the distance travelled. It showed that it can be beneficial to have a lower search frequency to reduce the computation time. Because the algorithm loops through the 'make offer' functions for every offer and time iteration the 50% search rate is applied many times, this helps to arrive at as good a solution as the full enumeration. The swapping heuristic could not make any significant gains. It was interesting to see that for a swapping-only method the computation times remained the same, this suggests that sorting parcels based on travel distance is not an essential step in creating a more efficient algorithm. This corresponds to the irregularities of the kick-out parameter evaluated in section 4. Clustering the parcels is a logical step in creating a more efficient algorithm. The applied greedy method assumes the cheapest insertion in the already existing route and the delivery centre, if the vehicles are already assigned to clustered areas the solution method can perform better and more efficiently. It also shows that the current real-life method of delivery areas per vehicle can be beneficial for the performance of the model. The choice to apply to clusters is difficult because it reduces the dynamic properties of the model. For further extensions of the model, clustering can be included in a more dynamic sense, for example by using historical data from previous weeks to decide on delivery centres. The combination of methods gives slightly better results in terms of travel distance, but significantly better results in terms of calculation time, with a decrease for the 50% search rate plus clustering of almost 40% compared to the base case. This combination is chosen to be applied to the case data. For the mixed fleet it can be seen that with a higher number of cargo bicycles, emissions decrease accordingly. On the other hand, the distance travelled increases with a higher number of cargo bicycles. When lower emissions are desired, a less efficient network with more cargo bikes can be chosen. For 6 cargo bicycles and 2 delivery vans, both the basic method and the improved solution method are not able to deliver all parcels. The lower vehicle speeds and the lower capacity of the bicycles make the routes less efficient, which leads to unfeasible results. This shows that the model has problems in efficiently assigning the farthest parcels to the vans and the shorter distance parcels to the cargo bikes. This can also be seen in figure 41 where more routes can be observed, but not necessarily short- and long-distance routes, as can be expected in such a mixed fleet system. In the case of congestion, the efficiency of the network decreases slightly as the congestion increases. However, the effects of lower vehicle speeds are marginal. Congestion should mainly affect the time constraint, as vehicles cannot pick up new parcels. In this small set of cases, this apparently has no effect, as everything can be delivered on time. It will be interesting to see how this behaves for the real dataset, where time may be a bigger constraint. The combination of congestion and a mixed fleet again shows that the effect of congestion is minimal. The modified model is less able to cope with the addition of real-time complexity than the original model, as more parcels are unassigned. The model as a whole can be made more efficient by applying the proposed heuristics, but the changes only affect efficiency and do not improve the logic of the model itself. The unfeasible results that arise when congestion and a mixed fleet are taken into account show the limited ability of the solution method to adapt to changes in the simulation environment.

5.2 Case Application

The following experiments will be conducted using the case data of 729 delivery points with the distance matrix calculated from the longitude and latitude of the geographical locations of these data points in the Delft areas.

5.2.1 Case data with a combination of random insertion and clustering

For the case data clusters are derived using k-means clustering, with 11 centroids distributed over the 729 data points. These centroids are then included in the coordinate and distance matrix of the algorithm, specified as fleet delivery centres. To be able to compare the results of the algorithm by Vlot (2019), the same input parameters are used. Due to the computation time the algorithm is not able to run multiple random seeds. Previous results have shown that the confidence interval of the results is acceptable and therefore it is assumed that this is also the case in this instance. The experiments again evaluate the results for distance travelled, calculation time and unassigned parcels. An overview of the predefined clusters with the centroids is given in Figure 29.

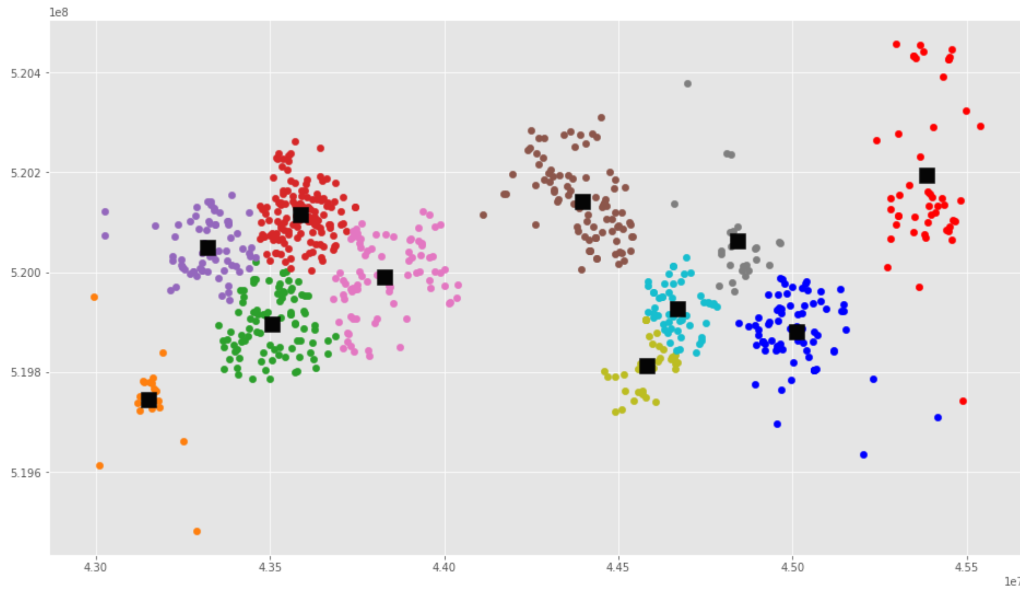


Figure 29: K-means clustering with the clusters visualised by the different colours and the centriodes depicted by the black boxes for the case data

The results for the combination of clustering and random insertion heuristics are presented in an overview Table 9. There, they are compared to the model solution for the same dataset as Van Duin et al. (2021). For comparison, an external centrally organised solver used in Section 3 is also included. The results in the previous experiments showed improvements in terms of improved model performance and efficiency. Looking only at the performance of the model, the modified version of the algorithm is able to improve the original algorithm by about 4.6%. It is interesting to note that not the entire fleet is used in the model. The current routing takes into account 9 out of 11 vehicles. This is possible because the capacity of a vehicle is set at 113 parcels. A similar phenomenon can be seen with the OR solver which also uses only 10 of the 11 available vehicles. This suggests that it is more efficient to use fewer vehicles. When the calculation time of the model is taken into account, it can be said that it is unable to find a solution efficiently. It took more than 3 hours to perform the full set of calculations. Due to the difference in computing power between this study and the study of Van Duin et al. (2021), the computation times cannot be compared. However, the OR solver and the new method can be compared. The OR solver can generate a solution within 874 seconds that is better than both the original method and the improved version of the decentralised algorithm. A graphical representation of the routes of the vehicles can be seen in Figure 30.

Table 9: Solution overview of the different solvers

	Vehicle distance travelled (km)	Computation time (sec)
Original SOLiD method	399,55	
Modified version	380,92	12443
OR solver	369,87	874

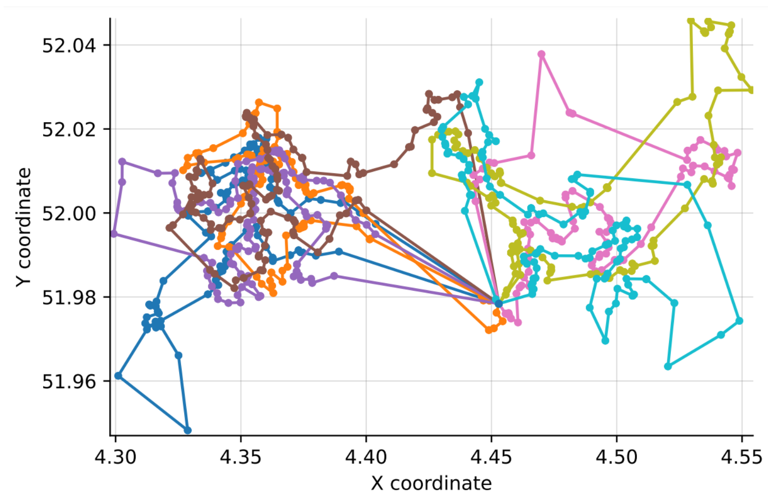


Figure 30: Overview of the routing for 11 vehicles for 729 delivery addresses derived by the combination of the random insertion and clustering method

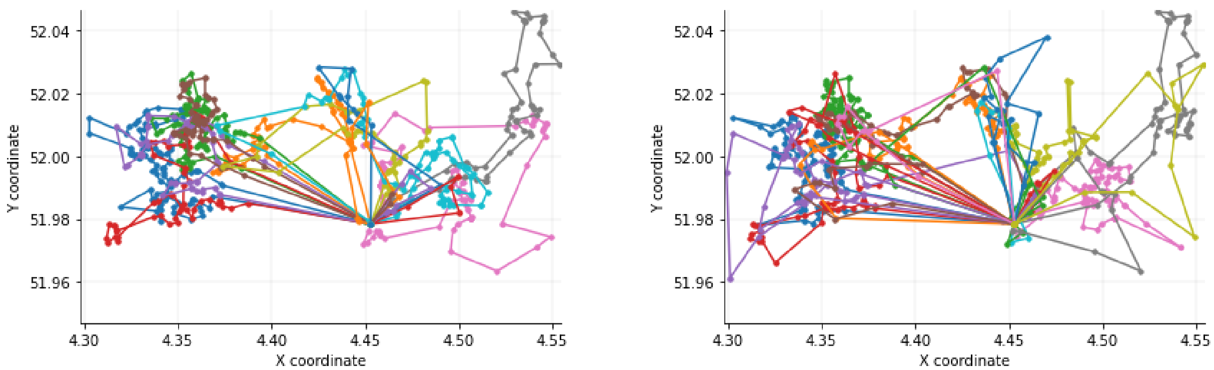
5.2.2 Case for mixed fleet and congestion

For this set of experiments, the simulation environment is modified for a mixed fleet and congestion. This is done for the modified model with the best performing heuristics of clustering and random insertion. The case data is used to see the effects of these changes, since run time is limited, the experiments are only conducted for one random seed. Both experiments are tested for distance travelled, run time and emissions. For the mixed fleet, the vehicle agent is adjusted so that there are 10 vans and 10 cargo bikes and 9 vans and 20 cargo bikes for two experiments. The total capacity of the fleet is kept approximately the same. The vehicle characteristics are further described in Table 4. The results of the experiments with the different heterogeneous fleet combinations are shown in Table 10.

Table 10: Table of solutions for the case data simulation of the combination of best-performing heuristics and different fleet combinations

	Mixed fleet 10 / 10	Mixed fleet 9 / 20	All vans
Number of unassigned parcels	60	37	0
Distance travelled (km)	500,277999	687,761	380,92
Run time (sec)	16786,6749	7895,5655	22400
Emissions (gr)	100098,53	84076,96	87612

For the fleet with only vans, all parcels can be delivered to all delivery points. However, for the mixed fleet, it can be seen that both for a fleet of 10 delivery vans with 10 cargo bikes and a fleet of 9 delivery vans with 20 cargo bikes, not all parcels can be delivered. Interesting to see is that the higher the relative number of delivery bikes does not mean the higher the number of unassigned parcels. As in the pr76 test case, the number of kilometres driven increases with the number of cargo bikes, but the emission values decrease. The run time for this experiment with the test case is similar for both cases and is again high. A visualisation of the routes is given in Figure 31.



(a) Mixed fleet 10 vans 10 cargo-bikes

(b) Mixed fleet 9 vans 20 cargo-bikes

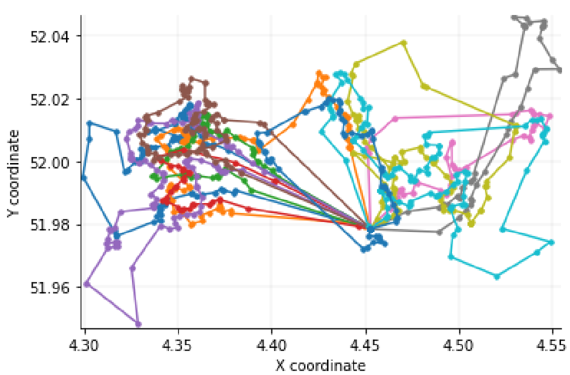
Figure 31: Routing for two heterogeneous fleet combinations of vans and cargo-bikes for the modified model with random insertion and clustering

The case data and the simulation are tested for different congestion situations. Again, three situations are considered, free flow of 26 km/h, light congestion of 17 km/h, and heavy congestion with 13 km/h for a fleet with only vans. The different congestion levels are implemented in the modified model. The results concerning the distance travelled, run times and emissions are shown in Table 11.

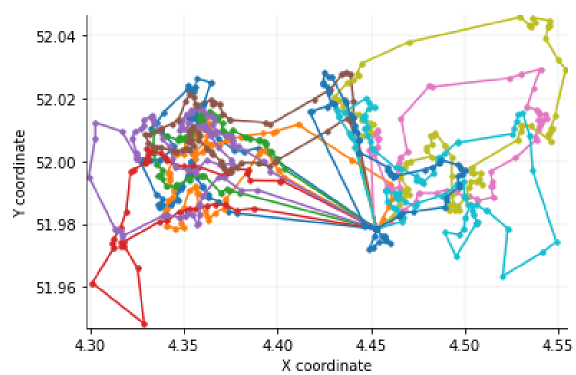
Table 11: Table of solutions for the case data simulation of the combination of best-performing heuristics and different levels of congestion

	Congestion 13 km/h	Congestion 17 km/h	Free flow
Number of unassigned parcels	0	0	0
Distance travelled (km)	450,61	442,26	380,92
Run time (sec)	25887	22233	22400
Emissions (gr)	103640	101721	87612

The model is capable of delivering all parcels for all congestion levels so that a feasible result is obtained. With higher congestion levels, the distance travelled increases. The emissions also increase with higher congestion levels, due to the direct relationship between distance and emissions. The travel times of the three methods are within the same time range. For the highest congestion level, the run time is the longest. An overview of the routing is shown in Figure 32.



(a) Case data with congestion rate 13 km/h



(b) Case data with congestion rate 17 km/h

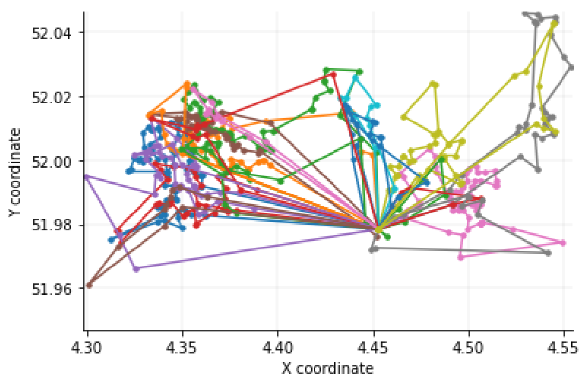
Figure 32: Routing for two states of congestion for the modified model with random insertion and clustering

The experiments showed that congestion did not significantly affect the operation of the system. The mixed fleet did show unfeasible results, with several parcels not being delivered. The cargo bikes in the mixed fleet situation are assumed to flow freely at all times and are not subject to the existing congestion. It is therefore interesting to see the effects of mixed fleet and congestion combined, the situation with the two different fleet combinations is combined with the two congestion factors of 13 km/h and 17 km/h. The results are shown in Table 12.

