MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

Winner-takes-all-effects in collaborative logistics

verfasst von / submitted by

Ievgen Gladun, BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2019 / Vienna 2019

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 915
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Betriebswirtschaft
Betreut von / Supervisor:	o. Univ.-Prof. Dipl.-Ing. Dr. Richard Hartl
Mitbetreut von / Co-Supervisor:	

TABLE OF CONTENTS

Acknowledgements. The author would like to give thanks to Ms. Margaretha Gansterer for the support at writing this master thesis as well as to the whole Chair of Production and Operations Management for providing in terms of this specialization not only theoretical knowledge but also practical knowledge which is also applicable in the practice and particularly programming knowledge. During the whole study period I have had complete support of the chair staff at all possible issues.

Abstract

This master thesis contributes to the field of collaborative logistics. Collaborative logistics as an important part of transportation in the sharing economy is currently gaining importance.

The master thesis contains information about the relevant state of the art in the scientific literature. Furthermore, the thesis goes into definitions of different forms of collaborative logistics which are centralized collaborative planning, decentralized planning without auctions and decentralized planning with auctions. Ultimately, a computational study simulates decentralized planning without auctions. The objective of the study was to evaluate auctions using different input data with node and depot coordinates, number of vehicles and maximal travel distance based on different kinds of node selection strategies. The results for all node selection algorithms are evaluated, compared and discussed. The main findings are that collaboration gains (in terms of the distance travelled by all vehicles and carriers) could reach 20%. However, the results vary depending on the kind of node selection applied and the level of competition among the participants.

1. Introduction

It is obvious that today's importance of sharing economy has increased significantly and is now higher than it was ten years ago. Examples of practical use of that are well-known companies and brands with worldwide operations such as Uber, Bla-Bla-Car, Couch surfing, etc. People from different countries got plenty of possibilities to communicate with each other directly and contact a person who can provide services directly, which were earlier accessible only via some chain of intermediaries.

First of all, it is necessary to provide a definition of sharing economy. It is necessary to say that scientific literature does not provide consistent and unambiguous definition what sharing economy is.

Koopman et al. (2015) define sharing economy as any marketplace that brings together distributed networks of individuals to share or exchange otherwise underutilized assets. It encompasses all manner of goods and services shared or exchanged for both monetary and nonmonetary benefit.

Frenken et al. (2015) defines the sharing economy as: *consumers granting each other temporary access to under-utilized physical assets ("idle capacity"), possibly for money*. On the other hand, Meelen and Frenken (2015) highlight fundamental difference between taxi journey performed by Uber, Lyft or Didi and a sharing of journey through BlaBlaCar or another hitchhiking or carpooling platform. The difference lies in the fact that a journey via Uber would not have taken place without preordering. Via preordering consumer creates a new demand and subsequently requires additional capacity. Such an approach is to be understood *as on-demand economy*. By contrast, in the case of hitchhiking/carpooling, a journey would take place in any case, regardless of the participation of another passengers.

Selling goods by consumers to each other is defined as a *second-hand economy*. This should not be understood as a sharing economy as well because consumers grant each other the permanent access, rather than a temporary access to their goods.

In this regard, it's necessary to mention features of sharing economy defined by Frenken et al. (2015). First, sharing economy activities take place in the C2C form. Secondly sharing economy encompasses only providing of temporary access to goods. And ultimately the sharing economy is about efficient use of physical assets and doesn't encompass services.

According to the previous definition, Airbnb should not be considered as part of sharing economy as well because apartments are not granted but rather rented. The problem studied in this master thesis can not also be understood as a sharing economy because it should take place in the B2B sector and it is more about services and not physical goods.

However, not all researchers working in this sphere share this point of view. For instance, Matovska and Share (2015) define sharing economy a *socio-economic ecosystem built around the sharing of human and physical resources. It includes the shared creation, production, distribution, trade and consumption of goods and services by different people and organizations.*

Furthermore, it is necessary to go into advantages of the sharing economy. For a great amount of people, **access to some goods is more preferable than ownership.** Such people don't find it meaningful to spend their lives for acquiring and owning more private property, nor would they take loans and pay interest for that while one can interact with other interested individuals for getting access to this property now and without large financial expenses.

Unused value is a wasted value (Frenken et al., 2015). For example, cars in most of cases are not used with the maximal capacity in terms of space. Sharing economy provides a possibility to find people who can be also interested in the same journey;

Also, the sharing economy can also provide a possibility to overcome **fear to deal with strange people** and **can help** to reach environmental goals because it is based on a sustainable way of consumption (Heinrichs, 2013). Each underconsumption occurring in the case of sharing economy activities reduces negative impact on environment

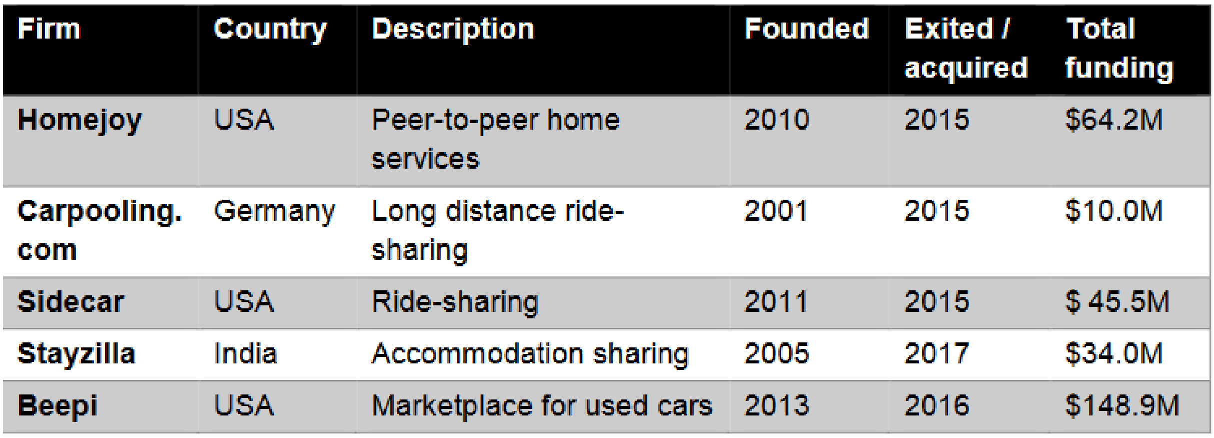
And ultimately the sharing economy increases **quality of service** through rating systems provided by companies involved in the sharing economy (Dyal-Chand, 2015). The quality of ratings may also facilitate and simplify social contacts on sharing platforms. As participants acquire more ratings over time, trust is codified and there is less need for face-to-face interaction.

Nevertheless, there are also drawbacks in the sharing economy. For instance, criticizers say that quality of services cannot be guaranteed because of inability to track all Internet activities of sharing economy participants.

In addition, tax issues are discussed in the science literature. There are a lot of feedbacks given about the negative externalities caused by real estate sharing. Such externalities consist in unfair competition between participants of sharing economy and traditional operators in such branches like tourism, hotel industry, transport. Moreover, there are enough reasons to assume that many providers of services or physical goods in terms of sharing economy avoid paying taxes or do not even know that this kind of activities is taxable, and profits earned are recognized by tax authorities as income.

In the study of Täuscher and Kitzman (2017) authors analyze reasons of failures in the sharing economy. The authors connect success of different projects in the sharing economy with two main factors: scalability and network effects. Scalability is firm's ability to flexibly offer its service or products to a large quantity of customers without additional proportional costs. For example, if a company develops an online platform in order to serve unlimited number of customers like Amazon, profit margins can be increased.

Network effects refer to attractiveness of the service for its (future) users as the network grows. Usually, the larger a network gets the more attractive it becomes. Below in the Figure 1, there is an overview of failure cases in the sharing economy.



Firm	Country	Description	Founded	Exited / acquired	Total funding
Homejoy	USA	Peer-to-peer home services	2010	2015	$64.2M
Carpooling. com	Germany	Long distance ride-sharing	2001	2015	$10.0M
Sidecar	USA	Ride-sharing	2011	2015	$ 45.5M
Stayzilla	India	Accommodation sharing	2005	2017	$34.0M
Beepi	USA	Marketplace for used cars	2013	2016	$148.9M

Figure 1. Overview of failure cases. Täuscher and Kietzmann (2017). Learning from Failures in the Sharing Economy

The author highlights following reasons of failures: low customer lock-in, low control over service quality, high competition for resources necessary for providing of service, low recurrence rate of transactions, changes in legal environment occurred unexpectedly, high costs of developing the networks of demand- and supply-side participants.

We can conclude that there is no uniform definition of the sharing economy in the scientific literature. But there is no debate about underutilized goods as objects what the sharing economy is about. In the chapter we considered also advantages, drawbacks of the sharing economy as well as failure reasons of different companies in terms of the sharing econ

2. Transportation in the Sharing economy

As it was previously mentioned, sharing economy encompasses many spheres of social life. Among others, importance of the Sharing economy increases rapidly in transportation and logistics. Below, there is a short overview of the Sharing economy implementations in the analysed realm:

- **Bike sharing system.** It is a service in which bicycles are placed at the disposal for shared use to individuals on a short term basis for a price or for free;
- **Carpooling.** It means that a person owning a car offers another individuals travelling to the same location to pool both journeys and possibly to share occurring transportation costs;
- **Car sharing** – car rental, mostly for a (very) short time;
- **On-demand delivery services.** It is a way of interaction between individuals where one individual can provide ride services for another one on a profit or non-profit basis. The most well-known example are companies providing food delivery;
- **Flight sharing.** This form of the Sharing economy occurs when pilots-beginners share costs of flights with passengers travelling to the same location;
- **Shared taxi.** Vehicles operate on a fixed or semi-fixed route without schedules and departing usually when all or at least majority of seats is occupied. Such vehicles can do usually a stop on a way in order to pick up or drop off passengers;
- **On-demand delivery services provided by Transportation network companies.** This service encompasses providers of mobility services (platforms), matching passengers with drivers;
- **Vanpool.** Similar to carpooling, but it means more people, typically a group of more than 5 people who commute to and from working place together in a van;
- **Collaborative logistics.** It occurs in the B2B sector when two or more participants pool their logistic capacities in order to reach cost savings for all of them. This

problem is the subject of the master thesis and will be discussed in the next chapter more precisely.

A good overview of the modern shared mobility models is provided by the study of Machado et al. (2018). Below, there is an illustrating schema for this classification.

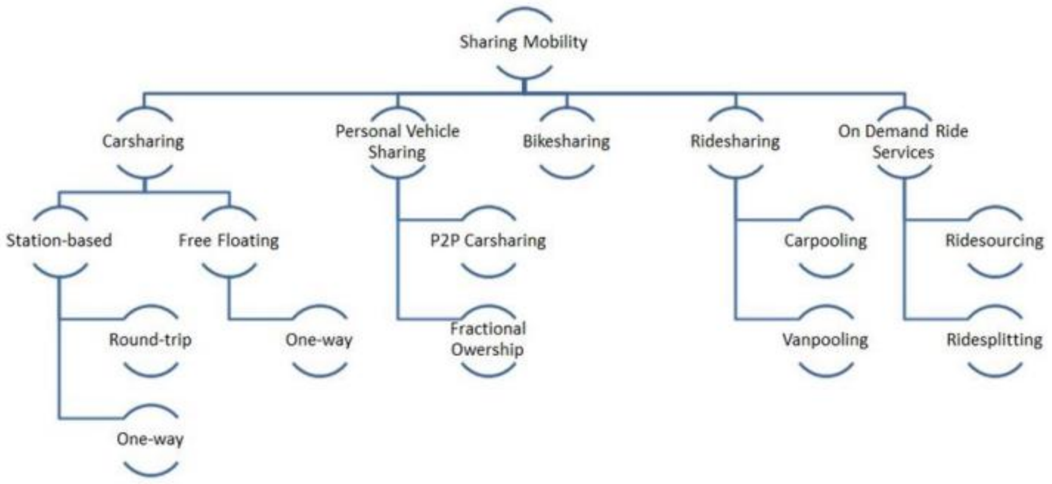


Figure 2. Shared mobility and its modalities. Machado C., De Salles H., Berssaneti F, Quintanilha j., 2018. An Overview of Shared Mobility.

Furthermore, it is necessary to say couple of words about Transportation Network Companies because this issue is closely connected with the topic of the study. Transportation network companies (TNCs, sometimes known as “ride-sourcing” platforms) are *a relatively new urban transportation option that has seen a rapid growth in the last 10 years. Through smartphone applications, TNCs connect individuals willing to pay for a ride with independent drivers willing to provide a ride in their privately owned vehicles.*

TNCs do not use traditional dispatch system where taxi drivers are controlled by a taxi office which dispatch them to customers. Instead of that, they provide an online platform where users can get connected to all drivers in the area. Drivers must be connected to the platform as well. Users can get information about feedbacks of drivers left by previous passengers.

The most well-known TNC is American multinational company Uber. Wallsen (2015) investigated the competitive effects of Uber and its influence on taxicab industry in New York. The study was related to reaching a competitive advantage by the taxicab industry in the times of rapid growth of TNCs. Such competitive advantage might take the form of being more courteous to passengers by switching off the radio if riders don't like that, not using of cell phone while driving, and so on. The difficulty of widespread application of these enhancements and the lack of ride's repetitions blunts such an approach. Nevertheless, even during one journey taxi drivers may behave more politely, in order to get better tips or reduce the chances that a passenger will complain. If such companies as Uber or Lyft have caused such a behaviour and competitive response by the taxicab industry, then even consumers who do not use services of TNCs may benefit.

In the paper, data from New York City and Chicago was assembled in order to test the hypothesis empirically, whether the rise of TNCs has reduced the number of complaints about taxi drivers. Different organizations and associations of taxi drivers in these cities collected much data about taxi rides and complaints. The New York City Taxi and Limousine Commission (TLC) provided the data on every taxi ride in this city from 2010 to 2014 (more than 1 billion observations). NYC's Open Data Project provides data on taxi complaints. The author analyzed also the data about complaints in Chicago. The data analyzed showed that the number of complaints about taxi drivers decreased along with the growth of TNCs. Particularly, in Chicago the growth of Uber was correlated with fewer complaints by taxi passengers about heating and air conditioning, broken credit card machines, and rude drivers.

According to Beard et al (2015), such results can also mean that competition causes some people to switch companies instead of complaining. Some of the decrease in complaints can be explained by the fact that people who would have complained, if there were no competitors of the taxicab industry, chose to switch rather than bother themselves with complaint procedures.

Furthermore, Cramer and Krueger (2016) examine the efficiency of services provided by TNCs taxis by comparing the capacity utilization rate of UberX drivers with that of traditional taxi drivers in five cities. The capacity utilization rate was calculated by the fraction of time a driver has a fare-paying passenger in the car while he or she is working and by the share of total miles that drivers log in while a passenger is in their car. The study revealed that UberX drivers spend a significantly higher fraction of their time while driving and drive a substantially higher share of miles with a passenger in their car than taxi drivers do. The researchers highlight following factors, causing a higher utilization rate of TNCs. First it is more efficient technology to match drivers with passengers. Secondly the authors mentioned a larger scale of Uber than that of traditional taxicab companies, and a larger amount of drivers. Moreover, they went into inefficient taxi regulations which can prevent taxi drivers who drop off a customer in a jurisdiction outside of the one that is allowed by their license from picking up another customer who would be ready to take taxi in that location. And ultimately Uber's flexible labor supply model and so-called surge pricing (Surge pricing automatically goes into effect when there are more riders in a given area than available drivers do. This encourages more drivers to drive to such surge zones over time and shifts rider demand to maintain reliability and restore balance) were pointed out.

