# The impact of decentral dispatching strategies on the performance of intralogistics transport systems

Von der Fakultät Maschinenwesen
der
Technischen Universität Dresden
zur
Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommene Dissertation

Dipl.-Wirt.-Ing. Nils Klein
geb. am 08.08.1981 in Marburg/Lahn

| | |
|---|---|
| Tag der Einreichung: | 20.12.2012 |
| Tag der Verteidigung: | 14.08.2013 |
| Gutachter: | Prof. Dr.-Ing. habil. T. Schmidt<br>Prof. Dr.-Ing. B. Schlecht<br>PD Dr.-Ing. G. Weigert |
| Vorsitzender der Promotionskommission: | Prof. Dr.-Ing. habil. U. Füssel |

# Abstract

This thesis focuses on control strategies for intralogistics transport systems. It evaluates how switching from central to decentral dispatching approaches influences the performance of these systems. Many ideas and prototypes for implementing decentral control have been suggested by the scientific community. But usually only the qualitative advantages of this new paradigm are stated. The impact on the performance is not quantified and analyzed. Additionally, decentral control is often confused with distributed algorithms or uses the aggregation of local to global information. In the case of the latter, the technological limitations due to the communication overhead are not considered. The decentral prototypes usually only focus on routing.

This paper takes a step back and provides a generic simulation environment which can be used by other researchers to test and compare control strategies in the future. The test environment is used for developing four truly decentral dispatching strategies which work only based on local information. These strategies are compared to a central approach for controlling transportation systems. Input data from two real-world applications is used for a series of simulation experiments with three different layout complexities. Based on the simulation studies neither the central nor the decentral dispatching strategies show a universally superior performance. The results depend on the combination of input data set and layout scenario. The expected efficiency loss for the decentral approaches can be confirmed for stable input patterns. Regardless of the layout complexity the decentral strategies always need more vehicles to reach the performance level of the central control rule when these input characteristics are present. In the case of varying input data and high throughput the decentral strategies outperform the central approach in simple layouts. They require fewer vehicles and less vehicle movement to achieve the central performance. Layout simplicity makes the central dispatching strategy prone to undesired effects. The simple-minded decentral decision rules can achieve a better performance in this kind of environment. But only complex layouts are a relevant benchmark scenario for transferring decentral ideas to real-world applications. In such a scenario the decentral performance deteriorates while the layout-dependent influences on the central strategy become less relevant. This is true for both analyzed input data sets. Consequently, the decentral strategies require at least 36% to 53% more vehicles and 20% to 42% more vehicle movement to achieve the lowest central performance level. Therefore their usage can currently not be justified based on investment and operating costs. The characteristics of decentral systems limit their own performance. The restriction to local information leads to poor dispatching decisions which in return induce self-enforcing inefficiencies. In addition, the application of decentral strategies requires bigger storage location capacity. In several disturbance scenarios the decentral strategies perform fairly well and show their ability to adapt to changed environmental conditions. However, their performance after the disturbance remains in some cases unpredictable and relates to the properties of self-organizing complex systems. A real-world applicability has to be called into question.

# Table of contents

# List of figures

# List of tables

# List of formulas

# List of symbols

| | |
|---|---|
| $a$ | Constant in autoregressive model |
| $A$ | Constant for Tukey window |
| $b$ | Radius for changing dispatching destination in feedback-based dispatching |
| $c$ | Feedback value |
| $c_{max}$ | Maximum feedback value |
| $C_j$ | Covariance for lag $j$ |
| $d_{vi}$ | Distance from current position of vehicle $v$ to location $i$ |
| $df$ | Degrees of freedom |
| $e_{ij}$ | Estimated demand of source $j$ at sink $i$ |
| $\boldsymbol{E}_i$ | List with estimated demands at sink $i$ |
| $f_{ij}$ | Remaining demand of source $j$ at sink $i$ |
| $\boldsymbol{F}_i$ | List with remaining demands at sink $i$ |
| $g_{ij}$ | Vehicles sent to source $j$ from sink $i$ |
| $\boldsymbol{G}_i$ | List of vehicles sent to source $j$ from sink $i$ |
| $h$ | Random variable from white noise process |
| $i$ | Subscript 1 |
| $j$ | Subscript 2 |
| $k$ | Number of lags |
| $k_{max}$ | Maximum number of lags according to SCHWERT'S rule of thumb |
| $K$ | Number of simulations |
| $l_p$ | Path length |
| $l_v$ | Vehicle length |
| $L$ | Number of simulation intervals |
| $m$ | Number of sources / sinks / storage locations in the system |
| $M$ | Number of observations |
| $n$ | Number of basic single-loop layouts in layout scenarios |
| $N$ | Index number for forecast intervals |
| $o$ | Number of switches / merges in the system |
| $p_{sim}$ | Probability of simultaneous confidence level inclusion |
| $p_0$ | Reset probability for feedback-based dispatching |
| $p_{id}$ | Probability for choosing current dispatching destination of vehicle |