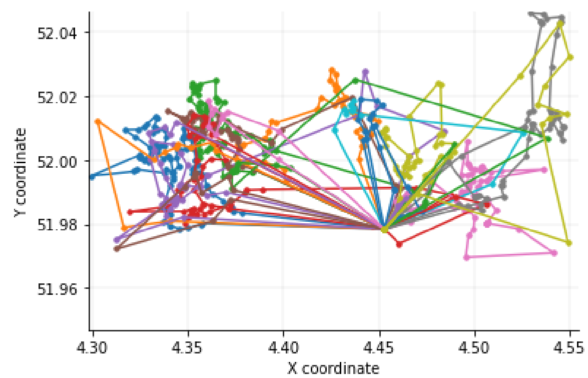
Table 12: Results for a combination of mixed fleet and congestion for different fleet sizes (van/cargo bike) and congestion rates

	Mixed fleet 9 / 20 and congestion 13 km/h	Mixed fleet 9 / 20 and congestion 17 km/h	Mixed fleet 10 / 10 and congestion 13 km/h	Mixed fleet 10 / 10 and congestion 17 km/h	Free flow and all vans fleet
Number of unassigned parcels	193	137	206	142	0
Distance travelled (km)	659	694	399	425	380,923
Run time (sec)	1987	3587	2872	3044	22400
Emissions (gr)	76673	82336	80915	86423	87612,29

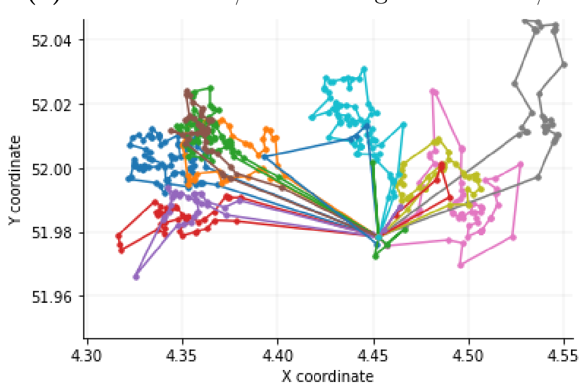
The results show that with higher congestion rates in the mixed fleet, fewer parcels are delivered. With a higher number of cargo bicycles, relatively more parcels are delivered at the same congestion rate. This is a logical consequence of the fact that cargo bikes do not experience congestion and can therefore improve the system. This suggests that in a system with congestion, a mixed fleet with delivery bikes is desirable. A graphical overview of the four situations is given below in Figure 33.



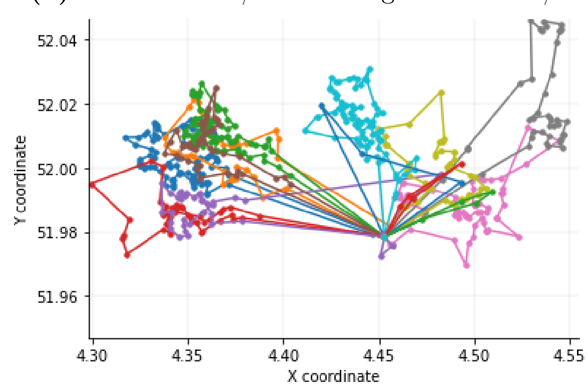
(a) Mixed fleet 9 /20 and congestion 13 km/h



(b) Mixed fleet 9 /20 and congestion 17 km/h



(c) Mixed fleet 10 /10 and congestion 13 km/h



(d) Mixed fleet 10 /10 and congestion 17 km/h

Figure 33: Overview instances

5.2.3 Interpretation of Results

The experiments for the test case already suggested that the model could be improved, both in terms of better routing and computation time. The results for the different solvers show that this is indeed the case. The modified version of the SOLiD method is able to improve the algorithm by 4.6%. Both clustering and random insertion contribute to this. By randomly choosing the available insertion point, the model can escape local minima and with the applied clustering the vehicles can easily assign parcels near their delivery centres. However, the visualisation of the new method shows several long detours and vehicles crossing their own route. This indicates that the model is far from optimised. This is also supported by the external solver used, which can generate a better solution. Another interesting fact is the number of vehicles used for the modified version. Due to the spare capacity of the vehicles defined in the input parameters, the model tends to use fewer vehicles to obtain better results. When considering the results of the mixed fleet, this idea is also supported, as it can be observed that more kilometres are made for more used vehicles. The mixed fleet results show that when a van is replaced by cargo bicycles, the model is not able to generate a feasible solution, as about 8% of the parcels are not delivered. This can be explained by the slower speed of the vehicles, which makes each cargo bikes subject to the time constraints present. The performance of the model at different congestion levels showed that the total distance increased at higher congestion levels. This can also be explained by the fact that the lower speeds cause time constraints, which makes the vehicles less able to accept parcels and thus the route becomes less efficient. For a combination of congestion and a mixed fleet, this can be seen as fewer parcels are delivered in the mixed fleet at higher congestion rates.

When the calculation time is taken into account, it becomes apparent that the algorithm is not efficient in obtaining a result. The computation time of the algorithm for all experiments done on the case data shows the high computational efforts of this model. In the method of Van Duin et al. (2021), the problem is solved on an Intel Xeon(R) Gold 6140 72-core (2.30GHz) CPU with 32 GB RAM. For these experiments, an Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz with 16 GB RAM is used. A comparison of the two shows that the former can generate more than 3.5 times the computing performance of the latter[3]. This does not allow a comparison of the model. Moreover, it is good to mention, because it brings the performance in line with the computing power of the machine used.

This study aims to develop an improved model for the decentralised dynamic distribution of parcels, taking into account its applicability in real cases; it must meet the requirements of acceptable computational complexity, self-organisation capability, and the ability to handle a dynamic sequence of parcels. The experiments provide a proof-of-concept of the method, but at the same time demonstrate its limitation. With the integration of heuristics, the performance can be improved by 4.6% while delivering all parcels. However, the two main limitations are, first, that computation times are still high even when some heuristics are applied. Second, to obtain better results, the dynamic behaviour property must be relaxed to cluster the parcels in advance. Finally, at higher congestion levels, the method is able to achieve feasible results, but the distance to the vehicle fleets increases. For a mixed fleet, not all parcels can be delivered.

[3] `https://www.cpubenchmark.net/compare/Intel-i7-8565U-vs-Intel-Xeon-Gold-6140/3308vs3132`

5.3 Potential Implementation

The introduction of the proposed system requires a radical change from the current way of doing business and handling packages. The steps to achieve such an implementation are therefore unclear. This chapter explores possible ways forward. So far, the state-of-the-art chapter has shown us the current applications and possibilities for a self-organising last-mile system. Most of the literature is conceptual in nature and little empirical research has been done. The simulation model and the application of the algorithm have given practical insights into these theoretical concepts and provided a proof-of-concept of this system, but have shown the limitations for implementation. This section uses the information from the interview that is conducted with an industry expert (see Appendix R) to help shape the implementation steps.

The steps for further development consist of *formulating a clear vision, setting preconditions, considering constraints, identifying key partners, and defining incremental steps to achieve the set vision.* The drivers of current innovation in the parcel sector are increasing customer-driven logistics, larger volumes to be shipped, sustainability, price, speed and visibility in the supply chain. Combine this with the trend towards automation and robotisation and increasing connectivity, and the way is clear for self-organisation. In theory, self-organisation could provide a more dynamic and decentralised solution that, when applied correctly, could offer a more customer-centric approach because the customer can provide the parcel preferences to be taken into account in the auction process. In addition, through better sorting and routing, the system can achieve lower emissions and faster delivery at lower costs. The vision of the self-organising logistics system is in line with the three main characteristics mentioned in the scope of this study; acceptable computational complexity, self-organising capability, and the ability to handle a dynamic sequence of parcels. The system must be able to remain robust in a dynamic environment, be organised in a decentralised way so that the parcels can make their own decisions, and be applicable in situations where no information is available at the time of allocation. Ideally, this would produce a system where each parcel always follows the most efficient route, which would contribute to achieving overall optimality. During the research, it became apparent that designing a fully self-organising system was not within the scope of this study. The focus is on the improvement of efficiency a model that embraces the features of decentral decision making. The experiments have shown that the method can be improved in terms of efficiency and performance, but when placed in the context of the central method it performs poorly.

A precondition would be that the system is applied to ongoing operations for a specific area. A clear decision on the availability of information must be made beforehand. This is to know the dynamic behaviour of the system. The experiments have shown that clustering is beneficial, but is only possible if all parcels are known in advance. An important limitation of this system in implementation is still the computational complexity. The system should be able to handle millions of parcels, but experiments have shown that it already has difficulty with 729 parcels. Another important limitation of the current dynamic method is the situational awareness of the parcel. The parcel industry is a customer-dominated environment driven by the customer's convenience (Quak et al., 2018). In the decentralised method, the parcels are distributed among a number of vehicles, but the actual allocation of the parcels may change over time. Therefore, the system cannot give a clear insight into the delivery times of the parcels. This consideration is also visualised in Figure 34. The final limitation is that this is a completely new system that will probably require a new design of the sorting depot that allows for relocation and dynamic allocation. This new design has yet to be created and tested. The current system already works well. In order to introduce the new system, a convincing pilot project or model must first be carried out showing that it can compete with the current system, otherwise the system will not be implemented at all.

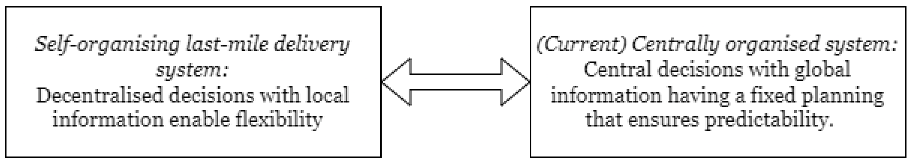


Figure 34: Trade-off between a self-organising and central organised system

The main partners to create this system would initially be Prime Vision and a logistics service provider, who are willing to experiment with this kind of system. Since this system is still at an early stage of development, it is sufficient to test it between these two parties first. If the test and trial system could be applied, a whole range of parties could be considered for implementation. You would need new partners to make new sorting machines, you would need new partners to make a handheld app and so on. A whole chain of partners is needed.

To bridge the gap between long-term vision and short-term operations, step-by-step action is needed to realise such a system. Because of the fundamental change in the system, a complete overhaul of the decision-making and physical centres is required. This makes it difficult to define clear steps for implementation. The current algorithm has shown that it enables the decision-making software to generate a feasible parcel allocation and routing. Turning a conceptual idea and model into practice is another story. The first step is to provide a proof-of-concept that shows the potential benefits and limitations of the system. This thesis contributes to that. If the proof-of-concept is successful, the next step would be for decision-making software to start thinking from a more decentralised perspective, instead of focusing on batches of parcels that are sent out, software developers could define parcels as separate entities that interact with their environment. This should lead to physical requirements for the depots and an initial physical design for the system. A simulation model showing the physical design in combination with the agent-based decision model would help in understanding the system. The model would exist of a discrete sequence of events with agents switching between different states. By feeding this model with real-life data, insight should be gained into the performance of the system (e.g. throughput times, throughput and vehicle utilisation). If the simulation model is able to achieve the desired results, the next step is to create a physical test facility that demonstrates the practical limitations present. If successful, the system can be tested for less critical postal moments, for example, evening delivery. The final step would be the full implementation of the system in a sorting depot.

5.4 Summary

The application of the heuristics for the test case of random insertion and k-means clustering shows a better performance in terms of distance travelled and calculation time. The swapping did not yield significant benefits. Therefore, the combination of random insertion and the clustering method was chosen as the best combination of heuristics. The results are even better than their separate performance for travel distance and calculation time. For the implementation of a heterogeneous fleet, it could be seen that the number of kilometres travelled increases with a higher number of cargo bikes, but the emissions decrease. With higher congestion rates, the distance travelled also increases. When the combination of the best performing heuristics is applied, the modified version can achieve a 4.6% improvement in vehicle distance, but at the cost of high computation times. Looking at the mixed fleet and congestion, we see that with a mixed fleet of similar capacity, the model is no longer able to deliver all parcels in the area. In case of congestion, all parcels are delivered, but the distance travelled increases with congestion. The experiments demonstrate the improvements and at the same time show the limitations of the method, with the long calculation times, the marginal performance improvements, the limited dynamic behaviour and finally the unfeasible results in case of system changes. When considering the possible application of this method, it can be said that there is still a long way to go. The steps that can be considered for further development are, formulating a clear vision, setting preconditions, considering constraints, identifying key partners, and defining incremental steps towards the vision. One of the most important steps for implementation is to create a convincing model that demonstrates the benefits of the method and its feasibility.

6 Conclusion

This chapter summarises the findings of this study by answering the sub-questions and finally the main research question. This study describes and evaluates the possible improvements of decentralised dynamic parcel logistics for last-mile delivery. A literature study was conducted in the field of last-mile logistics, self-organisation, and auction methods in logistics. This showed that there is a gap in the research on decentralised dynamic parcel delivery methods, and in validating the feasibility of such methods compared to the current centrally organised system. A conceptual model was created to outline the operational optimisation of the agent-based method. The current model was studied, benchmarked and validated. This provided the necessary knowledge to evaluate the results of the experiments carried out for method extensions and changes in the simulation environment. It also showed the limitation of the previously proposed method. The sub-questions are answered separately and the combined insights answer the main research question. This chapter concludes with suggestions for further research.

Sub-questions

1. What is the state of the art of self-organisation and auction methods in last-mile logistics?

Optimisation of last-mile distribution is widely studied because of its inefficiencies. The problems that arise are often related to the NP-hard properties that result from the combinatorial problems of last-mile distribution. To deal with this complexity, the current distribution system performs the allocation and routing of the parcels separately. It first relies on the clustering of postcodes and then a TSP is resolved for each vehicle. Since the delivery zones are often fixed for several weeks, daily fluctuations in the number of parcels can lead to suboptimal utilisation and routing of the vehicles within the delivery fleet. A decentralised dynamic parcel distribution system, inspired by self-organising principles, would allow for combinatorial optimisation and dynamic allocation, thus overcoming the limitation of strict delivery zones. Self-organisation is a broad concept with many different interpretations. In logistics, self-organisation can be defined by decentral control, openness, and intelligence of the agents. These capabilities offer promising opportunities to cope with the increasing complexity and need for robustness of logistics chains. The actual application of self-organisation in logistics systems has only been studied to a limited extent in the literature. There are even fewer examples of a decentralised dynamic assignment method by use of single parcel auctioning to capture these benefits of self-organising logistics. Collaborative logistics auctions can be an inspiration for the current parcel allocation method. Market principles could be applied to enable decentralised parcel allocation. This has proven effective for parcel bundles for competing carriers. The auction system defined for the SOLiD method is distinctive in that it considers allocation within a non-competitive environment for individual parcels. In general, the current gap in research is the lack of implementation of self-organising inspired systems and experiments being done to give insights into the benefits and limitations. Furthermore, there is little or no information on the applicability and improvements of the proposed method for decentralised dynamic packet distribution, where auctions of individual parcels are used for the allocation.

2. What is the effect of alternative routing heuristics on the efficiency of the parcel delivery solution method in terms of computation time, and vehicle mileage?