Criticism and drawbacks of TNCs are also discussed intensively in the scientific literature. Since the rapid rise TNCs (after about a year after Uber's launch) they were banned or faced with different limitations in many countries. Rogers (2015) emphasized following points of criticism. First, TNCs' drivers often enter the market **without following regulations and legislation** for taxicab industry, what creates a competitive advantage for them. Secondly, TNCs strive for **becoming a monopoly**. Thirdly, TNCs' cars and vehicles are unsafe or underinsured, there are no official requirements to safety and obligatory insurance. Moreover, TNCs can invade customers' privacy, there are a lot of concerns related to data security and data gathering not only by TNCs but also by other multinational companies. Also, TNCs undermine working standards for taxi drivers and the compensation for the drivers is often insufficient because they don't have to follow all requirements and working standards guaranteed by the law all over the world. And ultimately, TNCs enable discrimination by drivers and passengers in a way that both categories can give each other negative reviews or feedbacks for a reason which is not related to the journey.

The study of Ashworth (2018) goes into changes in London's taxicab industry caused by Uber. Advantages of surge pricing, flexible fares and larger range of cars available are highlighted. On the other hand, the author draws attention to protests against Uber held by the taxi industry. The reasons were unequal barriers the market entry, namely traditional taxi drivers have to possess 2 years of experience, they are also restricted on cars which they drive, they have to possess full insurance and provide wheelchair access. Further, it was noted that Uber pays much lower corporation tax as a multinational company (£22,134 in tax on its £866,000 UK profit). All taxi companies of London have to pay taxes as usual national company. This causes an unfair competitive advantage for Uber due to cost savings through taxes it can afford lower fares others can not. Bypassing other requirements such as wheelchair access created also a competitive advantage because Ubers' drivers did not have to bear these costs as well.

According to Ashworth (2018), until October 2016, it was so that Uber's drivers were classified as self-employed. This released Uber from compulsory registration the drivers as employees, paying wage costs, pensions and providing of minimal wage. Currently, Uber has to give its workers the rights mentioned above. The author highlights following changes in the whole industry since the start of Uber. First, an increase in the supply of workers in the taxi market was observed and respectively the competition got higher. Secondly, the researchers pointed out fall of market prices and increasing importance of new technologies in the industry (for example each car provided with a credit and debit card terminals). Thirdly, consumer preferences shifted to Uber and taxi services became much more popular among the people. It causes also a congestion increase due to higher demand (It doesn't correspond with the statement that rise of the sharing economy can contribute to a better environment). Moreover, reduced journey time in average was recorded. There is a competitive advantage for traditional taxi drivers because they are able to use bus lanes. And ultimately obligatory knowledge test on London's topography was introduced.

An example of failure in on-demand ridesharing was investigated in the study of Täuscher and Kitzman (2017). They tracked the case of company Sidecar in order to show risks of the survivorship bias. Sidecar founded in 2011 in San-Francisco was a pioneer TNC at a time when Uber was providing a limousine service (hiring drivers) and Lyft had not yet been founded. After 2012 all three companies adopted the similar business model. But after that Uber and Lyft started to engage new drivers and customers offering new drivers a bonus up to 750 $. It was assumed that firms could achieve a larger profitability once they have gained a superior market position and larger number of drivers. This strategy turned out to be successful and finally in 2015 Sidecar was put out of the market. The management of Sidecar decided to change their business model to a business-to-business delivery service in 2015. In the transportation market, however, Sidecar was not successful as well due to competition from firms possessing larger financial resources and experience on the market. The business was closed months later.

Carpooling is one more form of sharing economy in transportation. In the scientific literature it is not being discussed as intensively as TNCs' activities and ride sharing. A possible explanation is that there are considerably less reasons to accuse providers of these services of law violation. Carpooling has a large importance for such countries as USA where public transport is not as used as private.

In the study of Chan and Shaheen (2011), there were following forms of carpooling proposed. First. acquaintance-based carpooling which occurs among relatives, friends or co-workers. Secondly, organization-based carpooling which requires that riders and drivers join the service with a formal membership or simply visit the organization's website. And ultimately, ad hoc' ridesharing which is a form of carpooling that requires minimal preliminary acquaintance and connection between participants, does not include any obligatory membership and is self-organized.

History and contemporary functioning of carpooling has been discussed in the study of Shaheen et al. (2017). Particularly, it was investigated how level of income and usage of carpooling platforms correlate with each other. BlaBlaCar, the focus of the study, was founded in 2006. It rapidly became the most used carpooling online platform in France and it encompasses about 90% of the market. The quality and efficiency of the online platform for identifying a driver ready to share the ride opportunity is a key success factor. The researchers conducted an online survey launched in May 2013 among the users of BlaBlaCar. The main goal of the survey was to understand the socio-demographic background of BlaBlaCar carpooling customers in France and their overall usage patterns.

The survey consisted of two main parts: one focused on an individual's carpooling experiences and transportation habits and the second part was related to socio-demographic information. Most of the questions were in the form where there were predefined responses; however, certain questions were open. The results showed that the riders using carpooling in France are mostly people with low incomes, whereas people with higher incomes are mainly drivers.

Furthermore, Shaheen, Cohen and Bayen (2018) highlight following social benefits of carpooling. First, it causes reduction of total mileage travelled by cars and also decrease of fuel consumption and greenhouse gas (GHG) emission, reductions in adverse air pollution impacts of low-income, minority. Secondly, it means cost savings for public agencies and employers (in case when employers provide free transportation to the working place for their employees). Thirdly, carpooling reduces need for parking slots. And ultimately Enhanced access, mobility, and economic opportunities for low-income and minority communities who may be unable to afford a vehicle otherwise were highlighted.

It's also necessary to say a couple of words about socio-demographic trends contributing to the growth of carpooling observed by the authors mentioned above. Main trends are increasing availability of local-based mobile services, growth and commodification of on-demand passenger travel, rise of urban congestion, demographic changes, such as rise of life expectancy, an aging population, shifted life milestones (such as marriage), a growing number of employees working outside of office, changing attitudes toward driving and vehicle ownership, continued growth and expansion of many suburbs.

These trends are relevant mostly for the USA, where employees usually get to working places by private transport. Furthermore, the authors discuss possible ways of support for carpooling by employers and government. This, in their opinion, can include tax incentives, providing of cars for carpooling by local governments for two to five passengers who want to carpool but may not have a suitable vehicle. In addition, implementing performance-based contracts for drivers that really contribute to congestion reduction is being also discussed.

Setiffi and Lazzer (2018) performed a study of BlaBlaCar in Italy. Among the reasons encouraging use of carpooling they called, first of all, economic motivation. People want to save time and money and carpooling provides them with such a possibility. Secondly, carpooling forces people to trust strangers and to enter into interaction with them (Parmiggiani 2013). The researchers also claim that other reasons like the awareness of ecological challenges are a secondary motivation.

Täuscher and Kitzman (2017) discussed in their study failure reasons of the largest German carpooling service Carpooling.com. This online platform was founded earlier than BlaBlaCar, in 2005, and was a well-known service with a developed network in Germany. Carpooling.com dominated the peer-to-peer carpooling market in Europe. In 2013, however, the firm imposed a small commission fee, which had to be paid in advance together with ride payments. Previously, it was so that passengers paid drivers' services only after the ride. The customers had not received the changes well and the company lost its popularity. After a dramatic reduction of the network, Carpooling.com was acquired by BlaBlaCar in 2015.

The paper of Cavallini (2017) deals with legal aspects of companies which operate in the sharing economy. The author noted that it is difficult to qualify relationships occurring in the sharing economy as employment or self-employment for direct providers of services. The platform operates as an intermediary helping to match supply and demand, as a provider of services and goods and as an employer establishing the most important rules which govern the transactions. It is also highlighted that companies which provide platforms qualified their relationship with employees as constant collaboration (it is the case, for example, for the Foodora delivery platform). This allows them not to follow working standards and underpay the workers.

Cavallini (2017) also noted that the circumstance that workers of online platforms are free to choose when and where to work does not allow to classify these relationships as labor law-related unambiguously. Services provided by workers could be classified as self-employment services with argumentation that the platform is only an intermediary and its participation ends at the moment when worker is connected with customer. On the other hand, the platform acts as an irreplaceable middleman and the contracts for intermediary services are deeply connected with appropriate self-employment contract. In fact, the contract for intermediary services sets the frame in terms of which several self-employment contracts are arranged by the worker and all possible clients. So, it can be assumed that platforms do not only provide intermediate services; they also provide services to users by connecting them to the workers who would perform the required activity following the rules adopted by the platforms themselves.

Participation of shared taxis in the sharing economy is investigated in the paper of Martinez et al. (2017). In this master thesis, the subject of research was a taxi model that offers lower prices in exchange for sharing the car with another passenger(s) with a similar route. The paper proposes and tests an agent-based simulation model in which a set of rules for space and time matching between a request of a client and the shared taxis is identified. The authors noted that the concept of collective taxis with predefined routes is not new and has been used in many countries like Turkey or countries of the former Soviet Union, where this kind of transport became a wide used alternative to public transport. The advantage is that each passenger pays only a portion of the normal fare. In the paper, this concept was reconceived considering current communication possibilities and GPS satellite system. According to the results, shared taxis have the potential to reduce traffic congestion through a higher occupancy rate, decreased need for parking slots, and causing incentives to decrease the total number of vehicles on roads. Shared taxis are considered as intermediate between private modes and mass transit; they can be regarded as significant components of a comprehensive and efficient transportation system in urban areas.

In a similar vein is the study of Leich and Bischoff (2018) which simulates replacement of conventional bus lines with shared autonomous vehicles of different sizes and capacities in a suburban area of Berlin. The researchers used MATSim (multiagent transport simulation). MATSim is an open-source framework to implement large-scale agent-based transport simulations, where a large number of individuals (so-called “agents”) are simulated (Horni et al. (2016)). The authors took into consideration a real district in Berlin with low density of population and the existing bus routes operating there. One of the main assumptions was that shared taxis could reduce costs per passenger due to lower volume and fuel consumption. Taxis were operating in the door-to-door mode, departure and travel times could vary because a taxi could depart only if all seats were occupied.

The resulting benefits were fewer than expected. Total costs of exploitation could be decreased. Nevertheless, due to the door-to-door operation, walk distances were reduced, detours were needed so that the total travelled distance could soar. Departure and travel times were unpredictable.

Finally, within this chapter, it is necessary to say a couple of words about car sharing and its forms provided by the study of Machado et al. (2017). First, it is **Station-Based Round-Trip Car sharing.** Vehicle is picked up at a designated station and must be returned to the same place. Payment takes place on an hourly basis. Secondly, researchers pointed out **Station-Based One-Way Car sharing.** This mode is similar to the previous one except the circumstance that the vehicle does not have to be returned to the same station. Typically, payment is on a minute basis. And ultimately it is necessary to define **Free-Floating One Way Carsharing.** According to this model, vehicle can be both picked up and dropped off inside of designated area. There are no specific stations and even if the customer is outside of a designated area, the vehicle can be parked using certain parking spaces provided by local authorities.

Heilig et al. (2018) highlight that in the last few years car sharing has been rising rapidly. In Germany in 2016, for instance, station-based car sharing (both in Station-Based Round-Trip and Station-Based One-Way mode) recorded an 18.8% increase in customers, whereas free-floating car sharing had a 51.0% increase.

In this chapter different forms of sharing economy impacts in transportation were considered. We can conclude that some of them like carpooling are not widespread all over the world but rather in countries where certain economic prerequisites exist.

3. Collaborative logistics

In the scientific literature it is also possible to find the concept of “collaborative vehicle routing”. It is defined as *all kinds of cooperations which are intended to increase the efficiency of vehicle fleet operations* (Gansterer and Hartl (2018)).

Collaborative logistics is considered in this master thesis as a part of sharing economy. Gansterer and Hartl (2018) define horizontal cooperation in logistics as coalition of carriers which can get together in order to perform part of their operations jointly.

Analysing the relevant scientific literature, it’s possible to say that collaboration in logistics is considered as highly profitable in terms of economy, infrastructure and ecology. Nevertheless, in highly competitive environment it could be difficult to achieve the revelation of all relevant information by possible participants. This circumstance can exacerbate the search of optimal solution.

Respective literature highlights following advantages of collaborative logistics. First, due to high contribution of transportation to CO2 emissions, it is possible to reduce its negative impact on the environment (Ballot and Fontane, 2010). Collaboration in logistics can lead to significant decrease of distance travelled by trucks, which is actually also shown in terms of this study; the number of trucks needed for transportation of same goods amount can be also decreased. Secondly, collaborations in logistics can improve service levels, gain additional market shares, enhance capacities, and reduce the negative impacts of the bullwhip effect (Audy et al., 2012).

It is also necessary to say a couple of words about how collaboration in logistics can occur at all. Gansterer and Hartl (2018) identified three major streams of researches in the scientific literature. **Centralized collaborative planning -** collaborative decisions are made by a central authority such as online platform which can calculate optimal solutions for any logistic problem and also bring together all interested parties. Furthermore, the researchers pointed out **decentralized planning without auctions**. In such a decentralized setting collaborators might cooperate individually or supported by a central authority. Participants of the process have to decide which partners can take over some requests and which requests can be offered. And ultimately there is also auction-based decentralized planning. In such a mode, requests can be distributed through auctions. It means that all requests have to be previously submitted to a common pool and the partner that can perform it for the cheapest price get it.

Furthermore, Gansterer and Hartl (2018) identified problems which carriers must tackle in case of auction based decentralized planning. They have to make a decision which of their requests they want to preserve and which should be distributed over other carriers (*request evaluation*). During the auction process, they have to generate bids reflecting their willingness to buy requests from other carriers. Ultimately, the third issue is that a route must be created in a way that both own requests and taken over ones could be served.

A typical auction should consist of 5-phase procedure (Berger and Bierwirth 2010):

1. Carriers decide which of their requests are to be submitted in the common auction pool;
2. Auctioneer generates bundles of requests from different carriers and offers them to all carriers;
3. Participants (also carriers) compete with each other for the offered bundles giving a bid;
4. Auctioneer determine winners for each auction and allocate bundles to carriers based on their bids;
5. All profits are split between the involved carriers.

The widespread profit-sharing method is the Shapley value (Guajardo and Rönnqvist, 2016). However, for calculating Shapley values, carriers would have to reveal sensitive information regarding, for example, their complete set of customers.

Gansterer and Hartl (2018) defined also the theoretical model of Bundle Generation Problem (BuGP). The point is to make the set of offered bundles with which the total marginal profit can be maximized.