| | |
|---|---|
| $p_{ix}$ | Probability for all dispatching locations except current vehicle destination |
| $p_{ij}$ | Probability at switch $i$ for choosing dispatching location $j$ |
| $\boldsymbol{P}_i$ | Dispatching table with all probabilities at switch $i$ |
| $q_{ij}$ | Known dispatching location $j$ at sink / switch $i$ |
| $\boldsymbol{Q}_i$ | List with all known dispatching locations at sink / switch $i$ |
| $r_{ij}$ | Received loads at sink $i$ |
| $\boldsymbol{R}_i$ | List with all received loads at sink $i$ |
| $s_{ij}^{est}$ | Estimated fill level of storage location $j$ at sink $i$ |
| $\boldsymbol{S}_i^{est}$ | List with all estimated fill levels at sink $i$ |
| $s_i^{net}$ | Net stock level at location $i$ |
| $s_i^{max}$ | Maximum storage location capacity of location $i$ |
| $s_i^{rel}$ | Relative fill level of storage location $i$ |
| $s_i^{thr}$ | Threshold value for evaluation of net stock level at location $i$ |
| $s_j^v$ | Fill level information about storage location $j$ provided by vehicle $v$ |
| $t$ | Time |
| $t_{dist}$ | Time of disturbance |
| $t_{norm}$ | Time when path failure is fixed and system returns to standard operation |
| $t^N$ | Length of each forecast period |
| $t_j^{StWait}$ | Standard waiting time at source or storage location $j$ |
| $t_v^{Wait}$ | Waiting time vehicle $v$ perceived at its current departure location |
| $t_j^{AvWait}$ | Average waiting time at location $j$ |
| $t_{thr}$ | Threshold value for feedback calculation at sources |
| $t_{sou}^{rel}$ | Threshold value for vehicle release at sources |
| $t_{sto}^{rel}$ | Threshold value for vehicle release at storage locations |
| $u_p$ | Utilization of path $p$ |
| $u_{thr}$ | Threshold value for path utilization |
| $v$ | Vehicle index number |
| $v_i^{cur}$ | Number of vehicles currently at location $i$ |
| $v_i^{app}$ | Number of vehicles approaching location $i$ |
| $v_p^{cur}$ | Number of vehicles currently on path $p$ |
| $w_i$ | Number of waiting loads at location $i$ |
| $W_k$ | Weighting function |
| $x$ | Time series value |
| $y$ | Random variable / observation |
| $z_i$ | Priority value for location $i$ |

| | |
|---|---|
| $\alpha$ | Significance level |
| $\beta$ | Weighting factor 1 for calculation of multi-attribute decision criterion |
| $\gamma$ | Weighting factor 2 for calculation of multi-attribute decision criterion |
| $\delta$ | Power factor for calculation of decision criterion |
| $\varepsilon$ | Weighting factor for exponential smoothening in forecast dispatching |
| $\rho$ | Weighting factor for update of estimated fill level in forecast dispatching |
| $\tau$ | Weighting factor for calculation of average waiting time in feedback dispatching |
| $\omega$ | Decision criterion |
| $\varphi$ | Autoregressive factor |
| $\theta$ | Relative error or precision |