The analysis of the current method revealed several limitations, including the difficulty of applying the model to real-scale cases, which is due to the computational complexity resulting from the lack of built-in heuristics. Moreover, the dependence of the model's outcome on the input parameters is a limitation. The performance of the considered decentralised base method is mediocre when compared across multiple test instances with the best-known solution and a centralised solver. For the possible improvements of the model, the application of three heuristics is considered, namely random insertion, two-opt swap, and k-means clustering. These are first tested for a smaller test instance, after which the best combination of heuristics is applied to the case data. Application of the heuristics for the test instance of random insertion and k-means clustering shows better performance in terms of distance travelled and computation time. Swapping did not yield any benefits in terms of performance and efficiency. Therefore, the combination of random insertion with a search frequency of 50% and the clustering method was chosen. The results of the combined method are better than those of their separate applications in terms of performance for travel distance and computation time. The combined method was able to achieve significantly better computation times with a 40% faster run time. In general, the heuristics were able to improve the performance of the model, but the improvements in travel distance were minimal. When the combination of the best performing heuristics is applied to the case data, a 4.6% improvement in vehicle travel distance can be achieved compared to the original SOLiD method, but still with high computation times. Comparison with an external centrally organised OR solver shows that the decentralised method might be improved but does not perform well, as the OR solver can find a better solution in a fraction of the time required by the improved decentralised auction method. Overall, it can be seen that the implementations of the heuristics have improved the efficiency of the solution method, but when placed in the context of a centrally organised solver, this is of no significance.

3. What will be the effect of a mixed fleet of vans and cargo bikes for the parcel delivery system in terms of calculation time, vehicle mileage, and emissions?

The original algorithm has not been tested on a heterogeneous base delivery fleet. A mixed fleet changes the simulation environment of the algorithm with respect to the implementation of different vehicle types with lower vehicle speed and capacity. The fleet is adjusted for three types of combinations of different amounts of vans and cargo bikes in the test case and the case data. The self-organising principles would suggest that the method can adapt to changes in the simulation environment, as all parcels can choose their best mode of transport. However, it could be observed that for a mixed fleet with similar capacity, the model is no longer able to deliver all parcels with a high number of cargo bikes in the fleet. In general, it was found that for a larger number of vehicles with lower capacity, routing is less efficient. On the other hand, the integration of the cargo bikes shows that for the test case and the case data, the emissions decrease with a higher number of cargo bikes. This indicates a trade-off between efficient routing and greener transport. The calculation time did not increase significantly for the inclusion of more vehicles.

4. *How will the adapted last-mile delivery solution method perform under different conditions of congestion?*

Congestion is a common disturbance factor in everyday last-mile delivery. To account for congestion, three congestion factors are chosen. A situation with free-flow (26 km/h), light congestion (17 km/h), and heavy congestion (13 km/h). The congestion influences the time constraint present for a vehicle to accept a parcel. More congestion means fewer possibilities for a vehicle to accept a parcel for insertion in its route. For the original solution method, the model is not affected by the inclusion of congestion. For the modified solution method, congestion causes a slight increase in travel distance. Emissions increase accordingly with congestion, due to the direct relationship between distance travelled and emissions. The calculation time is not significantly affected by the increase in congestion. For the case data, the inclusion of congestion has a more significant effect on the travel distance and the corresponding emissions. The higher the congestion the less efficient the routing. In contrast to the mixed fleet, the change in the simulation environment due to the addition of congestion does not affect the ability of the vehicles to deliver all parcels in a feasible manner. When the combination of congestion and a mixed fleet is considered, it can be observed that the modified method in this study produces more infeasible results than the original method for the test case. Moreover, it can also be observed that for fleet combinations with more congestion more parcels are not allocated for the test case as well as the real-scale case. Due to the number of unassigned parcels, the travel distances are no longer comparable.

5. *What are possible other extensions of the decentralised dynamic parcel distribution solution method and what are the steps to implement this technology in last-mile logistics?*

To put the concept of self-organising last-mile parcels into practice, a convincing proof-of-concept must be presented. The implementation depends on providing evidence of the benefits of such a system. Considering the found limitations of the method and the poor performance, the first decision should be whether it is desirable to continue with this method. To overcome the limitations for the algorithm itself, there are many areas where it can be improved. One might consider applying a set of known heuristics in literature to make the method more efficient in finding a solution. A lot of research has been done on meta-heuristics. The ant colony optimisation algorithms could be an interesting extension to reduce the computational complexity of routing. Another interesting extension is the integration of a genetic algorithm for route optimisation. To make the model more realistic, fluctuating congestion can be used to simulate daytime and peak-hour traffic. In addition, replenishment could be an important addition. More at the level of the algorithm, by working with classes instead of updating lists, more insight can be created into what happens to the agents during a simulation run. In addition, alterations to the code (e.g. NumPy arrays instead of lists) can be applied for faster processing.

Steps for implementation are that first a clear vision must be developed that is very clear about the desired scope and requirements that the system must meet. The definition of self-organisation can be vague and is interpreted differently by different studies. Setting clear requirements helps in designing the system, and helps in verification afterwards. If not all requirements can be met (e.g. openness, intelligence and decentralised), the desired benefits may not be realised. The next step is the limitations, these are important to consider to be able to apply the method in practice. The experiments show that this system performs poorly in contrast with a centrally organised system. Furthermore, the computational complexity is unmanageable for large instances. Moreover, it is questionable what the trade-offs are in adopting this method. One of the main limitations that can already be identified is the inability to provide the customer with information on the situation and status of his parcels and to give a clear delivery date. The next step for implementation is to identify the key partners. For this first round of development, it is enough for the logistics service provider to cooperate with Prime Vision. For later phases, a whole chain of partners should be considered. Finally, precise steps in the work process must be outlined. When all the previous steps have been taken, a more structured development can be achieved.

Main research question

To what extent can the efficiency of a decentralised dynamic parcel distribution method be improved by applying different heuristics and adapting the simulation environment with congestion and a mixed fleet?

The current parcel distribution system is based on delivery zones that are fixed for several weeks; daily fluctuations in the number of parcels can lead to suboptimal utilisation and routing of vehicles within the delivery fleet. A decentralised dynamic parcel distribution system, inspired by self-organisation principles, would make it possible to overcome the limitation of strict delivery zones. The performance of the considered decentralised base method is mediocre when compared over several test instances with the best-known solution and a centralised solver. For the possible improvements of the model, the application of three heuristics is considered, namely random insertion, two-option swap and k-means clustering. These were found to be essential for improving the computation time. The application of a method of random insertion and clustering was able to reduce the computation time by 40%. However, the improvement in performance with respect to distance travelled was minimal with a 4.6% reduction for the case data. Comparison with an external centrally organised OR solver shows that although the decentralised method can be improved, it does not perform well, as the OR solver can find a better solution in a fraction of the time required for the improved decentralised auction method. The integration of mixed fleet and congestion allowed for further analysis by changing the simulation environment. When analysing the results of the integration of mixed fleet and congestion, it could be seen that for congestion the distance travelled increased. For a mixed fleet, it could be seen that not all parcels were allocated for a larger number of cargo bikes. For the results that were feasible with a mixed fleet, it could be established that the routing was less efficient when more vehicles were involved, however, the implementation of cargo bikes resulted in a decrease in emissions. This indicates a trade-off between efficient routing and greener transport with cargo bikes in the model. Overall, the results show that the efficiency of the decentralised dynamic parcel distribution method can be improved, but when contrasted with a centrally organised method, this improvement becomes of low significance. When considering further development. The steps for developing the model would be to structure the research and development by creating a clear vision and scope, defining the requirements, evaluating the limitations, identifying the key partners and defining step-by-step steps for the actual development of such a system. This study has contributed to the first three steps. It has demonstrated the proof-of-concept and created a more efficient model. However, given the method's limitations, further research should carefully consider whether it is desirable to continue with this specific method.

Further research

It can be said that the method of decentralised dynamic parcel distribution used is not a good method for the distribution of last mile parcels. The advantages that a decentralised dynamic approach through individual parcel auctions could bring are countered by the low efficiency of the model studied in this research. At the academic level, many papers point to the benefits of self-organising principles. It is widely believed in the literature that decentralised control can help improve the robustness and performance of systems. The gap in research that is not addressed in this study is the relationship between the efficiency of a model and its robustness. Further research should focus on uncovering this relationship to demonstrate the true potential of this type of method.

Besides this, further research can be done in four domains, namely providing a categorised overview of the existing literature, secondly, ways of modelling can be further explored, thirdly, limitations of current methods can be overcome, and finally, more tests in real cases. Firstly, the conceptual representation of a self-organising logistics model can be further elaborated and an overview of the benefits and drawbacks of such a model can be made. Given the limited research in this area, it would be desirable to have an overview of all applications of self-organisation in logistics and the levels of self-organisation. Moreover, a categorised overview of different auction methods, with their applicability with respect to the requirements of self-organisation, would help in mapping the topic and understanding its usability. Next, the current method of modelling is limited in its dependence on input data. Further research could explore other, more comprehensive modelling methods that can combine the state change of the agents with the sequence of discrete events in the distribution centres. If such a model could be constructed and applied over a longer simulation period, it would provide more insight into the feasibility of the method and the performance in terms of lead times, tardiness of parcels and utilisation of the available fleet. The third area for further research is the extension of the current method by removing the existing limitations; interesting elements are the integration of a dynamic inflow of parcels, the extension of the fleet with new vehicles such as small electric vans, and adding a maximum range to the vehicles given the electric vehicles. An interesting application would be a hybrid application of the system, i.e. instead of processing all parcels with decentralised dynamic assignment, only partially apply it. For example for the chute with parcels that do not fit within the static range of postal codes, a dynamic allocation can be done. Another interesting topic to investigate further is the application of market forces and auctions for the allocation of parcels by connecting the first and last mile. In such a case, the retailer or customer would register a parcel with its characteristics in the first mile of transport, and the entire route of the package would be auctioned within the capabilities of the logistics provider. This system could include a dynamic price based on the demand for that period. The advantages of such a system would be that in periods of high demand when delivery options are scarce, dynamic pricing could help to achieve more uniform demand. The last area of research is the application of the method to real cases. Many papers describe the conceptual potential of self-organisation, but there are few experiments. By expanding the knowledge base to include the above-mentioned domains, the potential of self-organising principles can be better mapped out.

7 Discussion and Recommendations

The goal of this study was to explore and develop improvements of a decentralised dynamic last-mile parcel delivery system. In this section, it is discussed to what extent the goal is achieved, and what considerations should be taken into account when interpreting the results of this study.

This study contributes to the literature on the analysis of a decentralised auction-based model in logistics and provides insight into its efficiency in terms of performance and computation time by applying it to a real-scale case study. The aim of the method is to provide an improved alternative to the current centralised method to cope with the increasing complexity and size of last-mile logistics networks. Most of the literature related to self-organisation and decentralised control is conceptual in nature. Due to the conceptual nature, the concepts appear to be distant visions of the future. Quak et al. (2018) state that there is currently no clear direction or set of actions to make logistic systems function more according to the principles of self-organisation. The model created provides an quantitative application of the concept of decentralised parcel allocation for sorting and routing, and gives insight into the efficiency of the method. The originally proposed method of decentralised dynamic parcel distribution has been improved to better handle larger data sets by including heuristics in the model; furthermore, the model has been modified to show the influence of congestion and a mixed fleet. The experiments show that this type of model is still undesirable compared to the currently used centralised methods, due to its poor performance and computational effort. One of the main contributions of this study is that it compares the existing method with other known solvers, thus providing a reference point for further research. Moreover, insight is given into the previously undiscussed limitations of the method.

The results of the study show that a single parcel allocation method can be applied to small instances as an alternative to the static centralised method currently used in practice. The study shows with a decrease in vehicle distance of 4.6% for the case data and a reduction in computation times in the test runs of up to 40% that the adjustments made can improve the efficiency of the method. The adjustments to the simulation environment affect the efficiency: congestion leads to less efficient routing and a mixed fleet leads to less efficient routing and unallocated parcels. The results suggest that this method is a promising alternative for last-mile delivery, but as mentioned before, the efficiency of the model with respect to the travel distance is still worse than known methods in the literature as described by Yousefikhoshbakht et al. (2013). Moreover, the computational effort required to run the model for a case data set shows that the model has limited applicability in practice. This disagrees with the conclusion of Van Duin et al. (2021) who state that the method is capable of improving the efficiency of current delivery operations. The outcome of this study supports the theory that one of the main limitations of optimising allocation and routing is due to computational complexity (Buckman, 2018, Kang and Lee, 2018). A comparison with MANETs has already been made in the literature review, the ability of these networks to function with centralised control depends on local interactions between nodes in the system. Hinds et al. (2013) mentions that the disadvantage of local interaction is the low throughput of information that can be achieved. This can also be observed in the method of auctions for matching individual parcels, where each individual parcel is paired based on the currently available transport options. The efficiency of the pairing depends on the previous rounds of pairing and the amount of available options. This is also one of the reasons why a centrally organised system is able to outperform the decentralised system. As seen in the collaborative auction literature, if information is known centrally the more efficient routing between carries can be achieved than for decentral information (Karels et al., 2020).

Berndt (2011) mentions that self-organising principles in logistics offer a solution to the conflicting demands of achieving high operational effectiveness and efficiency while maintaining the ability to adapt to a changing environment. The ability of self-organising principles to improve robustness and flexibility while remaining within acceptable efficiency is what makes it an interesting topic. Meepetchdee and Shah (2007) mention the difficulty of the combination by stating that the goals of design for efficiency and design for robustness are often in conflict. In this study, the current method can be examined on the basis of its efficiency. The possible advantages of distributed control, the robustness of the system, and flexibility of the system have not been addressed in this study. No conclusion can be made regarding them. However, the moderate efficiency provides insights for possible further studies. Knowledge of the limitations of such a decentralised method for parcel distribution may raise the question of to what extent this method is desirable, even though it can possibly achieve higher robustness and flexibility.

The scope of the study is inspired by the three main characteristics of a self-organising logistics system defined by Pan et al. (2016); openness, intelligence and decentral control. Moreover, Buckman (2018) defines two additional requirements for the application of allocation algorithms to be used effectively in real systems: acceptable computational complexity and the ability to process a dynamic sequence of parcels. In evaluating the self-organising capabilities in this study, openness can be verified as the system is able to process vehicles entering and leaving the system during the simulation. It should be noted that this does not happen randomly but at predetermined times. The intelligence element is contained in the auction method, allowing the parcel to choose its assignment. The decentralised organisation is an important specification but it is not fully met. The conceptual model shows that the model consists of three agents, namely the vehicle agent, the parcel agent and the platform agent. It is explained that the platform enables the matching of transport requests and transport offers. However, having the platform means that there must be a central entity that can evaluate the information of all bids. In fully self-organising decentralised systems, the information is only available locally and the agents interact through local interactions. This is an important consideration since many of the proclaimed advantages of self-organising systems revolve around the greater robustness resulting from the absence of a central decision-making system. The ability of the system to handle a dynamic sequence of parcels was also one of the conditions that the system had to fulfil. It can be said that the system is capable of this since changing the sequence of parcels does not significantly affect the outcome of the model. One of the limitations of the method is the dependency on the delivery centres of the vehicles, for test cases, it is good to already assign these delivery locations based on the layout of the nodes. However, this means that information about the delivery locations has to be assessed centrally and in advance, which reduces the self-organisation ability and the dynamic ability. It should be noted that for real applications, the delivery areas can be defined based on historical data, which should make it possible to define the delivery centres without prior knowledge of the entire batch of parcels.