For evaluating of some sensitive information (which does not have to be revealed) the researchers offered proxies. The proxies were based on the intuitive assumption that good bundles must be dense, isolated from another bundles and include requests which are feasible within a short period of time. A genetic algorithm solves the problem finding the optimal solution with which attractive and feasible bundles can be generated. The results were that even without a revelation of sensitive information, efficient bundles could be found. The algorithm worked both for set with limited number of bundles and for set with complete bundle pool. For the former, however, a small loss in solution quality was observed.

Masson et al. (2017) conducted a similar study where exploiting of public transport capacities for transportation of goods from a consolidation distribution center (CDC) to the customers in the city center was considered. This was presumed to be done only in terms of B2B for shops and other retailers. The start of deliveries was at a CDC. From the bus stops in the city center parcels are supposed to be transported by city freighters. City freighters are vehicles of relatively small capacity that can travel along the narrow and crowded streets of the city center to perform the required distribution activities (for instance, electric cars or other zero emission vehicles). Obviously, their capacity is pretty small for providing transportation of large amounts of parcels from the CDC. Therefore, one came to idea of using buses since it is often the case that they are used with a pretty low utilization rate. According to the author, it should be done beyond peak hours, to transport a certain number of parcels to a bus stop in the city center wherefrom they can be distributed to final customers by city freighters. So, the main idea is "to *reduce the global impact of deliveries in the city, including various factors such as cost, CO2 emissions, traffic congestion, noise, infrastructure, inconvenience caused by parked vehicles, etc." (*Mason et al., 2017).

Li et al. (2014) and Li et al. (2016) provided one another paper related to the topic. The authors proposed to use taxis in parcels transport. There searchers introduced sets of passenger and parcel stops and destinations and quantity of passengers, fares for transporting passengers and parcels, extra travel distances occurred due to taking parcels alone, lower bounds for time windows, discount factor for exceeding the due time as variables as well. Furthermore, authors investigated a more realistic case when passengers are able to accept or decline inserting of parcel into the vehicle. The conclusion of the paper is that taxi-sharing is quite promising for urban areas, also for parcel transportation.

In the further paper of these researchers (2016, they investigate possibilities of time slack strategy and show that medium-sized real-life instances can be solved within a relatively short processing time as well.

Strongly linked to collaborative logistics is *crowd-shipping. Crowd-shipping is a delivery policy in which, in addition to standard vehicle routing practices, ordinary people accept to deviate from their route to deliver items to other people, for a small compensation* (Macrina et al., 2019). The advantages of crowd-shipping are defined by the authors in the following way. Crowd-shipping causes a cost saving: generally, the compensation for the occasional drivers is less than the standard drivers' remuneration. Secondly, no infrastructure is needed: crowd-shipping does not require any significant additional infrastructure. Thirdly, drivers in crowd-shipping are much more flexible, while the traditional deliveries are fixed and planned in advance, for crowd-shipping, fast delivery is the key factor. And ultimately it must be taken into consideration that Reduced environmental impacts: sharing vehicles can lead to a reduction in polluting emissions, energy consumption, noise and traffic congestion.

Archetti et al. (2016) provided a computational study to Vehicle routing problem with occasional drivers. In this study, the possibility of using of occasional drivers for last-mile deliveries was investigated. The researchers took into consideration such factors as ratio of the occasional drivers number to the number of customers, the level of the flexibility for each driver, remuneration of occasional drivers for the travelled distance. It was also presumed that there can be two compensation schemes: the first one had fixed rate per delivery and the second one contained also additional compensation for mileage. The author compared costs occurring for performing deliveries by regular drivers with costs which one could bear in case of using occasional drivers. The study was conducted with varying compensation rates and flexibility rates and the researchers achieved a result which presumed prevailing use of occasional drivers at lower costs than if regular drivers would have been used.

Many companies developed their own delivery policies based on a crowd-shipping option. For instance, in 2015 Amazon created an application where one can register and after that be able to provide delivery of parcels for Amazon. There are several delivery opportunities offered by

Amazon Flex which consider different time windows within which a package has to be delivered: three or more hours, one or two hours, less than one hour. An occasional driver is able to select from among two ways to pick up delivery objects: by choosing dates on a calendar, then the platform sends delivery offers for these dates, or by checking for available delivery objects on the Amazon Flex home screen.

Nevertheless, crowd logistics has also some drawbacks. One of that is possible return of purchased products (Sampaio, 2017). In the European Union, as well as in the most countries of the world, it is possible to return a purchase without explanation of reasons within some time after the purchase and the value must be completely refunded. According to estimations of Reagan (2016), in 2015, 30% of goods purchased on-line were returned and it is well known that the return rate for apparel purchased online is at least 20%. Expensive items can hit 50%, brick-and-mortar stores demonstrate only the 10%-return rate. Moreover, many customers are willing to purchase only if a free return option is guaranteed.

In the study of Chen et al. (2017), solution to this problem was to use shops (if they are able to provide flexible delivery and pickup times), to build collection hubs for returned purchases in which en-route taxi services are included, e.g. via collection packages at shops, delivery of packages to the shops. There are different collection strategies developed in which way to dispatch the taxis to transport goods from shops to the distribution centers. Such strategies exploit the extra capacity for small parcel transported by taxis.

In this chapter we provided definition of collaborative logistics and also studied forms how it takes place. It was also mentioned that some companies implemented collaborative logistics in the practice.

4. Problem setting

In the first part of the master thesis, it was necessary to propose solutions to *Multi-Depot Travel Salesman Problem with Pickup and Delivery* (MDTSPPD) and *Multi-Depot Vehicle Routing Problem with Pickup and Delivery* (MDVRPPD) with several carriers containing pickup and delivery vertices between which there exists a one-to-one relationship. The problem consists in determining the way to merge all carriers' tours in terms of above-mentioned *auction-based decentralized planning*. Initial data with node coordinates is provided by instance files mentioned below. The first part of the algorithm must read in the data and provide initial schedule for each carrier. After that the nodes can be relocated in a way that collaboration gains can be achieved. The relocation is carried out in the tours of all carriers. If a carrier can insert a node (both pickup and delivery) in a way that cause cost reduction, the insertion must be performed. There are different insertion heuristics which can be applied.

A necessary constraint is that each pickup request is visited before its corresponding delivery vertex. Furthermore, the mathematical formulation of both problems as well as the verbal explanation are provided. Mathematical formulation for the MDTSPPD is extracted from the study of Gansterer, Hartl, Salzmann (2017) and for the MDVRPPD from the studies of Gansterer, Hartl, Wieser (2019), Dragomir et al. (2018)

4.1 Mathematical formulation

MDTSPPD

Notation

n number of customers

m number of depots

P set of pickup vertices, $P=\{1,...,n\}$

D set of delivery vertices, *D={n+1,...,2n}*

W set of depot vertices, *W={2n+1,...,2n+m+1}*

N set of all vertices, *N=P ∪D ∪W{1,...,2n+m+1}*

A set of all arcs *ij*, *A=N* × *N*

c_{ij} transportation cost when traveling from *i* to *j*

x_{ij} decision variable indicating whether arc *ij* is used or not

b_{ij} decision variable indicating whether vertex *i* is visited before vertex *j*

Formulation

$$minimize \sum_{ij\epsilon A} c_{ij}\, x_{ij} \qquad (1)$$

subject to:

$$\sum_{i\epsilon N} x_{ij} = 1 \;\; \forall j\epsilon N \qquad (2)$$

$$\sum_{j\epsilon N} x_{ij} = 1 \;\; \forall i\epsilon N \qquad (3)$$

$$b_{ki} \le b_{kj} + (1 - x_{ij}) \;\forall \text{ ij } \epsilon\; A \setminus \{2n+m+1{,}2n+1\}, k\epsilon N \setminus \{i\} \qquad (4)$$

$$b_{kj} \le b_{ki} + (1 - x_{ij}) \;\forall \text{ ij } \epsilon\; A \setminus \{2n+m+1{,}2n+1\}, k\epsilon N \setminus \{i\} \qquad (5)$$

$$x_{ij} \le b_{ij} \;\; \forall \text{ ij } \epsilon\; A \qquad (6)$$

$$b_{ij} = 0 \;\forall i\epsilon N \qquad (5)$$

$$b_{n+i,i} = 0 \;\forall i\epsilon P \qquad (6)$$

$$b_{i,i+n} = 1 \;\forall i\epsilon P \qquad (7)$$

$$b_{ij} = b_{n+i,j} \;\forall i\epsilon P, j\epsilon W \qquad (8)$$

$$b_{i,2n+1} = 0 \;\forall i\epsilon N \qquad (9)$$

$$b_{ij} = 1 \;\forall i,j\epsilon W|\; i < j \qquad (10)$$

$$b_{ji} = 0 \;\forall i,j\epsilon W|\; i < j \qquad (11)$$

$$b_{i,2n+m+1} = 1 \;\forall i\epsilon N \setminus \{2n+m+1\} \qquad (12)$$

$$x_{ij} \in \{0,1\}\ \forall i, j \in N \tag{13}$$

$$b_{ij} \in \{0,1\}\ \forall i, j \in N \tag{14}$$

MDVRPPD

Notation

R set of customer requests, $R=\{1,...,n\}$

P set of pickup nodes, $P=\{1,...,n\}$

D set of delivery nodes, $D=\{n+1,...,2n\}$

L set of depots (i.e. carriers), $L=\{2n+1,...,2n+m\}$

N set of pickup and delivery nodes, $N=P \cup D$

N_l set of all pickup nodes initially assigned to depot l

V set of all nodes, $V=N \cup L$

K_l set of vehicles at depot l, $K_l=\{1,..., k_l\}$

K set of all vehicles, $\cup_{l \in L} K_l$

c_{ij} cost of traveling from node i to node j

Q load capacity of each vehicle

T maximum tour duration for each vehicle

$$x_{ijk} = \begin{cases} 1, \text{ if vehicle k travels directly from node } i \text{ to node } j \\ 0, \text{ otherwise} \end{cases}$$

S_{ij} = loading amount of vehicle k at node i

t_{ij} = fulfillment time at node i on vehicle k

Formulation

$$minimize \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} c_{ij}\, x_{ijk} \tag{1}$$

$$\sum_{j \epsilon V} \sum_{k \epsilon K} x_{ijk} = 1 \;\; \forall j \epsilon N \tag{2}$$

$$\sum_{i \epsilon V} x_{ihk} - \sum_{j \epsilon V} x_{hik} = 0 \;\; \forall k \epsilon K, h \epsilon N \tag{3}$$

$$\sum_{i \epsilon V} x_{ilk} = \sum_{j \epsilon N} x_{ljk} \leq 0 \;\; \forall l \epsilon L, k \epsilon\, K_l \tag{4}$$

$$\sum_{j \epsilon N} x_{ijk} = \sum_{j \epsilon N} x_{i+n,j.k} = 0 \;\; \forall i \epsilon R, k \epsilon K \tag{5}$$

$$t_{jk} \leq \; t_{ik} + c_{ij} \; + M(1 - x_{ijk}) \; \forall i \epsilon V, j \epsilon V, k \epsilon\, K \tag{6}$$

$$t_{jk} \leq \; t_{ik} + c_{ij} \; - M(1 - x_{ijk}) \; \forall i \epsilon V, j \epsilon V, k \epsilon\, K \tag{7}$$

$$t_{ik} \leq \; t_{i+n,k} \; \forall i \epsilon R, k \epsilon K \tag{8}$$

$$\sum_{i \epsilon S} \sum_{j \epsilon S} \; x_{ijk} \; \leq |S| - 1 \;\; \forall \; k \epsilon \; K \;, S \subseteq \; N, |S| \geq 2 \tag{13}$$

$$x_{ijk} \; \epsilon \; \{0,1\} \; \forall i \epsilon V, j \epsilon V, k \epsilon \; K \tag{14}$$

$$t_{ik} \; \epsilon \; \mathbb{Z} \; \forall \; i \epsilon V, k \epsilon \; K \tag{15}$$

$$S_{ik} \; \epsilon \; \mathbb{N} \; \forall \; i \epsilon V, k \epsilon \; K \tag{16}$$

4.2 Verbal description

MDTSPPD

The objective function (1) ist o minimize overall travel costs (distance travelled by all carriers Constraint (2) and (3) mean that each pickup and delivery node ist o be visited only once.. In constraints (4)–(6) we copy the values of the routing decision variables to the precedence decision variables (Lu and Dessouky 2004). Precedences among depots and customers are provided by (7)–(14), where constraints (7)–(10) mean that each pickup node is to ne visited before the appropriate delivery node, ant they must be included in the same tour. Constraint (11) ensures that no node is visited prior to the first depot. The sequence of depots is determined by constraints (12) and (13). Constraint (14) provides that the depot (2n+m+1) is the last node in the Hamiltonian tour.

MDVRPPD

The objective function (1) ist o minimize overall travel costs (distance travelled by all carriers and vehicles). Constraint (2) means that each pickup and delivery node ist o be visited only once. Constraint (3) ensures visiting and leaving a node by the same vehicle (3). Constraint (4) provides start and end of a tour at the same depot Additionally, pickup and delivery points have to be included in the same tour (5). Constraints (6) and (7) provide precedence constraint, which means that a pickup node is visited before the appropriate delivery node (8). Constraint(13) prevents the use of subtours. Ultimately, constraints (14) to (16) define the decision variables.

5.Solution methods

5.1 MDTSPPD

5.1.1 Construction heuristic

First of all, as mentioned above, it was necessary to apply a heuristic which provides an initial solution and respectively generates a schedule for all carriers. It is possible to apply sequential heuristic which assigns all requests sequentially according to their position in the data set in order to have any feasible solution. However, in order to obtain a good solution, in which carriers can compete with each other, it was decided to apply a heuristic that computes such a solution at the beginning, namely *Cheapest insertion.* The procedure is described by Golden et al. (1980) and includes:

- Step 1. Start with a subgraph consisting of node *i* only
- Step 2. Find node *k* such that c_{ik} is minimal and form subtour *i* - *k – i*
- Step 3. Find constellation that *(i, j)* are in subtour and *k* is not, such that $c_{ik} + c_{kj} - c_{ij}$ is minimal and, then, insert *k* between *i* and *j*.
- Step 4. Go to Step 3 unless we have a Hamiltonian cycle (vehicle starts at the depot and comes back there).