# List of abbreviations

| | |
|---|---|
| ACO | Ant colony optimization |
| ADVR | Adaptive distance vector routing |
| AGV | Automated guided vehicle |
| AR model | Autoregressive model |
| $B^2D^2$ | Bidding-based device dispatching |
| BR | Buffer replenishment |
| BH | Baggage handling |
| D | Dispatching strategy |
| DCV | Destination coded vehicle |
| DF test | Dickey-Fuller Test |
| DSR | Dynamic source routing |
| E | Storage location for empty vehicles in test system |
| EC | Entrance control dispatching rule |
| GSR | Global state routing |
| FCFS | First-come first-served |
| I | Intersection control strategy |
| IML | Institute for Material Flow and Logistics |
| KPI | Key performance indicator |
| KPSS test | Kwiatkowski–Phillips–Schmidt–Shin test |
| M | Merge control strategy |
| MOD STTF | Modified shortest-travel-time-first rule |
| Multi Att | Multi attribute dispatching rule |
| Multi-Mod | Modified multi attribute dispatching rule |
| OLSR | Optimized link state routing |
| PRFM | Performance |
| Q | Source in test system |
| S | Sink in test system |
| STDF | Shortest-travel-distance-first |
| STTF | Shortest-travel-time-first |
| R | Routing strategy |

# Acknowledgments

Das Schreiben der Danksagung ist für jeden Doktoranden wohl einer der schönsten Momente. Denn nun stellt sich langsam das Gefühl ein, die Dissertation zu einem Abschluss bringen zu können. Den Weg dorthin hätte ich nicht ohne eine Vielzahl von Unterstützern zurücklegen können. Eben jene Personen möchte ich an dieser Stelle erwähnen.

Zunächst gilt mein Dank Herrn Prof. Dr. Thorsten Schmidt für die Betreuung meiner Arbeit. Obwohl wir uns vorher nicht kannten, hatte ich das Gefühl, sehr schnell eine gute Arbeitsebene mit ihm gefunden zu haben. Für jede Anfrage und Diskussion stand er mir immer kurzfristig zur Verfügung. Mit seinen Fragen und Einwänden hat er wesentlich zum Ergebnis dieser Arbeit beigetragen.

Darüber hinaus möchte ich mich bei Dr. Frank Schulze bedanken. Als Betreuer meiner Diplomarbeit hat er mir bereits im Jahr 2007 die Idee einer Dissertation eingepflanzt und diese auch während meiner Zeit in der Privatwirtschaft immer wieder in Telefonaten am Leben erhalten. Während der Anfertigung der Dissertation stand er mir dann für organisatorische Aspekte ebenso zur Verfügung wie für die Diskussion meiner Ergebnisse. Und auch bei noch so überfallartigen Anfragen hat er den Telefonhörer nie aufgelegt.

Im gleichen Atemzug möchte ich Karsten Turek und Christian Hammel meinen Dank aussprechen. Beide haben mich bei allen Fragen rund um den PC-Pool des Lehrstuhls unterstützt. Ohne diese Hilfe wären die erforderlichen Simulationen nie in der gegebenen Zeit möglich gewesen. Karsten hat zudem vor allem während meiner Anfangszeit in vielen Diskussionen schnell die Grundidee der Arbeit geschärft und immer wieder neue Denkansätze geliefert.

Mein Dank gilt ebenfalls der 4flow AG. Nur deren flexibles Arbeitszeitmodell hat diese Arbeit überhaupt ermöglicht. Ich habe nicht nur Unterstützung in Sachen Infrastruktur erhalten, sondern auch immer wieder gern Ablenkung und Abwechslung durch die Kollegen.

Karsten Ploog von der Firma Still und Siegfried Seelos vom Flughafen München haben dankenswerter Weise die Eingangsdaten für die Simulationen zur Verfügung gestellt.

Mein besonderer Dank gilt meinem privaten Umfeld, das mich während des Verlaufs der Arbeit nie im Stich gelassen hat. Ich möchte mich bei allen Freunden in Berlin, München, Frankenberg, Frankfurt, Bielefeld, Los Angeles etc. für die Erinnerung daran bedanken, dass es auch ein Leben neben der Dissertation gibt. Ich hoffe, dazu in Zukunft wieder mehr beitragen zu können, als während der Endphase der Arbeit.

Ich danke meinen Eltern und meinen Brüdern, dass sie mich immer wieder motiviert und irgendwie daran geglaubt haben, dass ich diese Dissertation zu einem Ende bringen kann. Vor allem in Situationen, in denen ich mir das gar nicht vorstellen konnte.