For the experiments, heuristics were applied that consisted of modifications of the existing model. All experiments were evaluated on the distance travelled and the calculation time. The computation time is not always a reliable measure as it depends on the available processing space during the simulation run. Moreover, adding more randomness by using different random seeds means that the result must be evaluated based on the dispersion of the results. Consequently, the computation time for incorporating random insertions would be the total of all runs rather than the average time when compared to a simulation that does not contain a stochastic element. For the experiments, a set of 10 random seeds is used to account for stochasticity; to obtain more reliable results, this number of runs should be increased. For the experiments, the dataset consisting of 729 delivery addresses is assumed to be a real scale case. For distribution in postal distribution centres, this number is considerably higher. This study can provide a solution for this real-life scale test case, but no conclusions can be drawn for the implementation of such a system for a complete distribution centre. Moreover, the model assumes a situation where all parcels are delivered and the delivery time is the same for all deliveries. In a real situation, there are many more uncertainties in the delivery of parcels. One of the limitations of the proposed method is the dependence of the input parameters; the calibration showed a distribution of the results that was not logical. The extreme value test showed that the number of kick-outs and the number of reassignments did not behave as expected. For this study, it is assumed that the input parameters proposed by previous studies would give adequate results. The validity of the method could be extended by testing the effect of varying all input parameters. This would also give more insight into the level of path dependency of this model. Moreover, in this study, the implementation of the model was set to the distribution of last-mile delivery. Due to the magnitude of this problem, many assumptions are made along the way. For a better representation of the real system, a more specific application of such a system could be specified first. As mentioned in the section on future research, it would be interesting to see the effects of this system on the parcels that cannot be sorted directly in the static postal code areas. With regard to the mixed fleet approach, the viability of introducing a different fleet combination depends heavily on the geographical characteristics of the area. Urban areas with narrow streets are ideal for cargo bikes, but less densely populated areas favour the use of vans more. This is something that has not been taken into account in the model.

Finally, building on the work of other master's students has its advantages and some serious disadvantages. The work that the predecessors put into this parcel distribution method paved the way for further exploration of the issues. It provided a pre-set test environment and a model to test. A disadvantage of this situation is that the results of this study will always be limited by the assumptions made along the way. A simulation model is always a simplification of reality, which depends not only on the skill of the person who created the model but also on his/her interpretation of the logic behind the model to which all further extensions are bound.

Recommendations:

On an academic level, it can be argued that the used method of decentralised dynamic parcel distribution is not a good method for the distribution of last mile parcels. The benefits that might result from a decentralised dynamic approach by auctioning individual parcels are accompanied by low efficiency of the model. On an academic level, many papers point out the advantages of self-organising principles. It is widely believed in the literature that decentralised control can help improve the robustness, resilience and performance of systems (Zhang et al., 2016, Serugendo et al., 2003, Pan et al., 2016, Van Duin et al., 2021, Hrabia et al., 2018, De Wolf and Holvoet, 2007). The gap in the research that is not addressed is the relationship between the efficiency of a model and its robustness. Further research should focus on uncovering this interaction to show the true potential of this type of method. Other academic recommendations relate to the future research component described in the previous chapter. On a managerial level, recommendations include defining a smaller scope, evaluating the trade-offs of the self-organising method and reconsidering the logic behind the model with possibly smarter heuristics.

One of the disadvantages of the self-organising concept for last-mile delivery is the large scope of the concepts involved. It involves adequate sorting, planning, routing, transport planning, vehicle availability, etc. It combines the implementation of all these concepts under a sometimes opaque definition of self-organisation, which is interpreted differently in many different research papers. The broad scope of the problem makes it difficult to draw general conclusions about the findings. What might help to overcome this is to focus on the application of self-organisation in only one research domain, for example only planning. Later, these separate methods could be combined into a more complete system. The application and concept of self-organisation should also be seen in the light of its trade-offs. One of the most obvious trade-offs is the situational awareness of the location and status of packages in the system, in a dencentral system, it can be hard to showcase this and provide clear delivery windows to the customer. Another important limitation is that change in large-scale systems is often gradual and evolutionary. New solutions for the distribution of parcels, as described in this study, require drastic changes in the current infrastructure, which limits a gradual transition. To realise the changes in the systems, large investments in the installed base are required. Furthermore, switching to a new type of system entails the risk of disrupting current business continuity. Moreover, logistics systems often involve several parties. Thus, the complexity of integrating this type of method is not only due to the complexity of the system but also to the complexity of obtaining cooperation from its users. The final limitation is that it is unclear whether the supposed benefits of this type of decentralised system can be realised in practice. In general, it should be carefully considered why and if such a system is desirable, even if the results show that the efficiency is far from ideal and with the limitations present. If so, the current model can be further improved by removing some of the limitations, but to further develop the idea of a more decentralised dynamic system, a more comprehensive, realistic agent-based model should be created, this model should include a smarter set of heuristics to improve efficiency. In addition, incorporating a dashboard or visual representation would help people accept the results of the model and understand how it works.

References

Arnold, F., Cardenas, I., Sörensen, K., and Dewulf, W. (2018). Simulation of b2c e-commerce distribution in antwerp using cargo bikes and delivery points. *European transport research review*, 10(1):1–13.

Bae, K.-H., Mustafee, N., Lazarova-Molnar, S., and Zheng, L. (2022). Hybrid modeling of collaborative freight transportation planning using agent-based simulation, auction-based mechanisms, and optimization. *Simulation*, page 00375497221075614.

Bartholdi III, J. J., Eisenstein, D. D., and Lim, Y. F. (2010). Self-organizing logistics systems. *Annual Reviews in Control*, 34(1):111–117.

Berndt, J. O. (2011). Self-organizing logistics process control: an agent-based approach. In *International Conference on Agents and Artificial Intelligence*, pages 397–412. Springer.

Botti, V. and Giret, A. (2008). *Holonic manufacturing systems*. Springer.

Boudoin, D., Morel, C., and Gardat, M. (2014). Supply chains and urban logistics platforms. In *Sustainable urban logistics: Concepts, methods and information systems*, pages 1–20. Springer.

Bousbia, S. and Trentesaux, D. (2002). Self-organization in distributed manufacturing control: state-of-the-art and future trends. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 5, pages 6–pp. IEEE.

Boyer, K. K., Prud'homme, A. M., and Chung, W. (2009). The last mile challenge: evaluating the effects of customer density and delivery window patterns. *Journal of business logistics*, 30(1):185–201.

Braekers, K., Ramaekers, K., and Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99:300–313.

Braquet, M. and Bakolas, E. (2021). Greedy decentralized auction-based task allocation for multi-agent systems. *IFAC-PapersOnLine*, 54(20):675–680.

Buckman, N. (2018). Decentralized task allocation for dynamic, time-sensitive tasks.

Cardenas, I., Borbon-Galvez, Y., Verlinden, T., Van de Voorde, E., Vanelslander, T., and Dewulf, W. (2017). City logistics, urban goods distribution and last mile delivery and collection. *Competition and regulation in network industries*, 18(1-2):22–43.

Čaušević, S., Saxena, K., Warnier, M., Abhyankar, A. R., and Brazier, F. M. (2021). Energy resilience through self-organization during widespread power outages. *Sustainable and Resilient Infrastructure*, 6(5):300–314.

Crowe, S., Cresswell, K., Robertson, A., Huby, G., Avery, A., and Sheikh, A. (2011). The case study approach. *BMC medical research methodology*, 11(1):1–9.

Dantzig, G. B. and Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1):80–91.

De Wolf, T. and Holvoet, T. (2007). Designing self-organising emergent systems based on information flows and feedback-loops. In *First International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2007)*, pages 295–298. IEEE.

Dias, M. B. and Stentz, A. (2000). A free market architecture for distributed control of a multirobot system.

Dorigo, M., Birattari, M., and Stutzle, T. (2006). Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39.

Echelmeyer, W., Kirchheim, A., and Wellbrock, E. (2008). Robotics-logistics: Challenges for automation of logistic processes. In *2008 IEEE International Conference on Automation and Logistics*, pages 2099–2103. IEEE.

Ehsanfar, A. and Grogan, P. T. (2020). Auction-based algorithms for routing and task scheduling in federated networks. *Journal of Network and Systems Management*, 28(2):271–297.

Eksioglu, B., Vural, A. V., and Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4):1472–1483.

Fleischer, W., Graf, C., and Lange, T. (2020). Supply chain of the future: Key principles in building an omnichannel distribution network.

Gansterer, M., Hartl, R. F., and Savelsbergh, M. (2020a). The value of information in auction-based carrier collaborations. *International Journal of Production Economics*, 221:107485.

Gansterer, M., Hartl, R. F., and Sörensen, K. (2020b). Pushing frontiers in auction-based transport collaborations. *Omega*, 94:102042.

Ghosh, S., Pal, A., Kumar, P., Ojha, A., Paranjape, A., Barat, S., and Khadilkar, H. (2021). A simulation driven optimization algorithm for scheduling sorting center operations. *arXiv preprint arXiv:2112.04567*.

Gorodetskii, V. (2012). Self-organization and multiagent systems: I. models of multiagent self-organization. *Journal of Computer and Systems Sciences International*, 51(2):256–281.

Gruber, J., Ehrler, V., and Lenz, B. (2013). Technical potential and user requirements for the implementation of electric cargo bikes in courier logistics services. In *13th World Conference on Transport Research (WCTR)*.

Hinds, A., Ngulube, M., Zhu, S., and Al-Aqrabi, H. (2013). A review of routing protocols for mobile ad-hoc networks (manet). *International journal of information and education technology*, 3(1):1.

Hrabia, C.-E., Lützenberger, M., and Albayrak, S. (2018). Towards adaptive multi-robot systems: self-organization and self-adaptation. *The Knowledge Engineering Review*, 33.

Junjie, P. and Dingwei, W. (2006). An ant colony optimization algorithm for multiple travelling salesman problem. In *First International Conference on Innovative Computing, Information and Control-Volume I (ICICIC'06)*, volume 1, pages 210–213. IEEE.

Kang, H.-Y. and Lee, A. H. (2018). An enhanced approach for the multiple vehicle routing problem with heterogeneous vehicles and a soft time window. *Symmetry*, 10(11):650.

Karels, V. C., Veelenturf, L. P., and Van Woensel, T. (2020). An auction for collaborative vehicle routing: Models and algorithms. *EURO Journal on Transportation and Logistics*, 9(2):100009.

Lee, J. H. and Kim, K. H. (2015). An auction-based dispatching method for an electronic brokerage of truckload vehicles. *Industrial Engineering and Management Systems*, 14(1):32–43.

Macioszek, E. (2017). First and last mile delivery–problems and issues. In *Scientific and technical conference transport systems theory and practice*, pages 147–154. Springer.

Meepetchdee, Y. and Shah, N. (2007). Logistical network design with robustness and complexity considerations. *International Journal of Physical Distribution & Logistics Management*.

Montreuil, B. (2011). Toward a physical internet: meeting the global logistics sustainability grand challenge. *Logistics Research*, 3(2):71–87.

Morganti, E., Seidel, S., Blanquart, C., Dablanc, L., and Lenz, B. (2014). The impact of e-commerce on final deliveries: alternative parcel delivery services in france and germany. *Transportation Research Procedia*, 4:178–190.

Nilsson, F. R. (2019). A complexity perspective on logistics management: Rethinking assumptions for the sustainability era. *The International Journal of Logistics Management*.

Nozick, L. K. and Turnquist, M. A. (2001). Inventory, transportation, service quality and the location of distribution centers. *European Journal of Operational Research*, 129(2):362–371. A Global View of Industrial Logistics.

Olsson, J., Hellström, D., and Pålsson, H. (2019). Framework of last mile logistics research: A systematic review of the literature. *Sustainability*, 11(24):7131.

Pan, S., Trentesaux, D., and Sallez, Y. (2016). Specifying self-organising logistics system: openness, intelligence, and decentralised control. In *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*, pages 93–102. Springer.

Phillipson, F. and de Koff, S. (2020). Immediate parcel to vehicle assignment for cross docking in city logistics: A dynamic assignment vehicle routing problem. In *ICORES*, pages 138–142.

Phillipson, F., de Koff, S., van Ommeren, C., and Quak, H. H. (2020). Dynamic assignment vehicle routing problem with generalised capacity and unknown workload. In *ICORES*, pages 329–335.

Quak, H., Kempen, E., and Hopman, M. (2018). Moving towards practical implementation of self-organizing logistics–making small steps in realizing the pi vision by raising awareness. In *IPIC 2018-5th International Physical Internet Conference*, pages 107–119.

Ranieri, L., Digiesi, S., Silvestri, B., and Roccotelli, M. (2018). A review of last mile logistics innovations in an externalities cost reduction vision. *Sustainability*, 10(3):782.

Robinson, S. (1997). Simulation model verification and validation: increasing the users' confidence. In *Proceedings of the 29th conference on Winter simulation*, pages 53–59.

Robinson, S. (2011). Choosing the right model: Conceptual modeling for simulation. In *Proceedings of the 2011 Winter Simulation Conference (WSC)*, pages 1423–1435. IEEE.

Robinson, S. (2014). *Simulation: the practice of model development and use.* Bloomsbury Publishing.

Sargent, R. G. (2010). Verification and validation of simulation models. In *Proceedings of the 2010 winter simulation conference*, pages 166–183. IEEE.

Schliwa, G., Armitage, R., Aziz, S., Evans, J., and Rhoades, J. (2015). Sustainable city logistics—making cargo cycles viable for urban freight transport. *Research in Transportation Business & Management*, 15:50–57.

Serugendo, G. D. M., Foukia, N., Hassas, S., Karageorgos, A., Mostéfaoui, S. K., Rana, O. F., Ulieru, M., Valckenaers, P., and Van Aart, C. (2003). Self-organisation: Paradigms and applications. In *International Workshop on Engineering Self-Organising Applications*, pages 1–19. Springer.

Tang, L., Liu, J., Rong, A., and Yang, Z. (2000). A multiple traveling salesman problem model for hot rolling scheduling in shanghai baoshan iron & steel complex. *European Journal of Operational Research*, 124(2):267–282.

Valckenaers, P., Van Brussel, H., and Holvoet, T. (2008). Fundamentals of holonic systems and their implications for self-adaptive and self-organizing systems. In *2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops*, pages 168–173. IEEE.

Van Belle, J., Valckenaers, P., Saint Germain, B., Bahtiar, R., and Cattrysse, D. (2011). Bio-inspired coordination and control in self-organizing logistic execution systems. In *2011 9th IEEE International Conference on Industrial Informatics*, pages 713–718. IEEE.

Van Dam, K. H., Nikolic, I., and Lukszo, Z. (2012). *Agent-based modelling of socio-technical systems*, volume 9. Springer Science & Business Media.

Van Duin, J., Tavasszy, L., and Taniguchi, E. (2007). Real time simulation of auctioning and rescheduling processes in hybrid freight markets. *Transportation Research Part B: Methodological*, 41(9):1050–1066.

Van Duin, J., Vlot, T., Tavasszy, L., Duinkerken, M., and van Dijk, B. (2021). Smart method for self-organization in last-mile parcel delivery. *Transportation Research Record*, 2675(4):260–270.

Viscido, S. V., Parrish, J. K., and Grünbaum, D. (2004). Individual behavior and emergent properties of fish schools: a comparison of observation and theory. *Marine Ecology Progress Series*, 273:239–249.

Vlot (2019). Power to the parcel: A method for self-organisation in last-mile parcel delivery. *Master Thesis Delft University of Technology*.

Wellman, M. P., Walsh, W. E., Wurman, P. R., and MacKie-Mason, J. K. (2001). Auction protocols for decentralized scheduling. *Games and economic behavior*, 35(1-2):271–303.

Winkelhaus, S. and Grosse, E. H. (2020). Logistics 4.0: a systematic review towards a new logistics system. *International Journal of Production Research*, 58(1):18–43.

Yavas, V. and Ozkan-Ozen, Y. D. (2020). Logistics centers in the new industrial era: A proposed framework for logistics center 4.0. *Transportation Research Part E: Logistics and Transportation Review*, 135:101864.

Yin, R. K. (2009). *Case study research: Design and methods*, volume 5. sage.