In our case, an essential feature of the start insertion in terms of step 2 is that we have pickup and delivery location. That’s why it was necessary to consider costs of insertion for both of them. Initial insertion works as follows:

Algorithm 1 The farthest node to other depots for MDTSPPD

Input: set of nodes with coordinates N, pickups P, deliveries D

```
1. for each request of the appropriate tour do
```

```
2.  determine insertion costs of pickup and delivery location (middle node between pickup
    and delivery must be considered)
3.      if current node is the farthest to one of depots of other carriers
4.      store position
5.          end if
6.          end for
7.  return the best start insertion
```

The operator applied in the study looks like this:

Algorithm 2 Cheapest insertion operator for MDTSPPD

Input: set of nodes with coordinates N, pickups P, deliveries D initial insertion (step 2)

```
1.  while there are unvisited requests do
2.      for each unvisited customer do
3.          for each position in the tour do
4.              determine insertion costs $c_{ik} + c_{kj} - c_{ij}$
5.              if current position is the cheapest and
6.              precedence constraint is not violated then
7.              store position
8.              end if
9.              end for
10.         if current request is the cheapest then
11.         store request
12.         end if
13.         end for
14.     insert the cheapest request at the cheapest position
```

```
end while
```

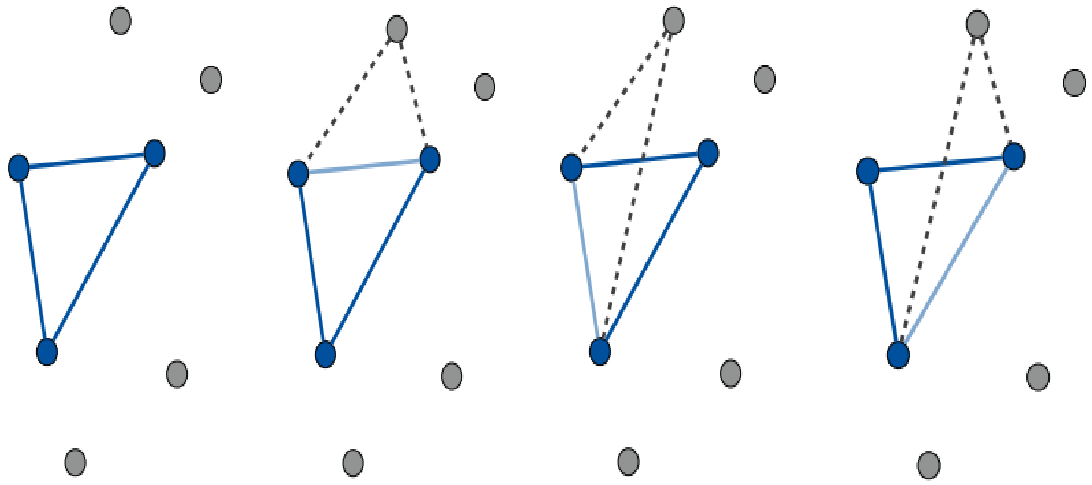


Figure 3. One way to perform cheapest insertion. Kiefer and Wolfinger (2018). Foliensatz zu IOT.

The main difference of the algorithm above to the one introduced by Golden et al. (1980) is that we have pickup and delivery nodes within one request and, therefore, it was necessary to apply an operator which prohibits insertion of a delivery node before the corresponding pickup. Calculation of the insertion costs must have been done differently within the operator because it may occur following situations for calculation. First it is calculation insertion costs before the depot at the end. Secondly, it can be necessary to calculate insertion costs after the depot at the beginning. And ultimately it is necessary to calculating insertion costs in the middle of a tour.

For an insertion at the beginning and the end of a tour, it’s necessary to consider only pickups and at the end - only deliveries are possible. For an insertion in the middle of a tour insertion costs can be calculated either for a delivery node or for a pickup node.

Moreover, in terms of possible neighbour nodes, following circumstances can occur. First, it is possible that we need to make an insertion between two pickup nodes. Secondly, it can be that an insertion is to be conducted between a pickup and delivery node, as well as the other way round between a delivery and a pickup node. And ultimately it is also possible that we make an insertion between two delivery nodes.

For each outcome mentioned above it was necessary to define separate calculation procedure using operators “if” and “else”. The construction heuristic was organized so that for each request in the data set the cheapest insertion was calculated. Subsequently, only the one with the lowest costs was inserted.

5.1.2 Insertions

After construction heuristic, an algorithm for tours merging was applied. In the algorithm, certain amount of iterations was conducted. In each iteration, first step was to select a node which a carrier can offer for other carriers to be inserted in their tours (auctions). It was done in a way that a node for insertion could be selected:

- Randomly;
- The node with the farthest distance to the depot;
- The node with the nearest distance to depots of other carriers.

The latter ones were done in the following way

Algorithm 3 Farthest node to other depots for MDTSPPD

```
Input: set of nodes with coordinates N, pickups P, deliveries D, tours
for each node of the tour, carrier of which conducts the auction do
        determine distance of the centroid (middle node between pickup and
        delivery must be considered) to the depot
    if current node is the farthest to the depot
    store position
    end if
    end for
return the farthest node to the depot
```

The coordinates of middle points between pickup and delivery locations are their centroids.

The algorithm for determining the nearest node to the depots works same but instead of searching the farthest distance, the lowest value must be found and preserved. And, finally, random selection is performed in a way that a random node from the tour is picked and offered for insertion to another carriers. From the results of the study, we can observe efficiency of application of each kind of node selection. We will see below that they can yield different results in terms of deviations from the initial tour.

After node selection it can be proceeded to the auction, i.e. to the check whether the appropriate request (**both pickup and delivery location**) could be inserted in the tours of another carriers in a way that yields a collaboration gain. As it was previously mentioned the additional distance to the existing tour must include costs of insertion for pickup **and** delivery location. The tour distance of the giving carrier is afterwards also to be calculated in order to evaluate reasonability of the appropriate insertion

---

Algorithm 4 Merging of tours for MDTSPPD

Input: set of nodes with coordinates N, pickups P, deliveries D, initial tours, node selected for the auction, carrier holding the auction (giving carrier), carriers inserting the node (taking carriers)

```
if size of the tour of the giving carrier is more than 0
    for all taking carriers
        if size of a taking carrier is more than 0
            calculate insertion costs for both pickup and delivery location of the request using above mentioned cheapest insertion operator with respect to precedence constraint
            calculate delta in the tour distance
        else
```

```
                Insert both pickup and delivery location after the depot
            end if
            if current insertion provides cheaper costs then
                store position
            end if
        end for
        if an improving move found, then
            insert the request in the appropriate tour
            remove the request from the initial tour
        end if
    else if the tour of the giving carrier is empty
    continue
    end if
return end of the iteration
```

This procedure can be repeated infinite number of times until the optimal solution is reached and no further improvement possible. Now we can see how the whole algorithm can work with all above mentioned procedures. All carriers operate alternately according to the sequence as giving and taking carriers.

Algorithm 5 Main function for MDTSPPD

```
for all instance files
    read in the data sets provided by the instance files
    create initial tours using above described construction heuristic
```

```
    while the computation time is less than 60 seconds
        perform merging tours selecting the farthest node to depots
        if all nodes are assigned to one carrier then
        break (the loop is used also for another procedures) end if
    end while
    while the computation time is less than 120 seconds
        perform merging tours selecting the nearest node to depots
    end while
    while the computation time is less than 210 seconds
        perform merging tours selecting a random node
    end while
end for
```

We can see that each data set is proceeded using all methods of node selection one by one. It could have been done in another way using only one method. We will see that results yielded by each procedure are different.

After applying all procedures, all data is deleted and the next instance file can be proceeded. All steps are automatized and are to be conducted at each iteration and for each instance file.

5.2 MDVRPPD

5.2.1 Construction heuristic

The main difference to MDTSPPD is that now we have not only three carriers but also restriction in terms of travelled distance. It means that as soon as the limit is reached the tour must be terminated and another vehicle must be used. Number of vehicles is constrained as well.

So, we have structures of carriers each of which in its turn has substructures of vehicles needed for satisfying all requests for the initial tour.

Another difference to the previous applied construction heuristic is that we don’t calculate the overall cheapest insertion for all requests but calculate cheapest insertion for each request (both pickup and delivery) sequentially according to their position in the data set sequence. This is because of two reasons. First, the number of instance files we deal with in terms of MDVRPPD is 6 times more than previously and the computation time for applying cheapest insertion operator at all requests increases as well. Secondly, if we apply the overall cheapest insertion, it could be so that a pickup location of a request is inserted and afterwards, insertion of the appropriate delivery location is not possible anymore because the limit of travelled distance is reached **before** the iteration when this location can be considered as the overall cheapest insertion.

That’s why all requests are inserted sequentially with simultaneous insertion of pickup and delivery location. If the case occurs that a pickup location is inserted and the appropriate delivery can’t be inserted, pickup is removed from the tour and both are put into the tour of a new vehicle if the constraint of vehicle numbers is not violated. It’s also necessary to say that the maximal travelled distances provided by the instance files were increased by 50%. The reason is that some of them were not enough in order to enable each insertion in the existing tours. Below is the overview of construction heuristic applied:

Algorithm 6 Construction heuristic for MDVRPPD

Input: set of not assigned nodes, pickups P, deliveries D

```
while (true)
    if size of the tour = 0 then
        provide initial insertion
    else
        Calculate the cheapest insertion for the current request in the sequence of not assigned nodes (at first for pickup location)
```

```
    insert pickup location
    the same procedure for the delivery location
    calculate new travelled distance
        if travelled distance restriction is exceeded then
        remove both inserted pickup and delivery from the tour
            if number of vehicles restriction is not exceeded then
            start a new tour with a new vehicle
                else shift the request to the container with to be assigned later   requests
                erase the request from initially not assigned nodes
                end if
                else
            erase the request from initially not assigned nodes
            end if
        if all nodes are assigned
        break
        end if
    end if
end while
if the container with to be assigned later nodes is not empty then
    for each not assigned node
        for each carrier
            for each vehicle
            insert both pickup and delivery location
            calculate the new travelled distance
                if travelled distance restriction is exceeded
                remove both inserted pickup and delivery from the   tour
                else
                remove the request from the set with to be inserted
```

```
34.                 end if
35.             end for
36.         end for
37.     end for
38. end if
```

It is necessary to say that all requests which could not be inserted initially are moved in the special created container. They all are checked for any possible insertion sequentially at the first suitable position (line 24-37 of the algorithm). Such insertions are only possible in the tours of another carriers.

One more essential difference to the construction heuristic of the MDTSPPD is that vehicle number constraint must be considered. Such a solution provides a good basis for applying insertions.

5.1.2 Insertions

Below there is the algorithm for merging of tours applied in the study.

Algorithm 7 Merging of tours for MDVRPPD

Input: set of nodes with coordinates N, pickups P, deliveries D, initial tours, node selected for the auction, carrier holding the auction (giving carrier), carriers inserting the node (taking carriers)

```
1. calculate delta of the giving tour in case if a request is removed
2. for each carrier
3.     for each vehicle
4.         for each node of a tour
```

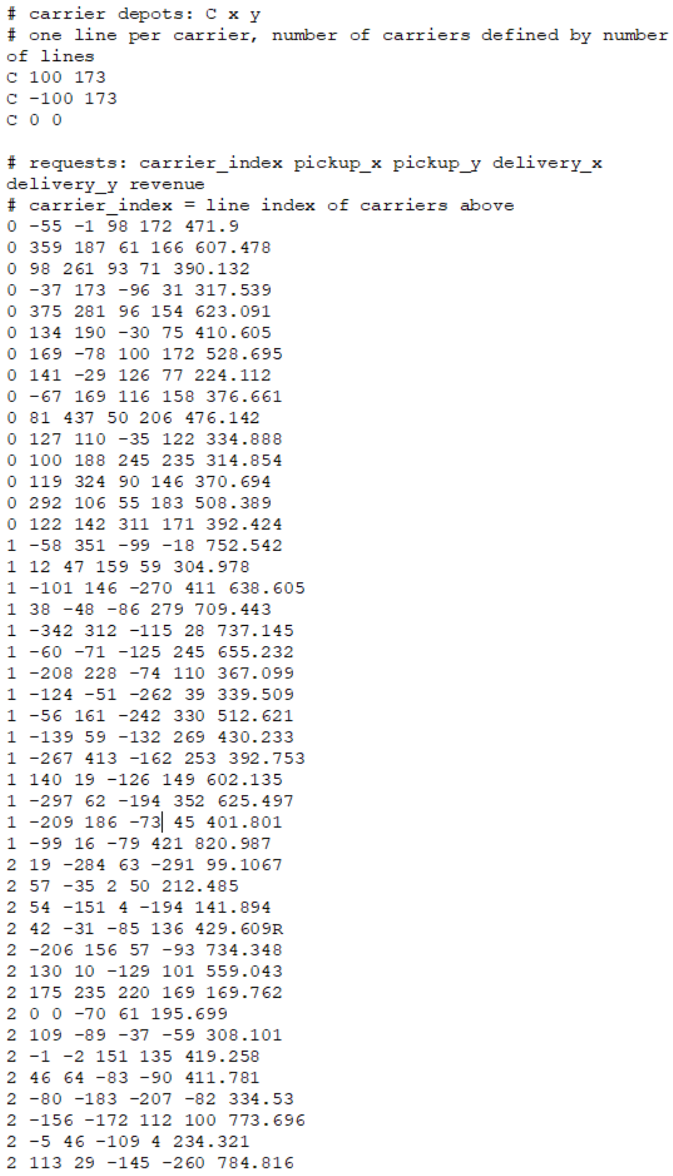
```
        if size of a taking carrier is more than 0
            calculate insertion costs for both pickup and delivery location of the request using
            above mentioned cheapest insertion operator with respect of precedence
            constraint
        calculate delta in the tour distance
        else
            Insert both pickup and delivery location after the depot
        end if
        if current insertion provides cheaper costs and travelled distance   constraint is not
        violated then
      store position
      end if
      end for
    end for
  end for
  if an improving move found, then
  insert the request in the appropriate tour
  remove the request from the initial tour
  end if
  else if the tour of the giving carrier is empty
    continue
    end if
  return end of the iteration
```

In this chapter algorithms used for the computational study were provided. These algorithms include heuristic for initial solution and heuristic for conducting auctions.

6. Instance files.

The algorithm was coded in C++ and the experiments were carried out single-threaded on an Intel Core i7-2640M processor with 2.80 GHz, operated under Windows 7.

Twenty (20) instance files were used for MDTSPPD in the study. All of them have been standardized, are uniform and contain information about 3 carriers, coordinates of their depots, coordinates of pickup and delivery locations, revenue for the requests, and carrier which these requests are assigned to. The number of requests is also standardized and is equal to 45. Therefore, each carrier has initially 15 instances.