Nur Isabel war vielleicht noch ein Stück näher dran an meinen mentalen Tiefflügen und Motivationslöchern. Vielen Dank für die Geduld, die Leidensfähigkeit und die vielen aufmunternden Worte, ganz egal ob per Mail, Telefon oder persönlich. Ich weiß, dass das nicht immer leicht war. Du hast auf jeden Fall einen ganz eigenen Anteil an dieser Arbeit. Die entspannte Version meines Ichs sollte jetzt kurzfristig wieder zur Verfügung stehen.

Nicht zu vergessen sind auch „Hot Water Music“ und „Boy Sets Fire“. Ohne die wären die letzten beiden Jahre ganz schön langweilig gewesen.

And Kieran, thanks for proof-reading this document.

Nils Klein

Berlin, Dezember 2012

# 1 Introduction

The importance of the logistics industry has grown continuously during the last few decades. The overall transport volume and thereby the revenues of logistics companies around the world have shown a constant increase and are expected to grow further in the future (see HAHN-WOERNLE 2010). According to a forecast project which was funded by the German ministry of traffic, the transport volume within the German logistics infrastructure can be expected to grow by 50% between 2007 and 2050 (ICKERT ET AL. 2007). Similar developments are forecast for other countries. The growth trend is enabled by three main drivers:

- Globalization
- Mass customization
- Shorter product life cycles

Nowadays, companies and customers no longer have to stick to their domestic markets. Companies can make their goods available around the globe and accessible to everybody. Customers can buy products from virtually any place via the internet. This leads to an increase in global logistics demand.

Besides using the opportunity to buy everywhere in the world, the customers favor products with individual characteristics. Companies try to satisfy this need with mass customization. They either allow the customers to choose from a wide variety of product variants or offer additional services. These sales strategies affect logistics operations. On the one hand, the size of shipments decreases while there is an overall increase in shipment quantity. On the other hand, logistics companies have to be able to offer value-adding services.

But the customers are not only looking for customized products. In addition, new products are required more frequently. This is not only induced by the customers, but also by the selling companies that try to increase their revenues by creating additional demand. For logistics this leads to an increase of the transport volumes and means at the same time that logistics systems have to deal with frequently changing product characteristics and demand patterns. Rigid systems become less desirable.

There are other trends and changes in the way companies and customers do business today. But the previously mentioned aspects seem to be the most influential ones for logistics operations. In order to cope with the increasing complexity and dynamics which are created by the mentioned trends, future logistics systems need to show the following characteristics (see FURMANS ET AL. 2011):

- Flexibility (e.g. modular design)
- Reconfigurability
- High availability
- Plug & play configurability (standardization of interfaces and communication)
- Scalability
- Reusability
- Adaptivity
- Energy-efficiency and resource-efficiency

Viewing a logistics transport network as a graph of nodes and arcs, the increasing inter-company transport requirements do not only affect the transport volumes between the nodes of the graph. They also highly influence the operations at the nodes where the goods are produced, modified and handled. Although manual or semi-automated systems would be the most suitable choice for achieving the required flexibility at those nodes, automated systems have to be used in many cases for logistics processes. It might be the pure throughput requirements which make those systems economically reasonable. Other reasons for their usage are labor costs or the product characteristics. It can be expected that demographic changes make automated systems even more important in the future (see HAHN-WOERNLE 2010).

State-of-the-art automated logistics systems show weaknesses in fulfilling the requirements which were stated above (NIEKE 2010). These systems consist of hardware and software components. The centrally structured control systems especially limit the functional capabilities. The control algorithms are tailored to the specific customer requirements and often poorly documented. They only achieve an acceptable performance for one hardware configuration. Therefore the systems are extremely hard to modify. Changes always require a test period and in many cases a costly shutdown of the whole system. The system flexibility is limited to the extension scenarios which were already considered during the planning phase. During operation the central control systems have a single point of failure which requires a costly “warm” or “hot” stand-by system. Hardware components from different manufacturers are hard to integrate and combine.

Knowing about the weaknesses of state-of-the-art automated systems and having the increasing challenges of dynamic complexity in logistics systems in mind, the concept of “decentral control” has gained more and more attention in material handling literature during the last few years. Decentral control systems are said to offer a number of advantages compared to centrally controlled systems (SCHOLZ-REITER ET AL. 2009; FAY ET AL. 2008; TEN HOMPEL ET AL. 2010). Decentral systems should be easier to implement and configure. Based on their modular structure, these systems can be set up easily. Changes do not result in high costs. Excellent flexibility, reconfigurability and expandability can be achieved. The systems ought to show a better robustness regarding disturbances. There is no single point of failure and after a disruption the systems can return quickly to working conditions.