Yousefikhoshbakht, M., Didehvar, F., and Rahmati, F. (2013). Modification of the ant colony optimization for solving the multiple traveling salesman problem. *Romanian Journal of Information Science and Technology*, 16(1):65–80.

Yousefikhoshbakht, M. and Sedighpour, M. (2012). A combination of sweep algorithm and elite ant colony optimization for solving the multiple traveling salesman problem. *Proceedings of the Romanian academy A*, 13(4):295–302.

Zhang, Y., Qian, C., Lv, J., and Liu, Y. (2016). Agent and cyber-physical system based self-organizing and self-adaptive intelligent shopfloor. *IEEE Transactions on Industrial Informatics*, 13(2):737–747.

A Vehicle routing problem

In the last 60 years, research has been conducted on the problem of vehicle routing. Dantzig and Ramser (1959) were the first to address the problem of truck routing and modelled how a truck fleet could serve several gas stations. However, current models of vehicle routing differ greatly from those introduced by Dantzig and Ramser (1959), as they attempt to incorporate real-life complexities (Braekers et al., 2016). The goal of these vehicle routing problems is to find optimal delivery routes (Eksioglu et al., 2009). In a classical vehicle routing problem (VRP), each vehicle travels only one route, there is a homogeneous fleet, and there is only one central depot (Braekers et al., 2016). There are a wide variety of variants to extend this classical model. These features add additional complexity. An important notion to consider is that VRPs are an NP-hard problem, meaning that the required solution time increases greatly with the size of the problem. Heuristics and meta-heuristics are more suitable for these situations. Eksioglu et al. (2009) states that the VRP literature has been growing exponentially at a rate of 6% every year. It is therefore hard to keep track of the developments and the different variants. Braekers et al. (2016) define the following variants:

- *Capacitated VRP*: is the classical VRP, the optimal delivery routes are determined for one vehicle that travels only one route, and each vehicle has the same characteristics, and there is only one central depot.
- *Periodic VRP*: is used when the schedule runs over a period of time and deliveries to the customer may occur on different days.
- *VRP with time windows*: is used when deliveries to a particular customer must occur within a certain time interval that varies from customer to customer.
- *Dynamic VRP*: dynamics is usually related to incoming requests from customers. Little or no information is available on future requests. For parcel sorting, this means that the destination of the parcel is not known until it is assigned to a delivery vehicle.
- *Pickup and delivery problems*: goods must be picked up at a particular location and dropped off at the destination.
- *Vehicle routing with multiple depots*: assumes that multiple depots are geographically distributed across customers.
- *Vehicle routing with split deliveries*: in VRP with split delivery, not every customer needs to be visited exactly once, and split deliveries are allowed.
- *Green vehicle routing*: focuses on overcoming difficulties with green mobility, such as vehicles with limited range and infrastructure for refuelling.

In this study, the vehicle routing problem can be extended to other characteristics consistent with last-mile delivery. The single-depot vehicle routing problem is assessed with the expansion of a mixed fleet and restrictions on time windows for delivery. An example solution to a vehicle routing problem is depicted in Figure 13.

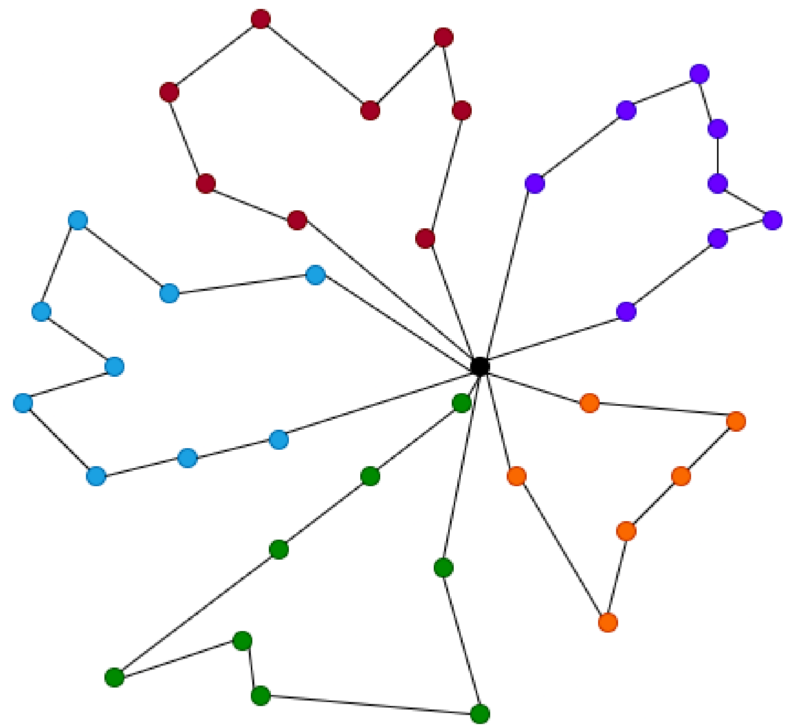


Table 13: Example solution of a vehicle routing problem with one depot and 40 delivery addresses

B Interview expert of the current sorting system

How does the current sorting process work?

In the sorting process, the object in question is the parcel. This parcel has a barcode (ID), an address (postcode + house number + house number suffix), a delivery moment (time moment), and services (signature on receipt, Sunday, evening, registered). You also have events, which can be divided into events that must take place before sorting and events that take place during sorting. The events that must take place before sorting are that a customer of the postal service, for example, bol.com has a number of orders and must report this in advance to the postal service, for example, PostNL. This happens via an API. From that moment on, all information about the parcel is known to the system. If this pre-notification is not made, the system cannot make a sorting decision for this parcel during sorting. What happens then is that OCR and/or Webcoding determine by visual inspection what should happen to the parcel. So this is the event that takes during the sorting.

What are the OCR and Webcoding?

These are tools that allow for visual inspection of the parcel's label. Both tools are provided by Prime Vision.

What are the different stages of sorting?

The sorting process consists of several stages. Almost all postal centres have at least two sorting stages. The first sorting stage is, for example, when bol.com delivers its parcels to a collection depot somewhere of the postal sorting company. That is the first stage. Then it is taken to a sorting centre that is nearby, where they do the first sorting. At PostNL they call this sorting by dispatch. They do not yet sort by house number, but only by postcode. If the parcel is in Amsterdam and the dispatch address is somewhere in Limburg, they send it to a sorting centre near Limburg. There it goes into the sorting machine again. Then it is sorted by house number and addition. Changes may occur between the sorting stages; for example, the end customer may decide that he prefers to receive a parcel the next day. Or the person may say I'll be away for a few days. In that case, the parcel is put on hold for a while. Another change that can occur is that PostNL can reschedule journeys. The sorting depot then receives a message that 'this parcel with this barcode' was scheduled for that journey. But it must now be included in another journey. This may come in just before it is sorted. These are the two disruptive elements that can influence sorting.

Once the package is sorted and assigned to a vehicle, can it be regrouped?

When it comes to sorting in the second stage, this is not the case. Then the parcel is on its way to the final address. If it is already on its way, nothing more can be done. That is where a decentralised system can add something. The central system, like this one, is powerless.

So the role of Prime Vision, is mainly in the system that handles the "no prior notice" packages?

Prime Vision does more, but this is indeed one of the most important. So it provides the OCR and the Webcoding platforms. But Prime Vision also does the sorting. We have many customers, but I am now limiting myself to PostNL in the Netherlands. We also have a system that controls sorting machines, because we also supply the logic for sorting. That system, at least for PostNL, is called SBS, which stands for sorting decision system.

How are the routes of the buses determined?

PostNL has various mail flows. A parcel from a private individual, they call home delivery. That is the standard. Other postal flows are medicines, food, bulky letters, and post boxes (companies). A separate distribution channel does not have to mean separate vans driving around. These distribution channels are used to centrally plan the journeys. This is done by PostNL itself. The journeys are then entered into a system that is managed by Prime Vision. This also shows how central it is (a central program containing all journeys). The rides from all distribution channels are mapped in the central system, which creates sorting tables. These tables enable the sorting in the sorting depot. The sorting tables are uploaded to all sorting depots. In the sorting depots, the app 'table changer' is used, which makes it possible to choose the right sorting table. The trips are independent of Prime Vision and are made in an external planning package. For each distribution channel, Prime Vision receives a file with trips. This determines how the mapping is carried out. The mapping process ensures that per postcode block the parcels go to a certain sorting output. The result of this process is the sorting tables.

How does the sorting process work in the depot?

Sorting takes place in shifts. For example from 02.00 to 04.00 hours. In such a shift, they determine, for example, that they are going to do home distribution and shifts after that mailboxes and foodstuffs. To do this, they select the corresponding tables in the table changer. That sorting table contains the information on what the sorting should answer when it receives a parcel. SBS sees the barcode and gives the sorter the correct sorting direction. After sorting, the parcels fall into roll containers. When the roll container is full, they are moved to the vehicle. Are OCR and web coding also performed in the final sorting stage? No, this is only done in the first step. It could be enabled in the second sorting step, but in 99% of the cases, it makes no sense because it has already been sorted once and therefore already has a known address.

How many sorting depots are there in the Netherlands?

If we only talk about PostNL, there are about 30.

What happens during a scan?

Every time there is a scan for that project, an observation is created. The observation has a special code. Based on this code, PostNL can see and trace what is happening with the parcel, what the status is and what they can still expect.

How many outputs does the sorter have?

The maximum number of exits for the sorter is 50. But PostNL has so many addresses to deliver to that it is impossible to ensure that there is no overlap in the postcode. That has been solved with a tool from Prime Vision, a mobile barcode scanner, and a handheld. To minimise overlap, Prime Vision has a chute mapping system based on linear programming with a lot of input parameters.

How do you view the concept of self-organisation?

The current system is based on a centralised paradigm. Decentralised would mean that whenever a packet has a scanning moment, the packet can broadcast information about what needs to be done so that it can decide on the spot what is the best step to take. In this way, the packet knows where it needs to go and what additional services are required. It may be that the parcel wants to be transported in a CO2-neutral manner. Or the parcel may want to be delivered as quickly as possible. Then there is a digital twin of the system with sorting logic that looks at what is possible in terms of transport. Then a negotiation process can take place. Another problem where self-organisation can be of great help is when a vehicle breaks down, then everything in that truck cannot be delivered. Then a replacement has to be arranged on the spot. Decentralisation can be much more flexible in this respect. Especially when there is a continuous process of bidding and offering. Conceptually, I think it is a better system if it is decentralised, but whether it works out that way in practice is the question. Certainly, in a simulation model, this will probably not be seen.

C Terminology self-organisation

Self-organisation is a concept broadly used in many research fields (Pan et al., 2016). The term is often used in combination or in overlap with the terms self-adaptation and emergence. It can be difficult to define the distinction between these three concepts (Hrabia et al., 2018). This section briefly explains the terms and their dependencies and relationships.

Emergence: A popular example of emergence is fish schools, which use collective intelligence to increase protection from predators for the individuals (Viscido et al., 2004). Emergence is the behaviour, patterns, and traits that emerge from the interaction of local parts of the system. In this system, the individual parts have no knowledge about the macro purpose of the system (Hrabia et al., 2018). Serugendo et al. (2003) describes emergent collective behaviour as an outcome of a process of self-organisation.

Self-adaption: for self-adaptation, single entity systems, apply feedback loops, which monitor the state of parts of the system, analyse it in relation to the intended behaviour and trigger desired responses (Hrabia et al., 2018).

Self-organisation: The term self-organisation is closely related to the previous two concepts. Zhang et al. (2016) describe self-organising systems as being able to allocate resources to specific tasks. While self-adaptation is described as the process of carrying out processes and eliminating disruptions. Serugendo et al. (2003) define self-organising systems as systems that function through local interactions and without central control. Pan et al. (2016) describe a self-organising logistics system, as a system that can operate without significant human intervention or central software control. Furthermore, Pan et al. (2016) define three crucial factors for self-organisation, namely, openness, intelligence, and decentralised control. Most definitions boil down to the fact that an entity can function optimally without or with minimal central control (Bousbia and Trentesaux, 2002, Serugendo et al., 2003, Pan et al., 2016, Zhang et al., 2016, Hrabia et al., 2018). One of the main differences between self-organisation and emergence is that self-organising systems can be aware of the global state of the system (Hrabia et al., 2018). Self-organising systems have been explored since 1953 with studies on the behaviour of insect societies. Many systems in nature demonstrate self-organisation, such as planets, cells, organisms and societies (Serugendo et al., 2003). An example is a biological process in which complex organisms can grow from a single cell. Here, a cell can interact with its environment to influence its activities (Van Belle et al., 2011).

Automation: Automation in the field of logistics mainly refers to the automation of logistics processes in the supply chain (Yavas and Ozkan-Ozen, 2020). The definition of automation in these tasks means that logistics tasks can be performed automatically. This is especially done for systems where intelligence does not play an important role (Echelmeyer et al., 2008). An example is the automation of shop floor logistics through the use of automated guided vehicles (Winkelhaus and Grosse, 2020). However, automated applications have been used in logistics since the 1950s in tasks such as transport, handling, storage and packaging (Echelmeyer et al., 2008). The difference with the previous three concepts is that automated logistics processes work through the coordination of a centrally controlled system. There is little or no intelligence for and between the different automated agents.