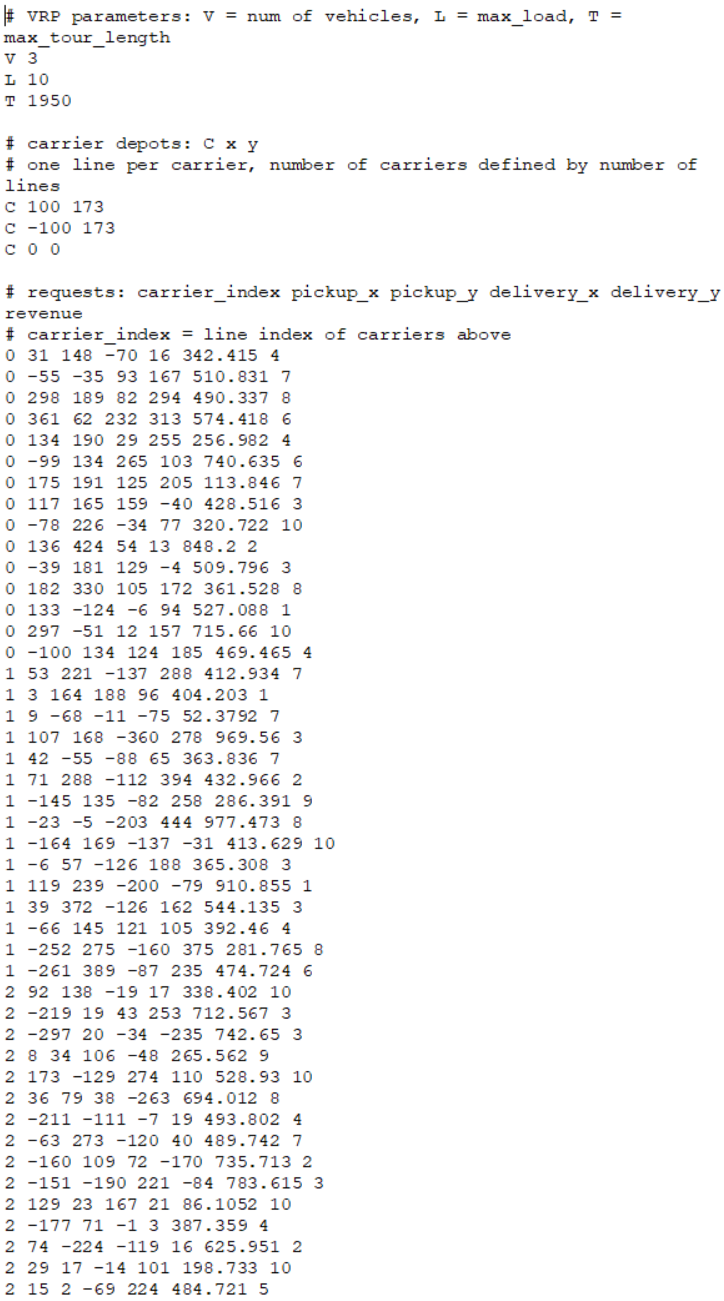
```
# carrier depots: C x y
# one line per carrier, number of carriers defined by number
of lines
C 100 173
C -100 173
C 0 0

# requests: carrier_index pickup_x pickup_y delivery_x
delivery_y revenue
# carrier_index = line index of carriers above
0 -55 -1 98 172 471.9
0 359 187 61 166 607.478
0 98 261 93 71 390.132
0 -37 173 -96 31 317.539
0 375 281 96 154 623.091
0 134 190 -30 75 410.605
0 169 -78 100 172 528.695
0 141 -29 126 77 224.112
0 -67 169 116 158 376.661
0 81 437 50 206 476.142
0 127 110 -35 122 334.888
0 100 188 245 235 314.854
0 119 324 90 146 370.694
0 292 106 55 183 508.389
0 122 142 311 171 392.424
1 -58 351 -99 -18 752.542
1 12 47 159 59 304.978
1 -101 146 -270 411 638.605
1 38 -48 -86 279 709.443
1 -342 312 -115 28 737.145
1 -60 -71 -125 245 655.232
1 -208 228 -74 110 367.099
1 -124 -51 -262 39 339.509
1 -56 161 -242 330 512.621
1 -139 59 -132 269 430.233
1 -267 413 -162 253 392.753
1 140 19 -126 149 602.135
1 -297 62 -194 352 625.497
1 -209 186 -73 45 401.801
1 -99 16 -79 421 820.987
2 19 -284 63 -291 99.1067
2 57 -35 2 50 212.485
2 54 -151 4 -194 141.894
2 42 -31 -85 136 429.609R
2 -206 156 57 -93 734.348
2 130 10 -129 101 559.043
2 175 235 220 169 169.762
2 0 0 -70 61 195.699
2 109 -89 -37 -59 308.101
2 -1 -2 151 135 419.258
2 46 64 -83 -90 411.781
2 -80 -183 -207 -82 334.53
2 -156 -172 112 100 773.696
2 -5 46 -109 4 234.321
2 113 29 -145 -260 784.816
```

Figure 4. An example of instance file for MDTSPPD

Figure 4 provides an example of instance file for MDTSPPD.

In comparison to the MDTSPPD instance files for MDVRPPD are not standardized. There is still constant number of carriers which is equal to three, but the number of requests varies and can be either 45 or 30 (15 or 10 per carrier). The instance files for MDVRPPD also contain information about depots’ and requests’ coordinates and in addition maximal tour length constraint and maximal load.



```
# VRP parameters: V = num of vehicles, L = max_load, T =
max_tour_length
V 3
L 10
T 1950

# carrier depots: C x y
# one line per carrier, number of carriers defined by number of
lines
C 100 173
C -100 173
C 0 0

# requests: carrier_index pickup_x pickup_y delivery_x delivery_y
revenue
# carrier_index = line index of carriers above
0 31 148 -70 16 342.415 4
0 -55 -35 93 167 510.831 7
0 298 189 82 294 490.337 8
0 361 62 232 313 574.418 6
0 134 190 29 255 256.982 4
0 -99 134 265 103 740.635 6
0 175 191 125 205 113.846 7
0 117 165 159 -40 428.516 3
0 -78 226 -34 77 320.722 10
0 136 424 54 13 848.2 2
0 -39 181 129 -4 509.796 3
0 182 330 105 172 361.528 8
0 133 -124 -6 94 527.088 1
0 297 -51 12 157 715.66 10
0 -100 134 124 185 469.465 4
1 53 221 -137 288 412.934 7
1 3 164 188 96 404.203 1
1 9 -68 -11 -75 52.3792 7
1 107 168 -360 278 969.56 3
1 42 -55 -88 65 363.836 7
1 71 288 -112 394 432.966 2
1 -145 135 -82 258 286.391 9
1 -23 -5 -203 444 977.473 8
1 -164 169 -137 -31 413.629 10
1 -6 57 -126 188 365.308 3
1 119 239 -200 -79 910.855 1
1 39 372 -126 162 544.135 3
1 -66 145 121 105 392.46 4
1 -252 275 -160 375 281.765 8
1 -261 389 -87 235 474.724 6
2 92 138 -19 17 338.402 10
2 -219 19 43 253 712.567 3
2 -297 20 -34 -235 742.65 3
2 8 34 106 -48 265.562 9
2 173 -129 274 110 528.93 10
2 36 79 38 -263 694.012 8
2 -211 -111 -7 19 493.802 4
2 -63 273 -120 40 489.742 7
2 -160 109 72 -170 735.713 2
2 -151 -190 221 -84 783.615 3
2 129 23 167 21 86.1052 10
2 -177 71 -1 3 387.359 4
2 74 -224 -119 16 625.951 2
2 29 17 -14 101 198.733 10
2 15 2 -69 224 484.721 5
```

Figure 5. An example of instance file for MDVRPPD

Figure 5 provides an example of instance files for MDVRPPD. Revenues for the requests were ignored in the study as well as the maximal load constraint for the MDVRPPD. It is also necessary to say that all maximal tour lengths were increased by 50%. The reason is that, initially, in some instances it was not possible to reallocate some requests because the existing tours were full and no further insertion was possible.

The names of the instance files for both algorithms were as follows:

run=0+dist=200+rad=200+n=15

The values after *"rad"* provide information about radius and will be discussed detailed in the next chapter.

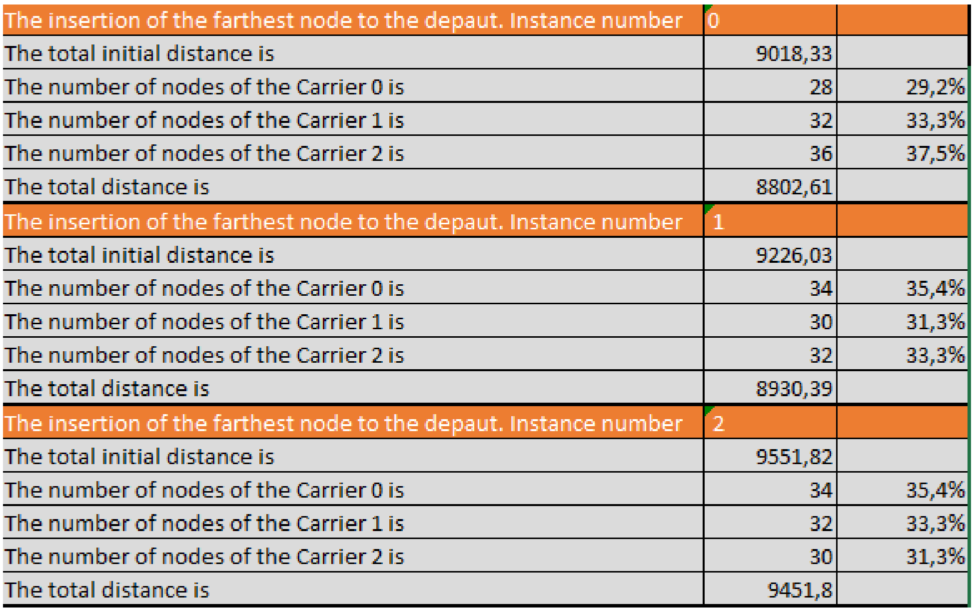
In this chapter the structure of the instance files used in the study were explained as well as some peculiarities of their names.

7. Results

7.1. MDTTSPPD

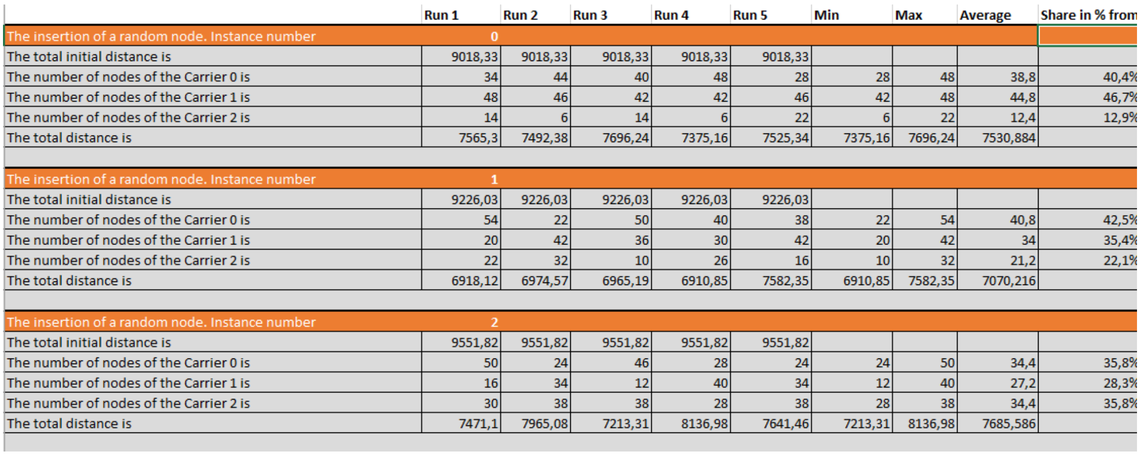
The output of the algorithms both for MDTSPPD includes:

- Results of initial assignment (construction heuristic);
- Results after insertion of the nodes farthest to depots of the giving carrier (ca. 20 sec.);
- Results after insertion of the nodes nearest to depots of the other carriers (ca. 20 sec.)
- Results after insertion of random nodes (ca. 30 sec.)



The insertion of the farthest node to the depaut. Instance number	0	
The total initial distance is	9018,33	
The number of nodes of the Carrier 0 is	28	29,2%
The number of nodes of the Carrier 1 is	32	33,3%
The number of nodes of the Carrier 2 is	36	37,5%
The total distance is	8802,61	
The insertion of the farthest node to the depaut. Instance number	1	
The total initial distance is	9226,03	
The number of nodes of the Carrier 0 is	34	35,4%
The number of nodes of the Carrier 1 is	30	31,3%
The number of nodes of the Carrier 2 is	32	33,3%
The total distance is	8930,39	
The insertion of the farthest node to the depaut. Instance number	2	
The total initial distance is	9551,82	
The number of nodes of the Carrier 0 is	34	35,4%
The number of nodes of the Carrier 1 is	32	33,3%
The number of nodes of the Carrier 2 is	30	31,3%
The total distance is	9451,8	

Table1. Example of output for the nearest and the farthest node insertion (MDTSPPD)



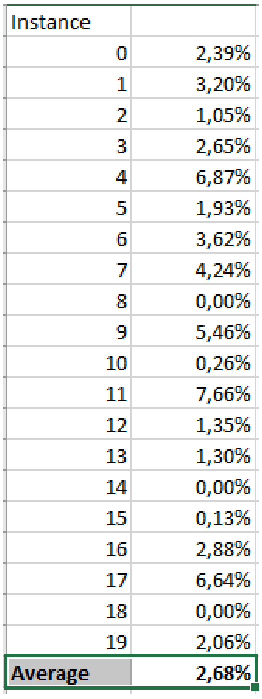
	Run 1	Run 2	Run 3	Run 4	Run 5	Min	Max	Average	Share in % from
The insertion of a random node. Instance number	0								
The total initial distance is	9018,33	9018,33	9018,33	9018,33	9018,33				
The number of nodes of the Carrier 0 is	34	44	40	48	28	28	48	38,8	40,4%
The number of nodes of the Carrier 1 is	48	46	42	42	46	42	48	44,8	46,7%
The number of nodes of the Carrier 2 is	14	6	14	6	22	6	22	12,4	12,9%
The total distance is	7565,3	7492,38	7696,24	7375,16	7525,34	7375,16	7696,24	7530,884	
The insertion of a random node. Instance number	1								
The total initial distance is	9226,03	9226,03	9226,03	9226,03	9226,03				
The number of nodes of the Carrier 0 is	54	22	50	40	38	22	54	40,8	42,5%
The number of nodes of the Carrier 1 is	20	42	36	30	42	20	42	34	35,4%
The number of nodes of the Carrier 2 is	22	32	10	26	16	10	32	21,2	22,1%
The total distance is	6918,12	6974,57	6965,19	6910,85	7582,35	6910,85	7582,35	7070,216	
The insertion of a random node. Instance number	2								
The total initial distance is	9551,82	9551,82	9551,82	9551,82	9551,82				
The number of nodes of the Carrier 0 is	50	24	46	28	24	24	50	34,4	35,8%
The number of nodes of the Carrier 1 is	16	34	12	40	34	12	40	27,2	28,3%
The number of nodes of the Carrier 2 is	30	38	38	28	38	28	38	34,4	35,8%
The total distance is	7471,1	7965,08	7213,31	8136,98	7641,46	7213,31	8136,98	7685,586	

Table 2. Example of output for the random node insertion (MDTSPPD)

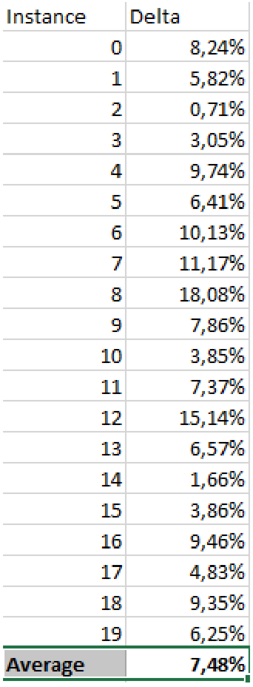
Within this master thesis, 20 instance files were tested. All algorithms for node insertions have been run independently from each other with initial construction heuristic and subsequent merging of tours according to the appropriate node insertion.