In complex systems optimal decision making is hard to achieve. The necessary information is either not available at all or already outdated once the decision is to be made. Therefore decentral systems use local information and rather simple decision rules. It is expected that a desirable overall performance emerges from these local decisions and interactions of the logistics entities. In the best

case the simplicity of decision making would lead to a higher efficiency than in central systems (SCHOLZ-REITER 2008).

The approaches for developing decentral systems which can deal with the current and future logistics requirements differ in their radicalness. Some approaches try to tackle the hardware and the software of the system at the same time. The KARIS system (HIPPENMEYER ET AL. 2009) or the Flexconveyor (MAYER 2009) are examples of this approach. The idea is to develop small scale conveyor units which can be combined using a plug & play functionality in order to create extremely flexible material handling systems. The Cognilog project at the University of Hannover follows a similar approach (OVERMEYER ET AL. 2012). The Fraunhofer Institute for Material Flow and Logistics (IML) develops several concepts. The most progressive one favors a cellular transport system (TEN HOMPEL ET AL. 2012). The vehicles in this system move completely independently and coordinate their behavior amongst themselves. Other concepts at the IML focus on developing new control systems for existing material handling systems. They use the available technical hardware infrastructure. At the University of Bremen a whole research cluster develops control strategies for networks of intelligent logistics entities. The entities are able to make autonomous decision in heterarchical and adaptive logistics systems (see e.g. SCHOLZ-REITER ET AL. 2009). Other researchers in Asia and Denmark follow the paradigm of multi-agent systems for the control of material handling systems (see LAU & WOO 2008 or HALLENBORG 2007).

According to their advantages, decentral control systems should be the perfect choice for developing and implementing state-of-the-art material handling systems. But up to now, a couple of questions remains unanswered when decentral control concepts are developed. These questions motivate the research which is the subject of this thesis. The next section reviews them in detail.

## 1.1 Research motivation

As already mentioned, many researchers work on developing prototypes in order to implement decentral control systems with self-organizing properties. In those systems logistics objects make decisions autonomously. The aim is to achieve emergent system characteristics which result from local interactions and coordination (SCHOLZ-REITER ET AL. 2007). Looking at the prototype systems and existing publications, shortcoming with respect to four different dimensions can be recognized. We will briefly state these shortcomings here and verify them with our literature review in Chapter 3.

1. Violations of the decentral paradigm

The existing implementations do not always follow a completely decentral approach. Sometimes compromises are necessary in order to ensure that the systems work properly and achieve the required goals. This means on the one hand, that central mediators or similar instances are used. On the other hand, those systems often do not use local information. They rather start with different types of local information which are then aggregated to global information for decision making. From a conceptual point of view, optimal decisions are not possible without global information (see SCHOLZ-REITER ET AL. 2008). As the development of a control system always strives for optimal performance, the usage of a coordination instance or the aggregation of local information makes sense intuitively and is tempting. But this approach leads rather to distributed systems or even to the idea of distributed problem solving instead of decentral control (see BOND & GASSER (1988) for a classification and comparison of distributed problem solving and decentral multiagent systems). The paradigm of decentral control states that a desirable system performance should emerge from local decision making and interactions. Optimality is either expected to arise automatically from the local interactions or treated explicitly for the advantages of decentral control systems which were mentioned above.