D Function diagram overview

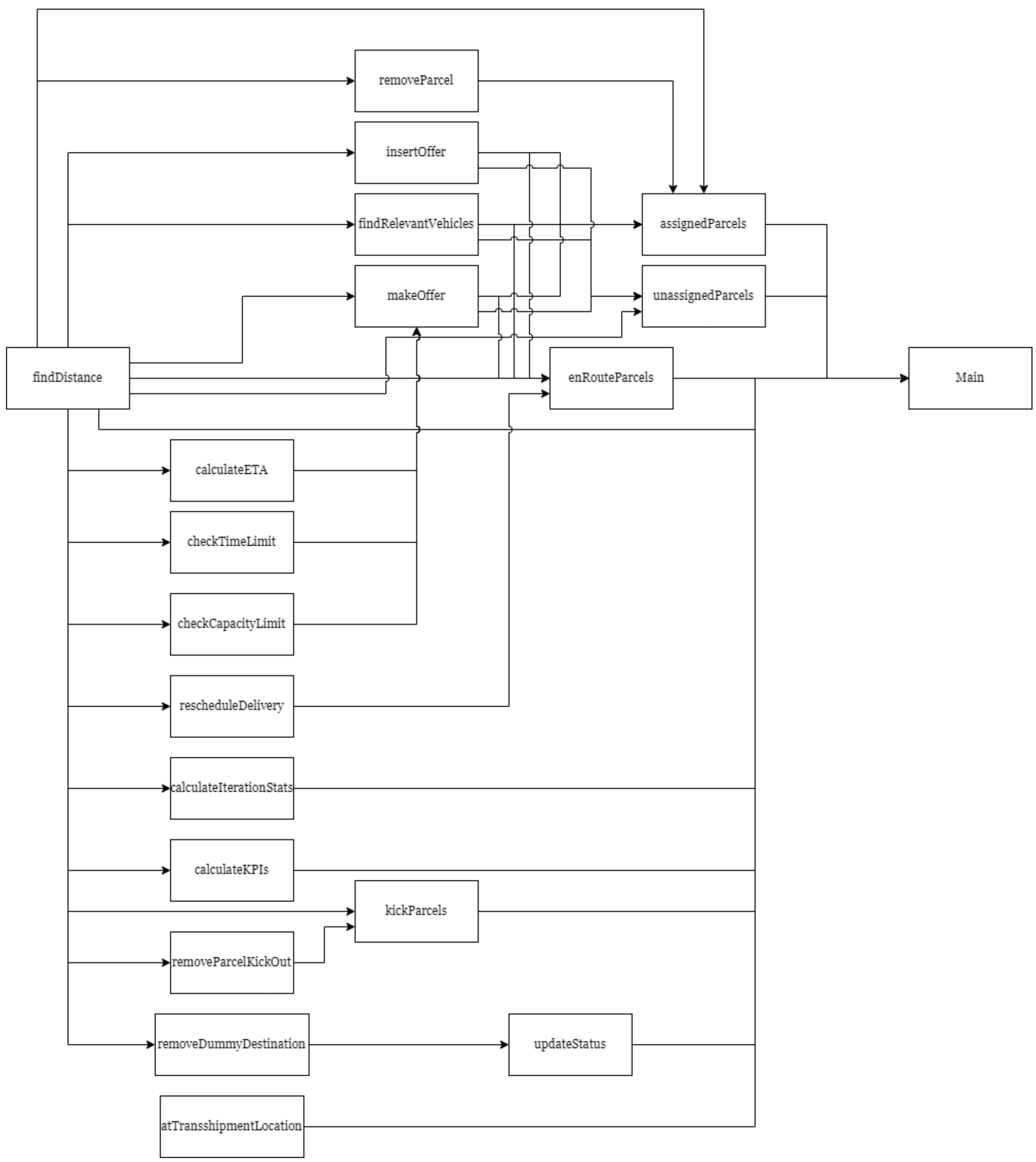


Figure 35: SOLiD algorithm overview based on interrelations functions

E Input data overview

Table 14: Overview of the input data used in the algorithm

Input data		Explanation	Predefined	Updated
Parcels	Parcel ID	Identification number for each parcel	Yes	No
	Status	Indicating the status of the parcel (unassigned, assigned, en route)	No	Yes
	Final destination point	Final delivery point of the parcel	Yes	No
	Priority	Gives parcels priority in auction process	Yes	No
	Location	Start location	Yes	No
	Onboard	Hold information whether the parcel is on board of a vehicle	No	Yes
	Intermediate destination	Hold information if the parcel has an intermediate destination	No	Yes
	Distance	The distance a parcel has traveled	No	Yes
	Cost	Cost of delivery	No	Yes
	Emission	Emission emitted by delivering of the parcel	No	Yes
	Tour	Tour number	Yes	No
	Platform	Used indicate the different logistic providers	Yes	No
Vehicles	Vehicle ID	Identification number for each vehicle	Yes	No
	Status	Indicating the status of the vehicle (en route, at sorting centre)	No	Yes
	Capacity	The amount of parcels that a vehicle is able to carry	Yes	No
	OpCost	Operating cost	Yes	No
	Speed	Speed in km/h per vehicle	Yes	No
	Emission	Co2 emission per km	Yes	No
	part Time	Participation time indicating when occasional drivers will note there presence	Yes	No
	start Time	Start time indicating the time (shift) a vehicle will be available	Yes	No
	end Time	End time indicating the time (shift) a vehicle will be available	Yes	No
	Origin	The origin location of the vehicle	Yes	No
	Destination	The final location of the vehicle	Yes	No
	Itinerary	The itinerary of the vehicle	No	Yes
	Delivery centre	Delivery centre of the vehicle	Yes	No
	Platform	Used indicate the different logistic providers	Yes	No

F Extensions of the model

Vlot (2019) described the following extensions of the model:

- *Improve the offer calculation:* A limitation of the proposed method is the way that additional pick-ups and deliveries added to a vehicle's routes may result in the vehicle being placed further and further away from the centre of its delivery area, as offers are made based on additional distances travelled and do not take into account the centre of the vehicle's delivery area. Therefore, it is recommended to develop a better approach for vehicles to calculate more realistic offers, including additional costs if a vehicle has diverse from its delivery area.

- *Incorporation of delivery time windows:* The delivery times for each individual package could be added as a constraint to obtain a situation with higher customer satisfaction and higher hit rates from the drivers.

- *Financial compensation system:* The financial compensation of combined delivery by multiple parties could be added to the model to explore the financial feasibility of such a system.

- *Tracking of the delivery priority compliance:* The model is not able to check whether a package does indeed meet the delivery priority.

- *Delivery priority refinement:* The current model uses a generalised cost function to account for customer preferences. This cost function could be further extended by indicating to customers how much they are willing to spend for faster delivery.

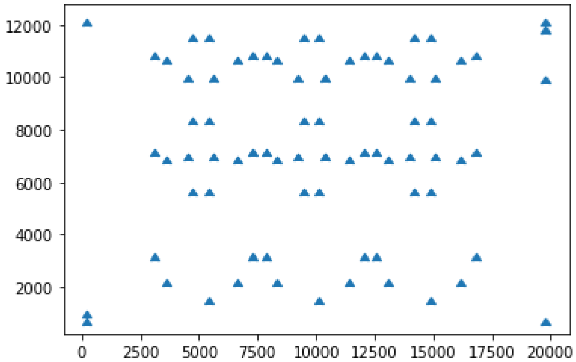
- *Traffic and network data:* In the current model the assumptions regarding routing lengths are highly simplified. Moreover, the varying vehicle speeds and traffic are not incorporated in the model.

- *Agent characteristics:* The characteristics of the agents could be further extended, for example, including the weight of the packages for capacity.

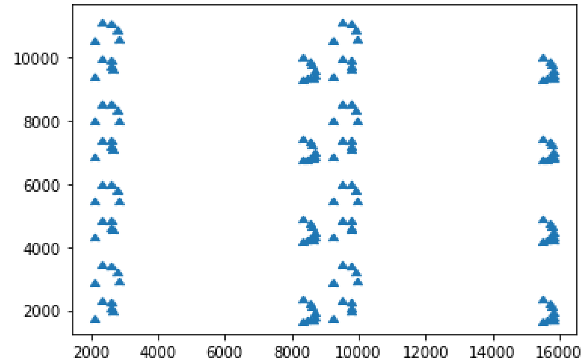
- *Applying the model on different datasets:* For the current model a delivery data sample of 1280 parcels and 11 delivery trucks was used. The model can be tested on different and larger datasets to test its validity.

- *Computation time:* Named by Vlot (2019) as the last limitation, but one of the largest limitations of the model. Running the model for the dataset 1280 parcels and 11 delivery vehicles for a large amount of time (6+ hours).

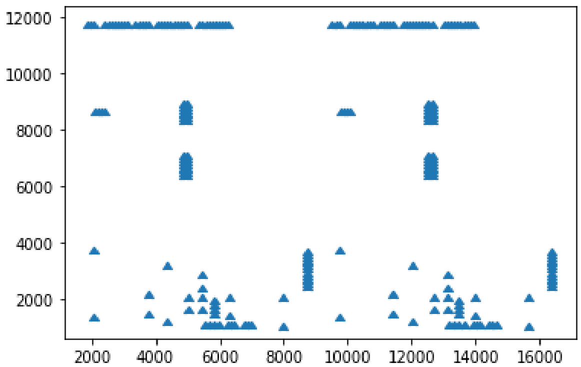
G Graphical overview instances



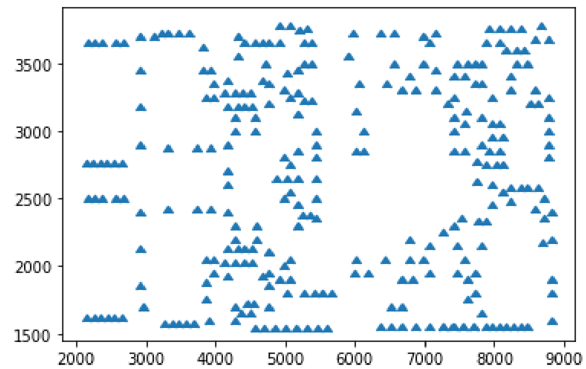
(a) Pr76



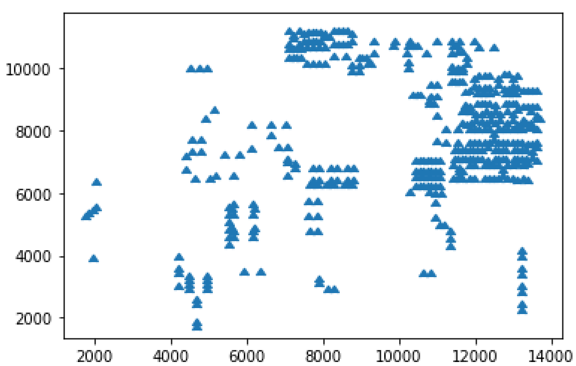
(b) p152



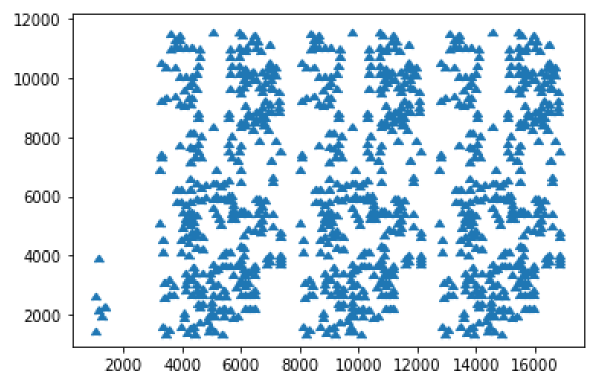
(c) pr226



(d) pr299



(e) pr439



(f) pr1002

Figure 36: Overview instances

H Input parameters

Table 15: Input parameters as described by Vlot (2019) (deterministic)

Parameter	Description	Value
T	Simulation time steps	660
reassignmentIterations	Number of assigned parcel auctions	8
numberParcelRemovals	Number of parcels removed from vehicles itineraries	5
pickupDurations	Duration of a pickup stop	1
deliveryDuration	Duration of a delivery stop	3
maxRelevantVehicles	Vehicles selected for auction	3
transshipmentLocations	Number of transhipment locations	50
transshipmentPenalty	Transshipment penalty cost	0,1

I Parameter variation

Table 16: Parameter variation of the reassignment iteration and the number of kicked parcels for the Pr76 and Pr152 instance

	Reassignment Iterations	Number of kick outs	Distance (m)	RT (sec)		Reassignment Iterations	Number of kick outs	Distance (m)	RT (sec)
Pr76	1	20	233487,00	19,50	Pr152	1	20	159213,30	49,76
	1	10	233487,00	19,47		1	10	238819,95	49,95
	1	5	233487,00	19,47		1	5	238819,95	49,48
	1	1	233487,00	19,31		1	1	238819,95	50,58
	5	20	187227,02	33,74		5	20	206728,94	146,86
	5	10	237181,65	29,39		5	10	199325,28	177,89
	5	5	195460,64	42,16		5	5	184174,32	188,52
	5	1	212228,19	38,30		5	1	171690,64	195,03
	8	20	187136,95	45,07		8	20	224540,27	167,54
	8	10	286117,94	36,04		8	10	170283,12	215,35
	8	5	213897,60	38,31		8	5	168063,02	268,69
	8	1	203193,46	43,36		8	1	166593,58	255,32
	12	20	187136,95	41,73		12	20	198265,00	204,87
	12	10	262053,11	43,01		12	10	190758,26	307,25
	12	5	276339,94	53,35		12	5	183798,80	359,29
	12	1	208894,08	46,53		12	1	166593,58	388,00
	20	20	202218,59	46,95		20	20	252702,66	252,21
	20	10	187136,95	40,57		20	10	211796,89	445,65
	20	5	274827,39	76,55		20	5	181127,45	399,09
	20	1	221684,33	54,37		20	1	166593,58	368,92

J Confidence interval table

Table 17: Distance travelled, average, standard deviation, confidence interval and highest/lowest values for a sample of 20 random seeds for instance pr76, pr152 and pr 226

Randomseed	pr76	pr152	pr226
1	183230,94	147176,40	184853,64
2	182202,24	149368,04	178020,01
3	175748,57	138782,46	188631,01
4	182687,54	141422,72	179373,37
5	163053,44	137790,67	209589,79
6	185631,51	163413,33	178856,30
7	180425,63	144299,09	205425,34
8	177480,88	140179,00	203897,93
9	171657,06	154175,61	182766,58
10	183976,87	157957,46	194350,15
11	180694,27	158972,15	196022,77
12	172501,83	153403,78	179680,37
13	168222,04	147453,60	189812,28
14	171368,58	156400,61	184973,37
15	175437,97	155684,07	183220,23
16	170489,95	152164,74	184477,11
17	187307,49	140540,39	182306,34
18	171246,69	170853,81	247306,29
19	169467,34	137049,67	180803,36
20	183733,61	156169,66	191909,90
Average	176828,22	150162,86	191313,81
Standard deviation	6829,91	9326,29	16118,85
Confidence interval	4489,93	4087,35	7064,27
upper range	269732,26	154250,21	198378,08
Lower range	260752,40	146075,51	184249,54

K Extreme conditions test

Table 18: Extreme condition input parameter test (1)

Pr76			Reassignment iterations				Number of parcel removals				Pick-up durations				deliveryDuration			
	Normal		0		max		0		20		0		20		0		20	
Randomseed	Distance travelled	Unassigned	Distance travelled	Unassigned	Distance travelled	Unassigned	Distance travelled	Unassigned	Distance travelled	Unassigned	Distance travelled	Unassigned	Distance travelled	Unassigned	Distance travelled	Unassigned	Distance travelled	Unassigned
1	183230,94	0	253496	0	67877	67	170530	0	80577	55	238961	0	ZeroDivisionError		184040	0	196356	0
2	182202,24	0	229184	0	26817	73	171195	0	79821	55	225788	0	ZeroDivisionError		168915	0	169303	0
3	175748,57	0	240765	0	47577	70	179467	0	74988	54	241156	0	ZeroDivisionError		171135	0	161814	0
4	182687,54	0	240947	0	69095	63	171654	0	63667	55	243262	0	ZeroDivisionError		171755	0	187544	0
5	163053,44	0	212587	0	54115	70	167849	0	149691	34	225032	0	ZeroDivisionError		166213	0	195805	0
6	185631,51	0	244930	0	IndexError		169406	0	76243	55	232230	0	ZeroDivisionError		182926	0	197567	0
7	180425,63	0	231120	0	63772	66	173728	0	0	76	248313	0	ZeroDivisionError		175167	0	171432	0
8	177480,88	0	197103	0	UnboundLocalError		179019	0	73899	55	232288	0	ZeroDivisionError		163532	0	173438	0
9	171657,06	0	244781	0	38990	71	169044	0	75667	54	220654	0	ZeroDivisionError		173213	0	170985	0
10	183976,87	0	247362	0	IndexError		174224	0	85893	55	219363	0	ZeroDivisionError		175135	0	ZeroDivisionError	0
Average	178609,47		234227,50		52606,14		172611,60		76044,60		232704,70				173203,10		180471,56	

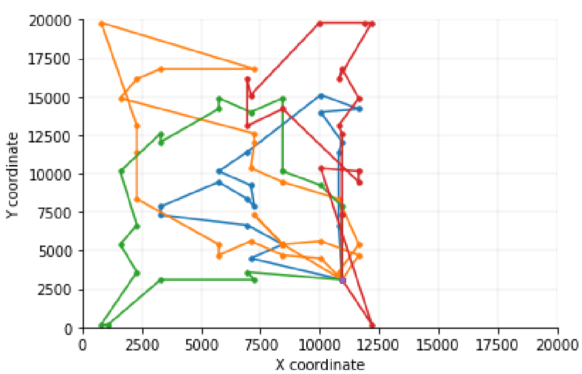
Table 19: Extreme condition input parameter test (2)