The farthest and nearest node selection algorithms were run only once because the results obtained would not change. The output is depicted in Table 2 which provides initial total distance after the construction heuristic, total distance after auctions, total number of requests in each carrier after auctions, share of requests per each carrier in total number of requests, delta of the total distance after auctions.

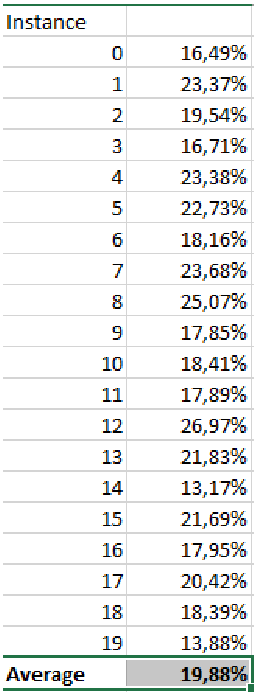
The algorithm with the random node selection was run 5 times and the results are summarized in the resulting table (above in Table 3, there is an example of output for some instances). In the resulting table, the information about the number of nodes which were assigned to a carrier after an algorithm is provided. After that, minimum, maximum and average values were calculated. According to average values of requests number and total distance, deltas and share of requests for each carrier were calculated.



Instance	
0	2,39%
1	3,20%
2	1,05%
3	2,65%
4	6,87%
5	1,93%
6	3,62%
7	4,24%
8	0,00%
9	5,46%
10	0,26%
11	7,66%
12	1,35%
13	1,30%
14	0,00%
15	0,13%
16	2,88%
17	6,64%
18	0,00%
19	2,06%
Average	**2,68%**



Instance	Delta
0	8,24%
1	5,82%
2	0,71%
3	3,05%
4	9,74%
5	6,41%
6	10,13%
7	11,17%
8	18,08%
9	7,86%
10	3,85%
11	7,37%
12	15,14%
13	6,57%
14	1,66%
15	3,86%
16	9,46%
17	4,83%
18	9,35%
19	6,25%
Average	**7,48%**



Instance	
0	16,49%
1	23,37%
2	19,54%
3	16,71%
4	23,38%
5	22,73%
6	18,16%
7	23,68%
8	25,07%
9	17,85%
10	18,41%
11	17,89%
12	26,97%
13	21,83%
14	13,17%
15	21,69%
16	17,95%
17	20,42%
18	18,39%
19	13,88%
Average	**19,88%**

Table 3. Deltas after each node selection algorithm (from left to right farthest, nearest and random nodes selection algorithm)

We have two values which are interesting for us, namely the travel distance and the number of nodes. The latter provides us the information about winner-takes-all effect in collaborative logistics – situation where a participant gains all or the majority of requests assigned initially to another participants. The objective was to test which node selection method can yield the most visible result. Interim results (deltas difference between initial travel distance and current in %) show that insertion according to farthest nodes from depots and nearest node from depots was not that efficient in comparison to random node selection.

We can see that farthest node selection can reduce the total travel distance only by 2,68% whereas random node selection yields a reduction of 19,88%.

>40%	>50%	>60%	>70%	>80%
0	0	0	0	0

>40%	>50%	>60%	>70%	>80%
8	0	0	0	0

>40%	>50%	>60%	>70%	>80%
17	2	0	0	0

Table 4. Summary of carriers' number which take certain number of requests after each node selection (top down farthest, nearest and random node selection) out of 20 instance files.

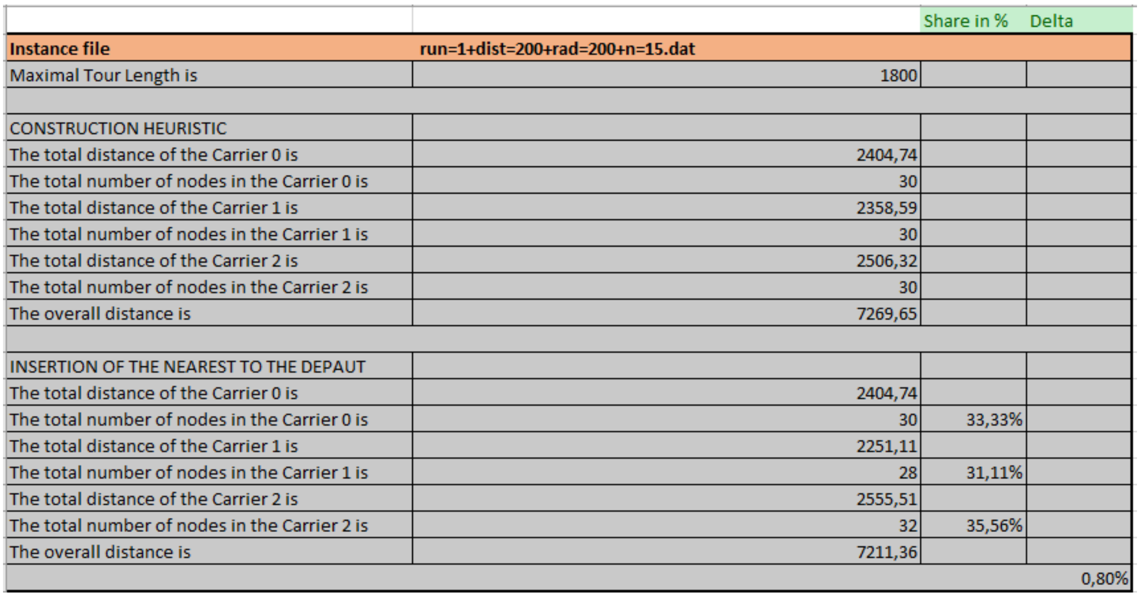
We can see that after applying the farthest node selection algorithm there were no carriers which took over even 40% of requests. Nearest node selection algorithm caused more changes and we see 8 cases when a carrier has more than 40% of requests. The proportion which is the most remoted from the initial assignment has resulted from the random node selection, 17 carriers have more than 40%, 2 out of them have more than 50%.

The reason why farthest and nearest node selection don’t yield significant changes in the tours is that there were no possibilities to insert requests at cheaper costs than initially. In the case of the farthest node selection, it could be so that a centroid is remoted from the depot but near to another request of the tour. The initial insertion was done accordingly to the cheapest insertion heuristic and it could have been pretty difficult to break up this connection. As soon as we have one request in a carrier which is the most remoted from the depot but could not be inserted at cheaper costs than previously, we get stuck in a local optimum. In the next iteration, where the carrier will be the giving carrier, the same request will be selected.

In the case of the nearest node selection, we deal with the nearest node to depots of another carriers. Same as with the farthest node selection, we can’t escape a local optimum once a selected node couldn’t be inserted at cheaper costs than initially. However, this algorithm yields better results than the farthest node selection.

7.2 MDVRPPD

In terms of the study, 120 instance files were tested. The output of the MDVRPPD algorithm with the farthest node selection and the nearest node selection applied is depicted in the table below.



		Share in %	Delta
Instance file	**run=1+dist=200+rad=200+n=15.dat**		
Maximal Tour Length is	1800		
CONSTRUCTION HEURISTIC			
The total distance of the Carrier 0 is	2404,74		
The total number of nodes in the Carrier 0 is	30		
The total distance of the Carrier 1 is	2358,59		
The total number of nodes in the Carrier 1 is	30		
The total distance of the Carrier 2 is	2506,32		
The total number of nodes in the Carrier 2 is	30		
The overall distance is	7269,65		
INSERTION OF THE NEAREST TO THE DEPAUT			
The total distance of the Carrier 0 is	2404,74		
The total number of nodes in the Carrier 0 is	30	33,33%	
The total distance of the Carrier 1 is	2251,11		
The total number of nodes in the Carrier 1 is	28	31,11%	
The total distance of the Carrier 2 is	2555,51		
The total number of nodes in the Carrier 2 is	32	35,56%	
The overall distance is	7211,36		
			0,80%

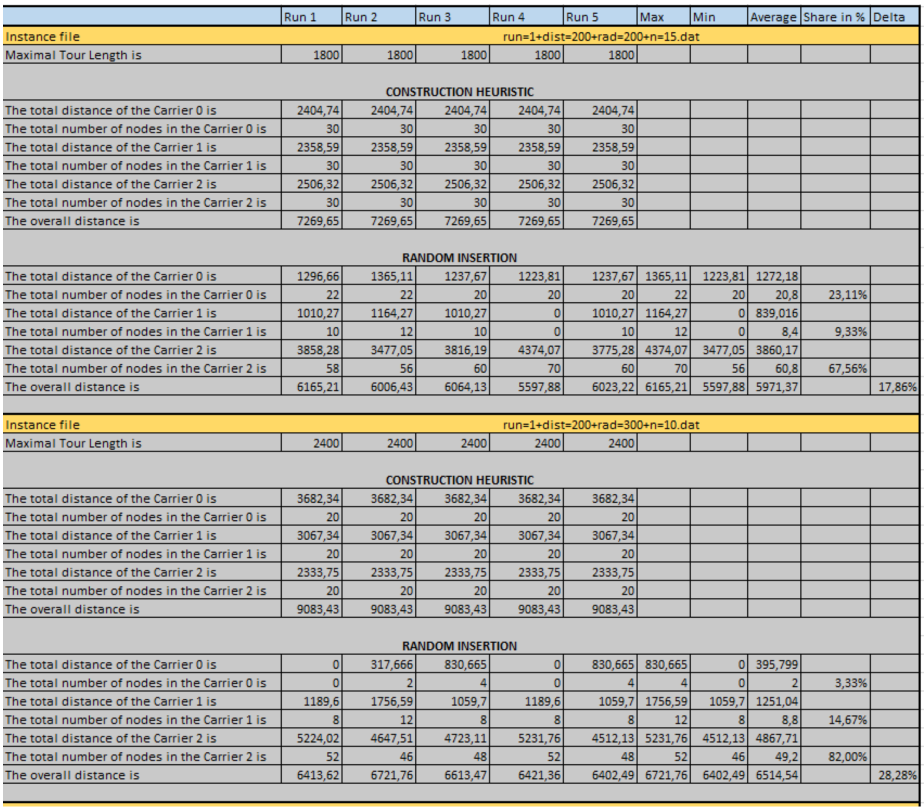
Table 5. Example of output for the nearest and the farthest node insertion (MDVRPPD)

As it was mentioned previously, maximal tour length constraint must have been considered in the algorithm. We can see appropriate values at the beginning of the appropriate entries (second row in the Table 6). Total distances per carrier show the sum of all distances travelled by vehicles assigned to this carrier (with several vehicles). In the construction heuristic, as soon as maximal tour length was exceeded, the request (both pickup and delivery location) must have been assigned to another vehicle. If maximal number of vehicles is exceeded as well, the request was put in the pool to be later assigned to following requests and it could be assigned to another carrier.

The node selections were conducted similarly to the MDTSPPD algorithm, The only difference is that instead of looking for the request farthest to the depot of the giving carrier or nearest to other depots, only in one tour (as in MDTSPPD, since we had only one vehicle) all tours assigned to the giving carrier were investigated in order to find out the node for insertion.

As it was mentioned previously, there was no possibility to apply Cheapest insertion heuristic in the form as it was in the MDTSPPD algorithm. Because of that reason, sequential insertion of all pickup and delivery location according to their position in the data set at cheapest position was conducted.

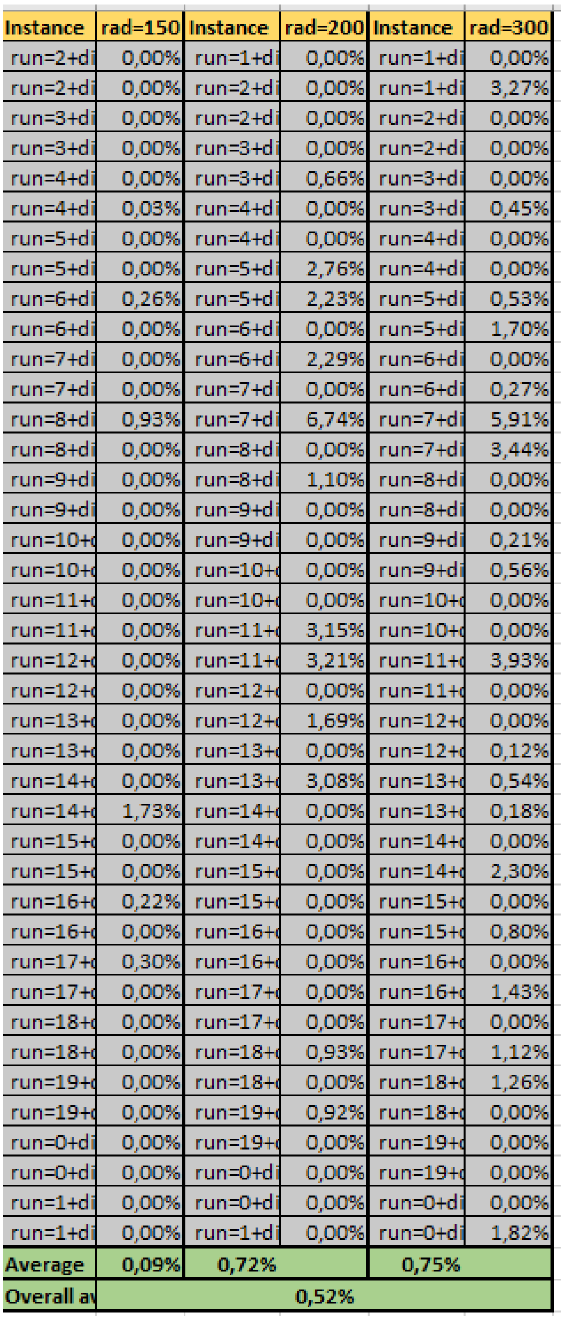
Total number of nodes shows us all nodes assigned to the **carrier** without detailed assignment per vehicle.



	Run 1	Run 2	Run 3	Run 4	Run 5	Max	Min	Average	Share in %	Delta
Instance file	run=1+dist=200+rad=200+n=15.dat									
Maximal Tour Length is	1800	1800	1800	1800	1800					
CONSTRUCTION HEURISTIC										
The total distance of the Carrier 0 is	2404,74	2404,74	2404,74	2404,74	2404,74					
The total number of nodes in the Carrier 0 is	30	30	30	30	30					
The total distance of the Carrier 1 is	2358,59	2358,59	2358,59	2358,59	2358,59					
The total number of nodes in the Carrier 1 is	30	30	30	30	30					
The total distance of the Carrier 2 is	2506,32	2506,32	2506,32	2506,32	2506,32					
The total number of nodes in the Carrier 2 is	30	30	30	30	30					
The overall distance is	7269,65	7269,65	7269,65	7269,65	7269,65					
RANDOM INSERTION										
The total distance of the Carrier 0 is	1296,66	1365,11	1237,67	1223,81	1237,67	1365,11	1223,81	1272,18		
The total number of nodes in the Carrier 0 is	22	22	20	20	20	22	20	20,8	23,11%	
The total distance of the Carrier 1 is	1010,27	1164,27	1010,27	0	1010,27	1164,27	0	839,016		
The total number of nodes in the Carrier 1 is	10	12	10	0	10	12	0	8,4	9,33%	
The total distance of the Carrier 2 is	3858,28	3477,05	3816,19	4374,07	3775,28	4374,07	3477,05	3860,17		
The total number of nodes in the Carrier 2 is	58	56	60	70	60	70	56	60,8	67,56%	
The overall distance is	6165,21	6006,43	6064,13	5597,88	6023,22	6165,21	5597,88	5971,37		17,86%
Instance file	run=1+dist=200+rad=300+n=10.dat									
Maximal Tour Length is	2400	2400	2400	2400	2400					
CONSTRUCTION HEURISTIC										
The total distance of the Carrier 0 is	3682,34	3682,34	3682,34	3682,34	3682,34					
The total number of nodes in the Carrier 0 is	20	20	20	20	20					
The total distance of the Carrier 1 is	3067,34	3067,34	3067,34	3067,34	3067,34					
The total number of nodes in the Carrier 1 is	20	20	20	20	20					
The total distance of the Carrier 2 is	2333,75	2333,75	2333,75	2333,75	2333,75					
The total number of nodes in the Carrier 2 is	20	20	20	20	20					
The overall distance is	9083,43	9083,43	9083,43	9083,43	9083,43					
RANDOM INSERTION										
The total distance of the Carrier 0 is	0	317,666	830,665	0	830,665	830,665	0	395,799		
The total number of nodes in the Carrier 0 is	0	2	4	0	4	4	0	2	3,33%	
The total distance of the Carrier 1 is	1189,6	1756,59	1059,7	1189,6	1059,7	1756,59	1059,7	1251,04		
The total number of nodes in the Carrier 1 is	8	12	8	8	8	12	8	8,8	14,67%	
The total distance of the Carrier 2 is	5224,02	4647,51	4723,11	5231,76	4512,13	5231,76	4512,13	4867,71		
The total number of nodes in the Carrier 2 is	52	46	48	52	48	52	46	49,2	82,00%	
The overall distance is	6413,62	6721,76	6613,47	6421,36	6402,49	6721,76	6402,49	6514,54		28,28%

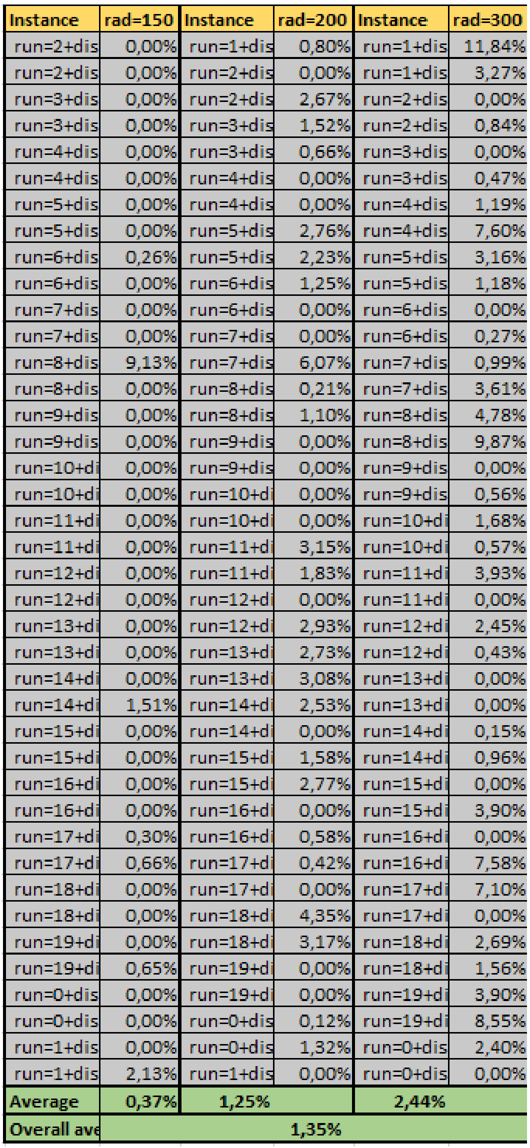
Table 6. Example of output for the random node insertion (MDVRPPD)

We can see in the table, as in the output for the MDTSPPD algorithm, initial number of nodes and travel distance per carrier as well as results of random node insertion of 5 runs conducted. As previously, minimum, maximum and average value for each parameter were calculated. Deltas and shares of each carrier have been extracted from the calculated average values of travel distance and number of nodes.



Instance	rad=150	Instance	rad=200	Instance	rad=300
run=2+di	0,00%	run=1+di	0,00%	run=1+di	0,00%
run=2+di	0,00%	run=2+di	0,00%	run=1+di	3,27%
run=3+di	0,00%	run=2+di	0,00%	run=2+di	0,00%
run=3+di	0,00%	run=3+di	0,00%	run=2+di	0,00%
run=4+di	0,00%	run=3+di	0,66%	run=3+di	0,00%
run=4+di	0,03%	run=4+di	0,00%	run=3+di	0,45%
run=5+di	0,00%	run=4+di	0,00%	run=4+di	0,00%
run=5+di	0,00%	run=5+di	2,76%	run=4+di	0,00%
run=6+di	0,26%	run=5+di	2,23%	run=5+di	0,53%
run=6+di	0,00%	run=6+di	0,00%	run=5+di	1,70%
run=7+di	0,00%	run=6+di	2,29%	run=6+di	0,00%
run=7+di	0,00%	run=7+di	0,00%	run=6+di	0,27%
run=8+di	0,93%	run=7+di	6,74%	run=7+di	5,91%
run=8+di	0,00%	run=8+di	0,00%	run=7+di	3,44%
run=9+di	0,00%	run=8+di	1,10%	run=8+di	0,00%
run=9+di	0,00%	run=9+di	0,00%	run=8+di	0,00%
run=10+	0,00%	run=9+di	0,00%	run=9+di	0,21%
run=10+	0,00%	run=10+	0,00%	run=9+di	0,56%
run=11+	0,00%	run=10+	0,00%	run=10+	0,00%
run=11+	0,00%	run=11+	3,15%	run=10+	0,00%
run=12+	0,00%	run=11+	3,21%	run=11+	3,93%
run=12+	0,00%	run=12+	0,00%	run=11+	0,00%
run=13+	0,00%	run=12+	1,69%	run=12+	0,00%
run=13+	0,00%	run=13+	0,00%	run=12+	0,12%
run=14+	0,00%	run=13+	3,08%	run=13+	0,54%
run=14+	1,73%	run=14+	0,00%	run=13+	0,18%
run=15+	0,00%	run=14+	0,00%	run=14+	0,00%
run=15+	0,00%	run=15+	0,00%	run=14+	2,30%
run=16+	0,22%	run=15+	0,00%	run=15+	0,00%
run=16+	0,00%	run=16+	0,00%	run=15+	0,80%
run=17+	0,30%	run=16+	0,00%	run=16+	0,00%
run=17+	0,00%	run=17+	0,00%	run=16+	1,43%
run=18+	0,00%	run=17+	0,00%	run=17+	0,00%
run=18+	0,00%	run=18+	0,93%	run=17+	1,12%
run=19+	0,00%	run=18+	0,00%	run=18+	1,26%
run=19+	0,00%	run=19+	0,92%	run=18+	0,00%
run=0+di	0,00%	run=19+	0,00%	run=19+	0,00%
run=0+di	0,00%	run=0+di	0,00%	run=19+	0,00%
run=1+di	0,00%	run=0+di	0,00%	run=0+di	0,00%
run=1+di	0,00%	run=1+di	0,00%	run=0+di	1,82%
Average	**0,09%**	**0,72%**		**0,75%**	
Overall a	**0,52%**				

Table 7. Deltas of travel distance after the farthest node insertion

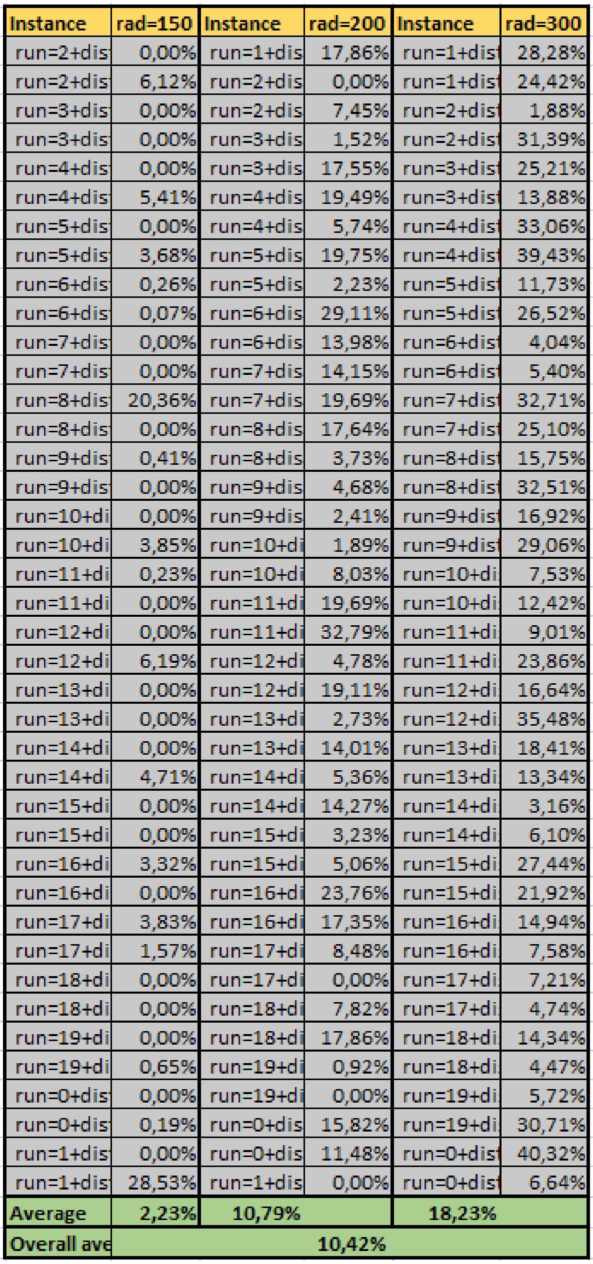


Instance	rad=150	Instance	rad=200	Instance	rad=300
run=2+dis	0,00%	run=1+dis	0,80%	run=1+dis	11,84%
run=2+dis	0,00%	run=2+dis	0,00%	run=1+dis	3,27%
run=3+dis	0,00%	run=2+dis	2,67%	run=2+dis	0,00%
run=3+dis	0,00%	run=3+dis	1,52%	run=2+dis	0,84%
run=4+dis	0,00%	run=3+dis	0,66%	run=3+dis	0,00%
run=4+dis	0,00%	run=4+dis	0,00%	run=3+dis	0,47%
run=5+dis	0,00%	run=4+dis	0,00%	run=4+dis	1,19%
run=5+dis	0,00%	run=5+dis	2,76%	run=4+dis	7,60%
run=6+dis	0,26%	run=5+dis	2,23%	run=5+dis	3,16%
run=6+dis	0,00%	run=6+dis	1,25%	run=5+dis	1,18%
run=7+dis	0,00%	run=6+dis	0,00%	run=6+dis	0,00%
run=7+dis	0,00%	run=7+dis	0,00%	run=6+dis	0,27%
run=8+dis	9,13%	run=7+dis	6,07%	run=7+dis	0,99%
run=8+dis	0,00%	run=8+dis	0,21%	run=7+dis	3,61%
run=9+dis	0,00%	run=8+dis	1,10%	run=8+dis	4,78%
run=9+dis	0,00%	run=9+dis	0,00%	run=8+dis	9,87%
run=10+di	0,00%	run=9+dis	0,00%	run=9+dis	0,00%
run=10+di	0,00%	run=10+di	0,00%	run=9+dis	0,56%
run=11+di	0,00%	run=10+di	0,00%	run=10+di	1,68%
run=11+di	0,00%	run=11+di	3,15%	run=10+di	0,57%
run=12+di	0,00%	run=11+di	1,83%	run=11+di	3,93%
run=12+di	0,00%	run=12+di	0,00%	run=11+di	0,00%
run=13+di	0,00%	run=12+di	2,93%	run=12+di	2,45%
run=13+di	0,00%	run=13+di	2,73%	run=12+di	0,43%
run=14+di	0,00%	run=13+di	3,08%	run=13+di	0,00%
run=14+di	1,51%	run=14+di	2,53%	run=13+di	0,00%
run=15+di	0,00%	run=14+di	0,00%	run=14+di	0,15%
run=15+di	0,00%	run=15+di	1,58%	run=14+di	0,96%
run=16+di	0,00%	run=15+di	2,77%	run=15+di	0,00%
run=16+di	0,00%	run=16+di	0,00%	run=15+di	3,90%
run=17+di	0,30%	run=16+di	0,58%	run=16+di	0,00%
run=17+di	0,66%	run=17+di	0,42%	run=16+di	7,58%
run=18+di	0,00%	run=17+di	0,00%	run=17+di	7,10%
run=18+di	0,00%	run=18+di	4,35%	run=17+di	0,00%
run=19+di	0,00%	run=18+di	3,17%	run=18+di	2,69%
run=19+di	0,65%	run=19+di	0,00%	run=18+di	1,56%
run=0+dis	0,00%	run=19+di	0,00%	run=19+di	3,90%
run=0+dis	0,00%	run=0+dis	0,12%	run=19+di	8,55%
run=1+dis	0,00%	run=0+dis	1,32%	run=0+dis	2,40%
run=1+dis	2,13%	run=1+dis	0,00%	run=0+dis	0,00%
Average	0,37%	1,25%		2,44%	
Overall ave		1,35%			

Table 8. Deltas of travel distance after the nearest node insertion

It is necessary to mention that, in contrast to the MDTSPPD algorithm, influence of **radius value** have been investigated. Radius means middle distance from depots to the nodes. The values are provided by the names of the instance files (numbers after “rad=”). It was determined and is shown in Tables 8, 9, 10 that there is a strong correlation between the radius values and results of the algorithms. We have three types of radius values (150, 200, 300). The more these values are, the greater deltas are obtained.

In Tables 8, 9, 10, all instances were split in separate columns according to the radius values and deltas for each instance are provided. The average values of deltas were calculated for each radius value and as a result of this the total average delta is extracted (shown in the lowest row of corresponding tables). We can see that the farthest node selection algorithm didn’t provide as significant changes in comparison to the initial heuristic. In the most of cases, there were no changes at all, the remaining ones yield the average delta of 0,52%. As we remember, the farthest node selection in terms of the MDTSPPD algorithm didn’t yield significant changes as well.