Pr76			Max relevant vehicles				Capacity				Speed			
	Normal		0		5		0		1000		0		10000	
Randomseed	Distance travelled	Unassigned	Distance travelled	Unassigned	Distance travelled	Unassigned	Distance travelled	Unassigned	Distance travelled	Unassigned	Distance travelled	Unassigned	Distance travelled	Unassigned
1	183230,94	0	0	76	178820	0	ZeroDivisionError		125492	0	OverflowError		169803	0
2	182202,24	0	0	76	168915	0	ZeroDivisionError		126499	0	OverflowError		168915	0
3	175748,57	0	0	76	171135	0	ZeroDivisionError		120317	0	OverflowError		171135	0
4	182687,54	0	0	76	171755	0	ZeroDivisionError		127689	0	OverflowError		171755	0
5	163053,44	0	0	76	166196	0	ZeroDivisionError		ZeroDivisionError		OverflowError		166196	0
6	185631,51	0	0	76	182926	0	ZeroDivisionError		127162	0	OverflowError		182926	0
7	180425,63	0	0	76	175167	0	ZeroDivisionError		125129	0	OverflowError		166900	0
8	177480,88	0	0	76	163399	0	ZeroDivisionError		ZeroDivisionError		OverflowError		163399	0
9	171657,06	0	0	76	173213	0	ZeroDivisionError		125265	0	OverflowError		173213	0
10	183976,87	0	0	76	170746	0	ZeroDivisionError		134822	0	OverflowError		175135	0
Average	178609,47		0,00		172227,20				126546,88				170937,70	

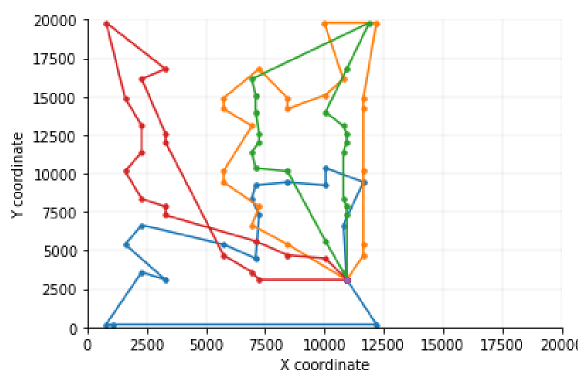
L Overview results random insertion

Table 20: Results overview for random insertion for the distance travelled, runtime, and unassigned parcels for different search percentages for sample sizes of insertion points

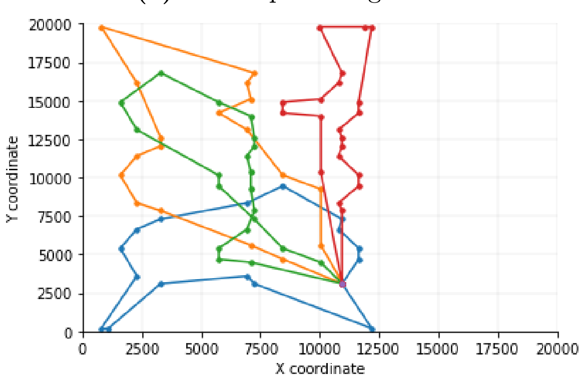
Searchpercentage	25%			50%			75%			100%		
Randomseed	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Distance travelled (m)	Runtime (sec)	Unassigned parcels
1	208879,23	37,54	0	182763,78	45,07	0	169803,71	52,10	0	183230,94	63,94	0
2	260890,30	41,56	0	191534,09	40,42	0	168915,17	44,37	0	182202,24	59,39	0
3	203315,46	46,12	0	179234,40	40,09	0	171135,60	47,61	0	175748,57	54,14	0
4	210302,76	48,92	0	172103,30	43,25	0	171755,01	47,50	0	182687,54	58,68	0
5	209458,74	39,33	0	180710,48	43,24	0	166196,05	60,44	0	163053,44	52,30	0
6	202330,45	39,64	0	159780,95	47,86	0	182926,54	45,48	0	185631,51	56,40	0
7	233929,52	28,26	5	176387,70	49,99	0	166900,25	51,73	0	180425,63	55,92	0
8	224056,10	51,77	0	186304,36	51,27	0	163399,44	48,90	0	177480,88	56,01	0
9	233912,29	48,64	0	176770,04	50,16	0	173213,87	46,34	0	171657,06	57,88	0
10	238593,84	50,94	0	184010,63	43,36	0	175135,71	45,98	0	183976,87	52,71	0
Average	222566,87	43,27		178959,97	45,47		170938,13	49,05		178609,47	56,74	
Best performance	202330,45	28,26		159780,95	40,09		163399,44	44,37		163053,44	52,30	



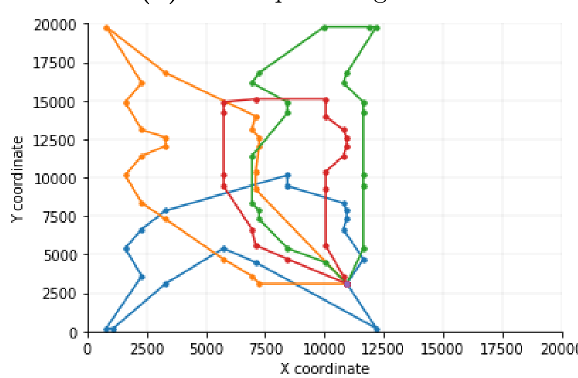
(a) Search percentage 25%



(b) Search percentage 50%



(c) Search percentage 75%



(d) Search percentage 100%

Figure 37: Pr76 example cases for routing for different search percentages

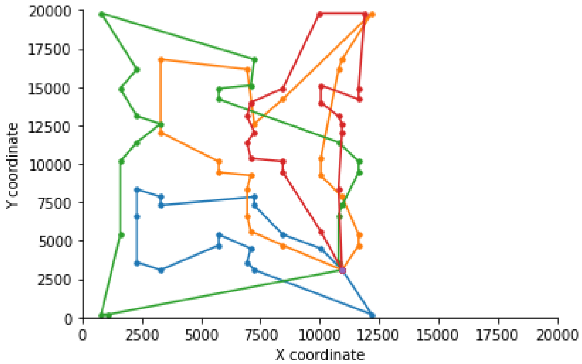
M Overview results swap

Table 21: Results overview for the two-option swap and kick-out for the distance travelled, runtime, and unassigned parcels

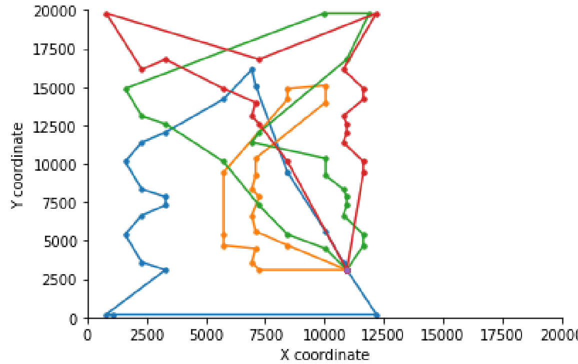
	2 Swap and kick-out			5 Swap and kick-out			8 Swap and kick-out		
Randomseed	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Distance travelled (m)	Runtime (sec)	Unassigned parcels
1	171270,38	42,24	0	184460,78	43,18	0	220798,05	58	0
2	181508,65	52,61	0	202377,78	43,75	0	190199,71	64	0
3	163033,59	46,40	0	196035,24	44,08	0	232704,25	56	0
4	171685,47	66,49	0	187232,78	44,04	0	204598,05	59	0
5	165660,72	62,05	0	172839,03	46,83	0	206317,60	58	0
6	174946,01	61,53	0	195638,94	48,51	0	223840,88	60	0
7	161563,89	54,56	0	183604,29	44,95	0	218827,34	52	0
8	183381,18	46,26	0	173877,53	44,72	0	216471,74	57	0
9	170920,46	47,53	0	189781,91	54,07	0	258690,63	58	0
10	167219,40	52,78	0	179230,36	54,05	0	234852,37	52	0
Average	171118,98	53,24		186507,86	46,82		220730,06	57,53	
Best performance	161563,89	42,24		172839,03	43,18		190199,71	51,88	

Table 22: Results overview for the two-option swap for the distance travelled, runtime, and unassigned parcels

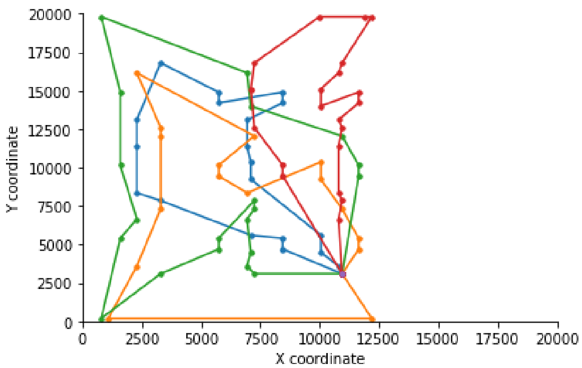
	2 Swap			5 Swap			8 Swap		
Randomseed	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Distance travelled (m)	Runtime (sec)	Unassigned parcels
1	168500,11	64,48	0	165621,36	49,14	0	165212,57	47,22	0
2	168997,49	57,11	0	167190,35	52,51	0	163840,01	53,83	0
3	175826,96	60,94	0	158071,04	57,51	0	168723,31	48,05	0
4	172168,37	58,37	0	181206,63	58,54	0	186720,95	62,18	0
5	162972,82	68,06	0	177115,79	47,37	0	161401,48	64,99	0
6	168803,47	70,88	0	180361,16	48,73	0	166091,10	64,97	0
7	169803,77	68,66	0	165537,00	53,68	0	172515,31	59,23	0
8	173632,00	69,42	0	183540,00	48,74	0	177833,81	69,55	0
9	158862,92	59,75	0	177223,79	48,38	0	178868,74	66,41	0
10	170123,23	53,60	0	170075,73	47,51	0	175767,63	57,36	0
Average	168969,11	63,13		172594,28	51,21		171697,49	59,38	
Best performance	158862,92	53,60		158071,04	47,37		161401,48	47,22	



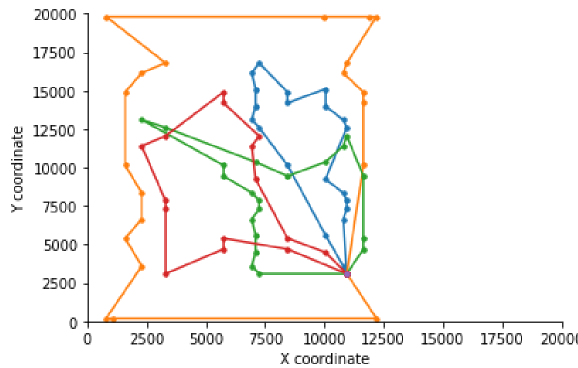
(a) 2 Swap and kick-out



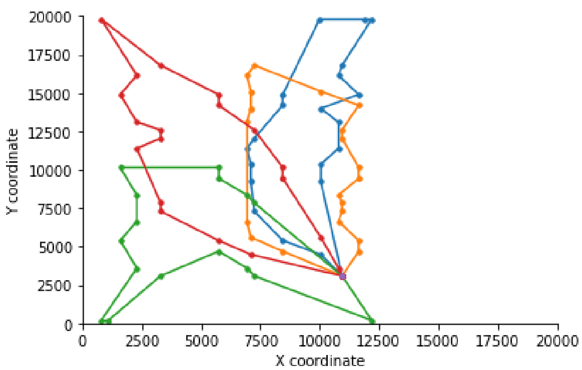
(b) 5 Swap and kick-out



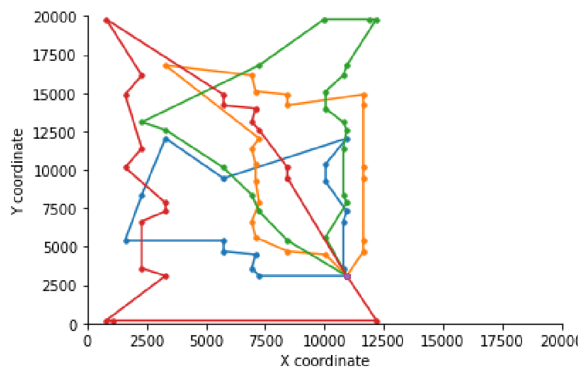
(c) 8 Swap and kick-out



(d) 2 Swap



(e) 5 Swap



(f) 8 Swap

Figure 38: Pr76 example cases for routing for different swap quantities

N Overview results clusters

Table 23: Results overview for the cluster method for the distance travelled, runtime, and unassigned parcels

	Cluster		
Randomseed	Distance travelled (m)	Runtime (sec)	Unassigned parcels
1	166879,12	62,77	0
2	166608,54	63,16	0
3	173572,91	56,52	0
4	168204,37	59,68	0
5	178013,49	54,98	0
6	179059,02	50,49	0
7	162230,87	48,22	0
8	172358,65	49,38	0
9	181413,01	46,39	0
10	158863,28	51,25	0
Average	170720,33	54,28	
Best performance	158863,28	46,39	

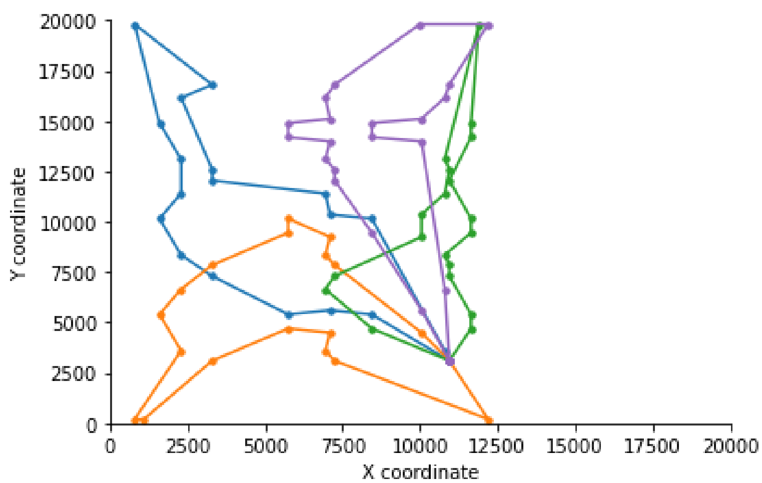
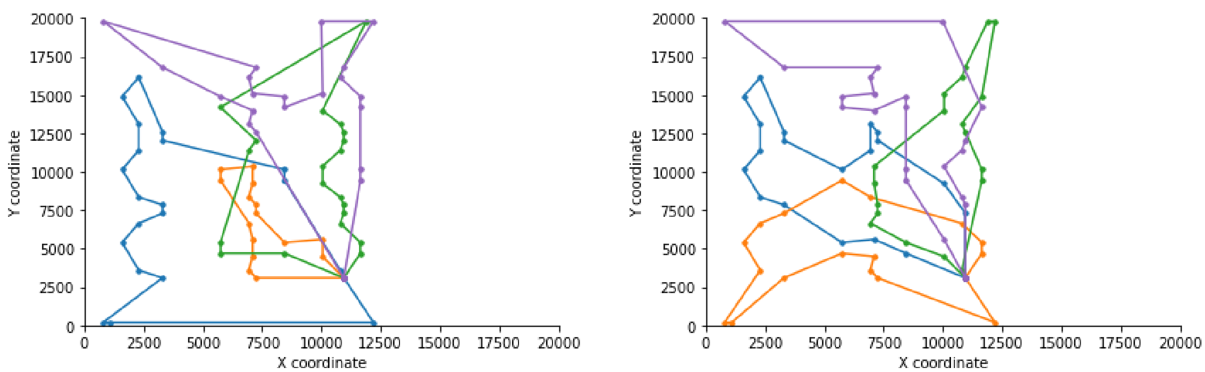