Instance	rad=150	Instance	rad=200	Instance	rad=300
run=2+dis	0,00%	run=1+dis	17,86%	run=1+dis	28,28%
run=2+dis	6,12%	run=2+dis	0,00%	run=1+dis	24,42%
run=3+dis	0,00%	run=2+dis	7,45%	run=2+dis	1,88%
run=3+dis	0,00%	run=3+dis	1,52%	run=2+dis	31,39%
run=4+dis	0,00%	run=3+dis	17,55%	run=3+dis	25,21%
run=4+dis	5,41%	run=4+dis	19,49%	run=3+dis	13,88%
run=5+dis	0,00%	run=4+dis	5,74%	run=4+dis	33,06%
run=5+dis	3,68%	run=5+dis	19,75%	run=4+dis	39,43%
run=6+dis	0,26%	run=5+dis	2,23%	run=5+dis	11,73%
run=6+dis	0,07%	run=6+dis	29,11%	run=5+dis	26,52%
run=7+dis	0,00%	run=6+dis	13,98%	run=6+dis	4,04%
run=7+dis	0,00%	run=7+dis	14,15%	run=6+dis	5,40%
run=8+dis	20,36%	run=7+dis	19,69%	run=7+dis	32,71%
run=8+dis	0,00%	run=8+dis	17,64%	run=7+dis	25,10%
run=9+dis	0,41%	run=8+dis	3,73%	run=8+dis	15,75%
run=9+dis	0,00%	run=9+dis	4,68%	run=8+dis	32,51%
run=10+di	0,00%	run=9+dis	2,41%	run=9+dis	16,92%
run=10+di	3,85%	run=10+di	1,89%	run=9+dis	29,06%
run=11+di	0,23%	run=10+di	8,03%	run=10+di	7,53%
run=11+di	0,00%	run=11+di	19,69%	run=10+di	12,42%
run=12+di	0,00%	run=11+di	32,79%	run=11+di	9,01%
run=12+di	6,19%	run=12+di	4,78%	run=11+di	23,86%
run=13+di	0,00%	run=12+di	19,11%	run=12+di	16,64%
run=13+di	0,00%	run=13+di	2,73%	run=12+di	35,48%
run=14+di	0,00%	run=13+di	14,01%	run=13+di	18,41%
run=14+di	4,71%	run=14+di	5,36%	run=13+di	13,34%
run=15+di	0,00%	run=14+di	14,27%	run=14+di	3,16%
run=15+di	0,00%	run=15+di	3,23%	run=14+di	6,10%
run=16+di	3,32%	run=15+di	5,06%	run=15+di	27,44%
run=16+di	0,00%	run=16+di	23,76%	run=15+di	21,92%
run=17+di	3,83%	run=16+di	17,35%	run=16+di	14,94%
run=17+di	1,57%	run=17+di	8,48%	run=16+di	7,58%
run=18+di	0,00%	run=17+di	0,00%	run=17+di	7,21%
run=18+di	0,00%	run=18+di	7,82%	run=17+di	4,74%
run=19+di	0,00%	run=18+di	17,86%	run=18+di	14,34%
run=19+di	0,65%	run=19+di	0,92%	run=18+di	4,47%
run=0+dis	0,00%	run=19+di	0,00%	run=19+di	5,72%
run=0+dis	0,19%	run=0+dis	15,82%	run=19+di	30,71%
run=1+dis	0,00%	run=0+dis	11,48%	run=0+dis	40,32%
run=1+dis	28,53%	run=1+dis	0,00%	run=0+dis	6,64%
Average	**2,23%**	**10,79%**		**18,23%**	
Overall ave	**10,42%**				

Table 9. Deltas of travel distance after the random node insertion

We can see that the average delta in absolute value is much lower than after the MDTSPPD algorithm (previously it was 2,68%). In Table 9, the summary of deltas after the nearest node selection algorithm is provided according to the same scheme as previously. We can see that the trend of correlation depending on radius values exists here as well, the more the radius value are the greater deltas become. Same as in the MDTSPPD algorithm, the nearest node selection yields a greater delta than the farthest node selection.

Also, it’s possible to conclude that the overall average delta of 1,35% (provided in the lowest row of Table 9) is in absolute value smaller than in the MDTSPPD algorithm, same as in the case of the farthest node selection.

Similarly, the summary of deltas after the random node selection algorithm is provided in Table 10. Same as in the previous two algorithms, the trend of correlation dependent on radius values can be also observed. As it was the case in the MDTSPPD algorithm, the greatest delta is obtained after the random node selection algorithm.

However, in absolute value, the average delta of 10,42% is two times lower than in the first algorithm.

	>40%	**>50%**	**>60%**	**>70%**	**>80%**	**>90%**
rad = 150	0	0	0	0	0	0
rad = 200	3	0	0	0	0	0
rad= 300	1	0	0	0	0	0
sum	4	0	0	0	0	0

Table 10. Summary of carriers’ number which take certain number of requests after the farthest node selection out of 120 instance files.

	>40%	**>50%**	**>60%**	**>70%**	**>80%**	**>90%**
rad = 150	1	0	0	0	0	0
rad = 200	5	0	0	0	0	0
rad= 300	8	0	0	0	0	0
sum	14	0	0	0	0	0

Table 11. Summary of carriers’ number which take certain number of requests after the nearest node selection out of 120 instance files.

	>40%	>50%	>60%	>70%	>80%	>90%	100%
rad = 150	11	4	2	2	1	1	0
rad = 200	32	22	16	7	3	1	0
rad= 300	37	31	25	19	13	2	0
sum	80	57	43	28	17	4	0

Table 12. Summary of carriers' number which take certain number of requests after the random node selection out of 120 instance files.

In Table 11, there is the summary of winner-takes-all effect for each node selection algorithm. We can make following conclusions. First it is necessary to say that the most significant changes, similarly to the MDTSPPD algorithm, are caused by the random node selection algorithm, the changes after the farthest node selection algorithm are minimal. Furthermore, even though the number of carriers taken more than 40% of requests is high in absolute values, in percentage to the number of instances the MDTSPPD algorithm caused more significant changes at the nearest and random node selection (in the MDVRPPD nearest node selection 14 out of 120 and in the MDTSPPD 8 out of 20, random node selection yielded in the MDTSPPD 17 out of 20 carriers and MDVRPPD 80 out of 120) And ultimately, if we consider really winner-takes-all effect, it is observable only at the random node selection in the MDVRPPD algorithm where there are a lot of cases when a carrier took over more than 70% of all requests.

It is necessary to say that, for the most part, differences in absolute values are caused by the fact that in the MDTSPPD algorithm only instance files with radius values 300 were used, whereas in the MDVRPPD algorithm the values 15 and 200 were applied for simulations as well.

In this chapter results of the computational study were provided and discussed. We saw that radius values and methods of nodes selection caused a big impact on the results. The results provided are first deltas in the distance travelled and secondly number of requests performed by each carrier after the reassignment.

8. Conclusion

This thesis contributes to collaborative logistics and its practical implementation in order to increase welfare of many stakeholders.

The objective of the thesis was to provide an overview on the state of the art in the scientific literature related to the sharing economy and collaborative logistics particularly. Additionally, we investigated the winner-takes-all effect in collaborative logistics in terms of decentralized planning with auctions. This effect describes a situation where several participants join their logistic capacities, but only one (or very few) of them takes over all requests from all participants.

In a numerical study, we assessed different kinds of node selection strategies for auctions. Also, we evaluated whether different radius values (middle distances from depots to the nodes) influences solution quality.

The thesis provided a computational study which simulates different processes in decentralized auction-based planning. It revealed that these processes generally perform well.

The study revealed that, depending on the heuristic applied and average distances between depots and nodes (radius), significant savings up to 20% can be achieved. Nevertheless, on average, there was no situation when a carrier took over all requests.

These findings are in line with a study by Wieser (2018), where the authors investigated collaborative pickup and delivery problem with centralized collaboration framework. In the MDTSPPD algorithm, the savings reached 40%, depending on the radius value. Similar to the thesis at hand the gains for the MDVRPPD algorithm were slightly lower.

The results could be interpreted in a way that companies which are in competition with each other can achieve a positive effect if they pool their logistic capacities. Not only these companies will be winners of such synergy but also other stakeholders such as state and citizens.

Historically, companies were reluctant to enter collaborations because sensitive information about, e.g., customers could be revealed. In France, however, Henkel pooled its logistic capacities with Colgate, GlaxoSmithKline and Sara Lee. This collaboration of independent companies and a logistics provider required a cultural shift within the firms, as well as certain commitment to exchange some information. In any case for implementing these processes in practice, it is necessary to build trust among the interested parties because, as it was already mentioned above, it requires revelation of sensitive information.

Gansterer and Hartl (2018) defined following directions for future studies in this sphere. First, it is the application of collaborative frameworks to more complex transportation systems. Secondly, it the investigation of strategic behaviour in terms of effective profit-sharing mechanisms. Thirdly, a comparison of auction-based compared to non-auction-based systems. And finally, estimations of the value of information in decentralized exchange mechanisms was emphasized. Clearly, collaborative logistics has a large potential for further theoretical and practical applications.

Zusammenfassung

Die vorliegende Masterarbeit ist ein Beitrag zur kollaborativen Logistik. Kollaborative Logistik ist ein wichtiger Teil von Transport in der Sharing Economy und nimmt heutzutage immer mehr an Bedeutung zu.

Die Masterarbeit beinhaltet Information über aktuelle wissenschaftliche Literatur, die sich auf das Thema bezieht. Weiterhin geht die Masterarbeit auf die Definition von unterschiedlichen Formen der kollaborativen Logistik ein und zwar zentralisierte kollaborative Planung, dezentralisierte kollaborative Planung mit Auktionen und dezentralisierte kollaborative Planung ohne Auktionen. Das Ziel der Masterarbeit war mithilfe von unterschiedlichen Eingangsdaten über Lokationen und Depots (im Sinne von ihren geografischen Koordinaten), Anzahl der Fahrzeuge, maximal erlaubten Weglängen und Methoden zur Kundenselektion Auktionen zu evaluieren.

Die Ergebnisse von allen Methoden zur Kundenselektion waren evaluiert, verglichen und diskutiert. Die wichtigste Feststellung ist, dass Gewinne im Falle der Kollaboration 20% erreichen können (im Sinne von der gesamten Distanz aller Fahrzeuge und Spediteure). Die Ergebnisse variieren jedoch abhängig von Methoden zur Kundenselektion und Höhe von Konkurrenz.

Literature

1. Archetti C., Savelsbergh M., Speranza M., 2016. The vehicle routing problem with occasional drivers. European Journal of Operational Research 254(2), 472–480 (2016)
2. Ashworth S., 2018. The Gig economy: How Uber impacted the London taxi industry. Journal of undergraduate research at NTU, Volume 1, Issue 1, pp.89-120
3. Audy J.-F., Lehoux N., D'Amours S., Rönnqvist M., 2012. A framework for an efficient implementation of logistics collaborations. InternationalTransactions in Operational Research 19 (5), 633–657.
4. Ballot E., Fontane F., 2010. Reducing transportation co2 emissions throughpooling of supply networks: perspectives from a case study in french retailchains. Production Planning & Control 21 (6), 640–650
5. Beard R., Macher J., Mayo J., 2015. Can You Hear Me Now?' Exit, Voice and Loyalty under Increasing Competition, February 2015.
6. Berger S, Bierwirth C (2010) Solutions to the request reassignment problem in collaborative carrier networks. Transp Res Part E Logist Transp Rev 46:627–638
7. Cavallini G., 2017. The (Unbearable?) Lightness of Self-Employed Work Intermediation: The Cases of Uber, Foodora and Amazon Mechanical Turk in the Light of the Italian Labour Law. in G. Zilio Grandi, M. Biasi (a cura di), Commentario breve allo statuto del lavoro autonomo e del lavoro agile, Wolters-Kluwer/Cedam, Milano, 285
8. Coopman C., Mitchell M., Thierer A., 2015. The Sharing Economy and Consumer Protection Regulation: The Case for Policy Change. The Journal of Business, Entrepreneurship & the Law. The Journal of Business, Entrepreneurship & the Law (Volume 8, Issue 2, Article 4)
9. Dyal-Chand R., 2015. Regulating Sharing: The Sharing Economy as an Alternative Capatilist System**.** Tul. L. Rev., 2015 - HeinOnline
10. Frenken C., Meelen T., Arets M., Van de Glind P. Smarter Regulation for the Sharing Economy, The Guardian May 20 2015, https://www.theguardian.com/science/political-science/2015/may/20/smarter-regulation-for-the-sharing-economy, Retrieved October 23, 2016.

11. Gansterer M.,Hartl R, Salzmann F., 2017. Exact solutions for the collaborative pickup and delivery problem. Cent.Eur.J.Oper.Res.26 (2018), 357–371.
12. Gansterer M.,Hartl R, Wieser S., 2019.Assignment constraints in shared transportation services
13. Gansterer M.,Hartl R., 2018. Centralized Bundle Generation in Auction-based Collaborative Transportation. OR Spectrum 40 (3): 613–635
14. Gansterer M.,Hartl R., 2018. Collaborative vehicle routing: a survey. European Journal of Operational Research, 2018 - Elsevier
15. Golden B, Levy L, Vohra R., 1980. The orienteering problem. Naval Research Logistics 1987; 34; 307-318.
16. Guajardo M., Rönnqvist M., 2016. A review on cost allocation methods in collaborative transportation. Int Trans Oper Res 23(3):371–392.
17. Dragomir A., Nicola D, Soriano A, Gansterer M.,2018. Multidepot pickup and delivery problems in multiple regions: a typology and integrated model. International Transactions in Operational Research 25(2):569–597
18. Heilig M., Mallig N., Schröder O., Kagerbauer M., Vortisch P., 2018. Implementation of free-floating and station-based carsharing in an agent-based travel demand model. Travel Behav. Soc.2018,12, 151–158
19. Horni, A., Nagel, K. and Axhausen, K.W. (eds.) 2016 *The Multi-Agent Transport Simulation MATSim*. London: Ubiquity Press. DOI: http://dx.doi.org/10.5334/baw. License: CC-BY 4.0
20. Kiefer A., Wolfinger D., 2018 Foliensatz zu IOT. The University of Vienna.
21. Machado C., De Salles H., Berssaneti F, Quintanilha j., 2018. An Overview of Shared Mobility. Sustainability 2018, 10, 4342
22. Martinez L.,Viegas J., 2017. Assessing the impacts of deploying a shared self-driving urban mobility system: An agent-based model applied to the city of Lisbon, Portugal. Int. J. Transp. Sci. Technol.2017, 6, 13–27
23. Matofska, B. 2014. What is the Sharing Economy? Available: http://www.thepeoplewhoshare.com/blog/what-is -the-sharing-economy/ [Accessed 22nd April 2015].

24. Rogers B., 2015. The Social costs of Uber. University of Chicago Law Review Online: Vol. 82 : Iss. 1 , Article 6.
25. Shaheen S., Stocker A., and Mundler M., 2017. Online and App-Based Carpooling in France: Analyzing Users and Practices. Transportation on Cities, pp. 181–196, Springer, 2017.
26. Täuscher K., Kietzmann. J., 2017. Learning from Failures in the Sharing Economy. MIS Quarterly Executive, 16, 4: 253-264.
27. Wallsten, S., 2015.The competitive effects of the sharing economy: how is Uber changing taxis? available at: www.researchgate.net/publication/279514652_The_Competitive_Effects_of_the_Sharing_Economy_How_is_Uber_Changing_Taxis (accessed 4 September 2015);
28. Wieser S., 2018. Fairness in Collaborative Pickup and Delivery Problems. Master's thesis, University of Vienna.

List of figures

List of tables