Figure 39: Pr76 example case for clustering method

O Overview results combination of methods

Table 24: Results overview for the combination of the random insertion and cluster method for the distance travelled, runtime, and unassigned parcels

	50% and cluster			75% and cluster			Base case		
Randomseed	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Distance travelled (m)	Runtime (sec)	Unassigned parcels
1	164119,03	33,86	0	178053,63	42	0	183230,94	63,94	0
2	163588,93	30,20	0	180.508	38	0	182202,24	59,39	0
3	170164,15	30,66	0	172.331	37	0	175748,57	54,14	0
4	173377,13	32,76	0	167.533	40	0	182687,54	58,68	0
5	163936,91	33,27	0	173339,82	39	0	163053,44	52,30	0
6	186939,41	34,31	0	172302,69	37	0	185631,51	56,40	0
7	168173,95	32,34	0	166.622	37	0	180425,63	55,92	0
8	162826,52	29,16	0	175.372	43	0	177480,88	56,01	0
9	174198,39	32,67	0	166.235	39	0	171657,06	57,88	0
10	176117,87	33,39	0	169.536	40	0	183976,87	52,71	0
Average	170344,23	32,26		172183,27	39,30		178609,47	56,74	
Best performance	162826,52	29,16		166235,22	36,68		163053,44	52,30	



(a) Search percentage 50% and cluster

(b) Search percentage 75% and cluster

Figure 40: Pr76 example cases for routing for different search percentages and clusters

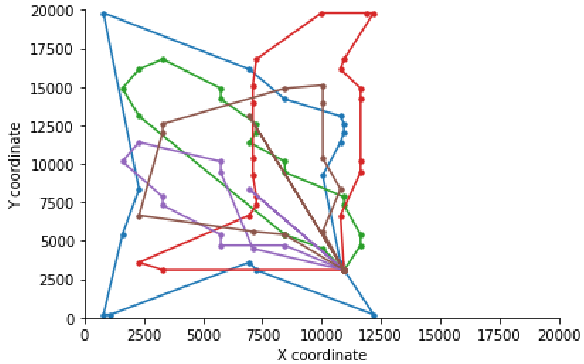
P Overview results mixed fleet

Table 25: Results overview for the mixed fleet and the base case (Pr76) for distance travelled, runtime, unassigned parcels, and emission for different fleet combinations

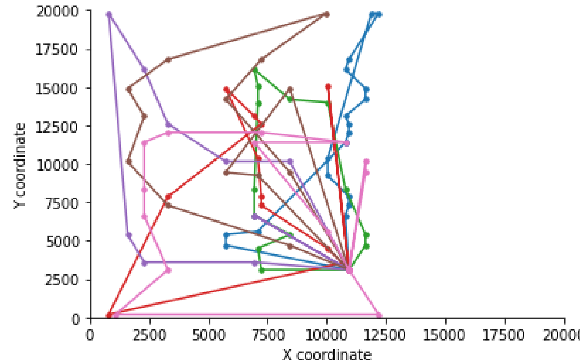
Fleet	Base case: 4 vans 2 cargo-bikes				Base case: 3 vans 4 cargo-bikes				Base case: 2 vans 6 cargo-bikes			
Randomseed	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Emission (gr)	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Emission (gr)	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Emission (gr)
1	270411,60	43,68	0	29567890,1	309855,41	34,11	0	20595673,84	330813,79	33,34	0	9787713,02
2	226678,13	43,62	0	28593260,1	306972,56	39,84	0	20742696,51	353040,89	31,42	0	4580325,80
3	269028,91	43,64	0	28867203,6	291609,75	37,27	0	15151201,01	395760,82	32,92	0	3653246,17
4	269758,14	49,22	0	33113249,3	310042,43	34,87	0	17637015,48	342213,44	33,73	0	3224261,44
5	260024,21	44,89	0	29315325,28	325561,98	40,46	0	19087716,33	358660,90	33,82	0	762000,58
6	264081,54	45,90	0	27571399,04	293102,32	35,13	0	16689220,44	340972,83	30,24	1	4230324,80
7	274498,09	43,65	0	29343349,40	295921,91	35,27	0	19637813,77	355015,16	34,80	0	2698556,65
8	275387,22	45,45	0	29782402,84	307345,44	34,15	0	18443663,43	344064,91	42,94	0	1562646,68
9	282720,79	41,67	0	29355859,8	319621,56	30,92	0	22413578,89	355264,85	31,65	5	3113250,33
10	269787,09	46,10	0	26527830,62	312665,99	33,62	0	20569869,45	358255,95	37,80	0	2460140,26
Average	266237,57	44,78		29203777,01	307269,93	35,56		19096844,92	353406,35	34,27		3607246,57
Best performance	226678,13	41,67		26527830,62	291609,75	30,92		15151201,01	330813,79	30,24		762000,58

Table 26: Results overview for the mixed fleet and combination of methods for distance travelled, runtime, unassigned parcels, and emission for different fleet combinations

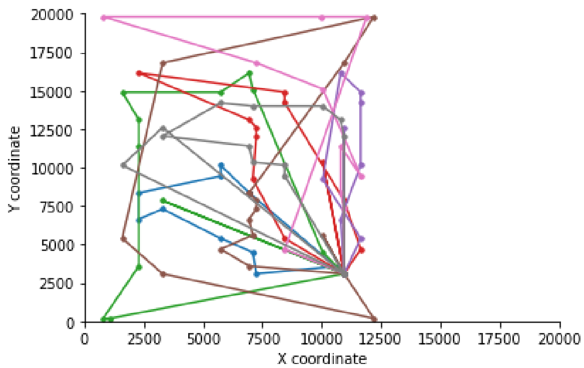
Fleet	4 vans 2 cargo-bikes				3 vans 4 cargo-bikes				2 vans 6 cargo-bikes			
Randomseed	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Emission (gr)	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Emission (gr)	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Emission (gr)
1	257197,00	35,43	0	29479117	296.901	36,25	0	20956565,00	362958,00	21,89	0	9371526
2	256272,00	26,03	0	28119506	286413,00	35,85	0	16938314,81	365826,00	22,21	0	6029144
3	246122,00	26,46	0	27312563	332530,00	38,60	0	17152781,76	391789,00	22,81	0	6854378
4	254867,00	31,70	0	28759139	262178,00	40,88	0	20639462,16	335341,00	28,44	4	12341797
5	242522,00	29,72	0	34084766	319248,00	32,81	0	15724824,92	359922,00	32,17	0	8906410
6	255920,00	28,71	0	28861279	264986,00	38,25	0	20600306,73	340339,00	31,86	5	8145286
7	254122,00	28,21	0	26719942	307463,00	36,45	0	21547674,66	324946,00	27,86	7	10544326
8	252534,00	30,15	0	27845478	319417,00	35,20	0	21000406,45	373155,00	29,84	0	9095526
9	257647,00	28,54	0	30508456	321115,00	34,22	0	14228400,53	374702,00	29,45	0	12042362
10	228202,00	28,14	0	28266482	297475,00	34,57	0	17819778,77	352657,00	28,33	0	13503595
Average	250540,50	29,31		28995672,80	300772,57	36,31		18660851,58	358163,50	27,49		9683435,00
Best performance	228202,00	26,03		26719942,00	262178,00	32,81		14228400,53	324946,00	21,89		6029144,00



(a) 4 vans 2 cargo-bikes



(b) 3 vans 4 cargo-bikes



(c) 2 vans 6 cargo-bikes

Figure 41: Pr76 example cases for routing for different heterogeneous fleet combinations for a combination of heuristics

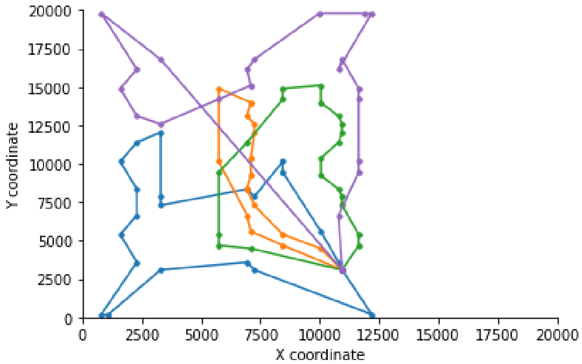
Q Results overview congestion

Table 27: Results overview for congestion and the base case (Pr76) for distance travelled, runtime, and unassigned parcels for different congestion rates

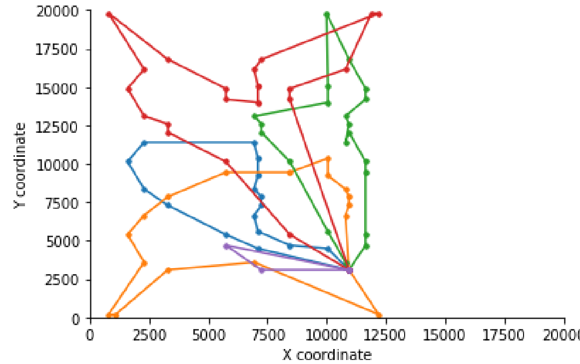
Congestion	Base case (26 km/h)			Base case: 17 km/h			Base case: 13 km/h		
Randomseed	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Distance travelled (m)	Runtime (sec)	Unassigned parcels
1	183230,94	63,94	0	183230,94	52,03	0	183230,94	57,24	0
2	182202,24	59,39	0	182202,24	56,35	0	182202,24	54,98	0
3	175748,57	54,14	0	175748,57	54,40	0	175748,57	59,85	0
4	182687,54	58,68	0	182687,54	55,62	0	182687,54	61,18	0
5	163053,44	52,30	0	163053,44	55,73	0	163053,44	51,31	0
6	185631,51	56,40	0	185631,51	60,00	0	185631,51	63,00	0
7	180425,63	55,92	0	180425,63	55,45	0	180425,63	57,99	0
8	177480,88	56,01	0	177480,88	57,35	0	177480,88	60,08	0
9	171657,06	57,88	0	171657,06	56,84	0	171657,06	52,53	0
10	183976,87	52,71	0	183976,87	58,83	0	183976,87	61,72	0
Average	178609,47	56,74		178609,47	56,26		178609,47	57,99	
Best performance	163053,44	52,30		163053,44	52,03		163053,44	51,31	

Table 28: Overview for different congestion rates for distance travelled, run time, and unassigned parcels

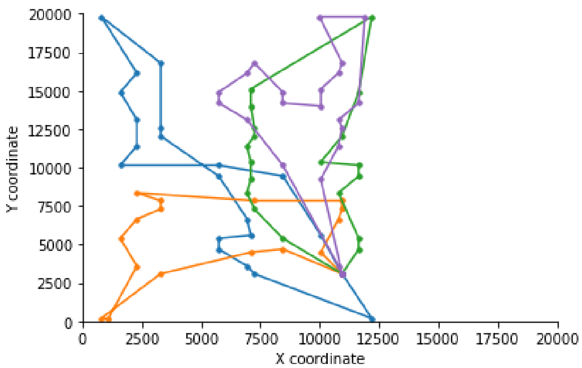
Fleet	26 km/h			17 km/h			13 km/h		
Randomseed	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Distance travelled (m)	Runtime (sec)	Unassigned parcels	Distance travelled (m)	Runtime (sec)	Unassigned parcels
1	164119,03	33,86	0	179854,00	33,36	0	169785,00	30,45	0
2	163588,93	30,20	0	182108,00	31,23	0	172194,00	37,69	0
3	170164,15	30,66	0	183317,00	29,40	0	193808,00	30,84	0
4	173377,13	32,76	0	191574,00	35,33	0	177529,00	37,89	0
5	163936,91	33,27	0	161364,00	31,89	0	168280,00	34,63	0
6	186939,41	34,31	0	162656,00	34,13	0	174418,00	34,59	0
7	168173,95	32,34	0	163686,00	33,02	0	163686,00	34,00	0
8	162826,52	29,16	0	175216,00	34,24	0	175216,00	35,30	0
9	174198,39	32,67	0	156597,00	29,91	0	202916,00	31,80	0
10	176117,87	33,39	0	174859,00	31,81	0	174859,00	33,42	0
Average	170344,23	32,26		173123,10	32,43		177269,10	34,06	
Best performance	162826,52	29,16		156597,00	29,40		163686,00	30,45	



(a) 26 km/h



(b) 17 km/h



(c) 13 km/h

Figure 42: Pr76 example cases for routing for different congestion rates

R Interview business expert Prime Vision

I can identify two major drivers for the parcel industry, namely, increasing development in customer-driven logistics and volumes in this industry increase, am I missing something?

Drivers for innovation in the parcel industry are price, speed, sustainability, one-day delivery, and audibility. The most fundamental drivers are still cost and speed. The lobby for sustainability is strong and logistics service providers want to become more and more sustainable. Another driver over the past five years is one-day delivery. By optimising the current system, companies are getting closer to this goal. Something else that can be considered a secondary driver is audibility. The idea of visibility in the supply chain helps with customer satisfaction, but it is also important for responsibility for damage to packages and to know where the package was. Moreover, the controlling party can see how the network is functioning. During a sustainability audit, for example, all kilometres travelled and the corresponding emissions can be easily displayed.

What does my algorithm currently not have but is of importance to bring such a system into practice?

It is still difficult to understand the conceptual idea of such a system and to translate it into a physical system. What does a distribution centre with such a dynamic sorting system physically look like? How can you physically move a parcel that has already been allocated and is waiting to be sent? And what is the trade-off you make? What is done now is highly efficient for everything up to the last mile. The elephant in the room is the behaviour and design of the depot; to create a self-organising logistics system, a convincing picture of what such a system will physically look like and how it can be implemented must first emerge. Otherwise, this system will not come about at all.

Implementing such a system at once can be a big step, can you think of incremental steps for implementation of this process?

There are steps to set up new sorting/routing systems, but it is unclear how the whole decision-making process of sorting and the physical sorting centre can be changed step by step because it is such a fundamental change. One possibility is to demonstrate it, in a limited production run. For the lower demand periods and perhaps the less critical evening delivery. This is to test it in a lower-risk period, but it does not show the transition needed for the whole system.

One of the important factors of the current system is to know where in the system each parcel is, having self-organising elements could reduce this situational awareness, what do you think of this consideration?

One of the considerations for self-organising systems is the delivery date. Right now, the system is predictable because it has been doing the same thing for 10 to 15 years. In a self-organising system, parcels can be allocated immediately, but they can also be allocated over three days. You don't have that clarity.

The current method considers the dynamic assignment of the parcels. In the current sortation process most information is already available, how would such a dynamic self-

organising system be useful in the current system?

An application of this model could be on the allocation side. This could be in the last mile or the first mile. Moreover, a connection between the last mile and the first mile can be made. In such a situation the retailer talks directly to the system and passes on the characteristics and quantity of the parcels. When the retailer places orders, these orders can then directly be linked to the available fleet.

By connecting the first and last mile when demand is high, you can create scarcity. By linking the first and last miles via an auction system, a market structure can be created in which the delivery prices go up when there is more demand. This results in a more dynamic price range during peaks (e.g. Christmas)?

That is an interesting research topic, but pricing and how to do it for such a system is another project. Besides, at the moment it is mostly settled by contracts.

What partners are necessary to roll out such a system?

At first, it is enough to have Prime Vision working with a logistics provider, but when you think of a real system, it becomes more complicated. You would need new partners to make new sorting machines, you would need new partners to make a handheld app and so on. You would have to create a whole chain of partners.