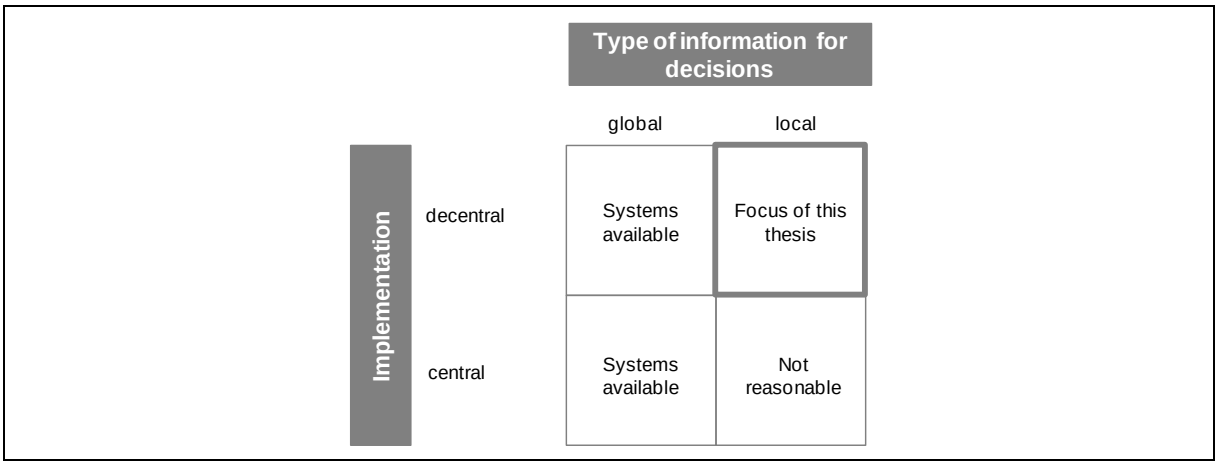



Figure 1: Matrix of implementation and information types

A clear differentiation of implementation structure and used information type is currently lacking in scientific literature. Central systems which use global information and decentrally implemented systems that either use a central coordination instance or aggregate local information to global knowledge are available. Central systems which use local information would not be reasonable. The major objective of this study is to develop and assess decentral control strategies which are only based on local information and do not use any kind of central entity (see figure above).

2. Technical limitations

To focus on local information only is also relevant because of another property of current decentral implementations. They usually require that the logistics entities communicate with each other in order to negotiate, coordinate, cooperate etc. In a heterarchical system in the worst case all entities communicate with each other and exchange information in order to make decisions. The result is an extreme communication overhead which increases with the number of entities in the system. This means that an infrastructure is necessary which can handle the communication requirements and does not limit the real-time functionalities of those systems. This is currently still a major bottleneck when implementing decentral systems. Many systems are limited to less than 100 or even 10 entities because of the communication overhead.

By using only local information for decision making the communication overhead can be reduced to a minimum. Only local communication is required to exchange information. But no information needs to be propagated in the network. A decentral control system which follows this approach would bypass the technical limitations for decentral systems which exist today. Accordingly, the control algorithms would become applicable for systems with more entities. This thesis evaluates if the reduction of communication is possible and how it affects the system performance.

3. No proper performance measurement

For an exhaustive evaluation of the decentral approach its qualitative advantages are to be supplemented by quantifying the performance impact. Many publications on decentral control systems are of a rather conceptual nature (see e.g. CHISU ET AL. 2010 or GÖHRING & LORENZ 2010). They do not provide a proper performance measurement which allows a comparison of central and decentral control strategies. In some cases only artificial input data is used for performance measurement. Additionally, it is required to verify some of the qualitative advantages, e.g. the robustness of decentral systems.

Currently, the approaches of different authors can hardly be compared. Material handling researchers traditionally focus either on specific real-world problems or develop test scenarios which are tailored to a certain study. There are no general test environments as they are used in other scientific disciplines (see SCHREIBER & FAY (2011) for a standardization approach in manufacturing control). Results and solutions are in most cases not comparable and not transferable.

This thesis therefore develops a generic test environment which can be used by other researchers to make the results of different control strategies comparable. The evaluation of different performance indicators shall allow a proper assessment of the decentral control paradigm. Two real-world input data sets are used. Several disturbance scenarios are analyzed to develop an understanding of the robustness of the systems.

4. Limited functional scope

In material handling systems several control functions can be distinguished. Besides routing, empty vehicle dispatching, merge control and divert control are the functionalities which influence the overall system performance. Most publications focus on decentral routing. Switch and merge control are in some cases mentioned in this context. Dispatching and especially decentral dispatching are hardly considered in any prototype system. We will therefore stress this aspect of the control system and also highlight interdependences with other control strategies whenever they become relevant.

Summarizing, this thesis aims at implementing and quantifying the performance impact of decentral dispatching strategies which only use local information. The subjects are intralogistics transport systems where transport orders enter the system without prior notice or information, i.e. planning is not possible. Besides known dispatching approaches, new mechanisms are developed and implemented. This is necessary as there are only a few truly decentral strategies available.

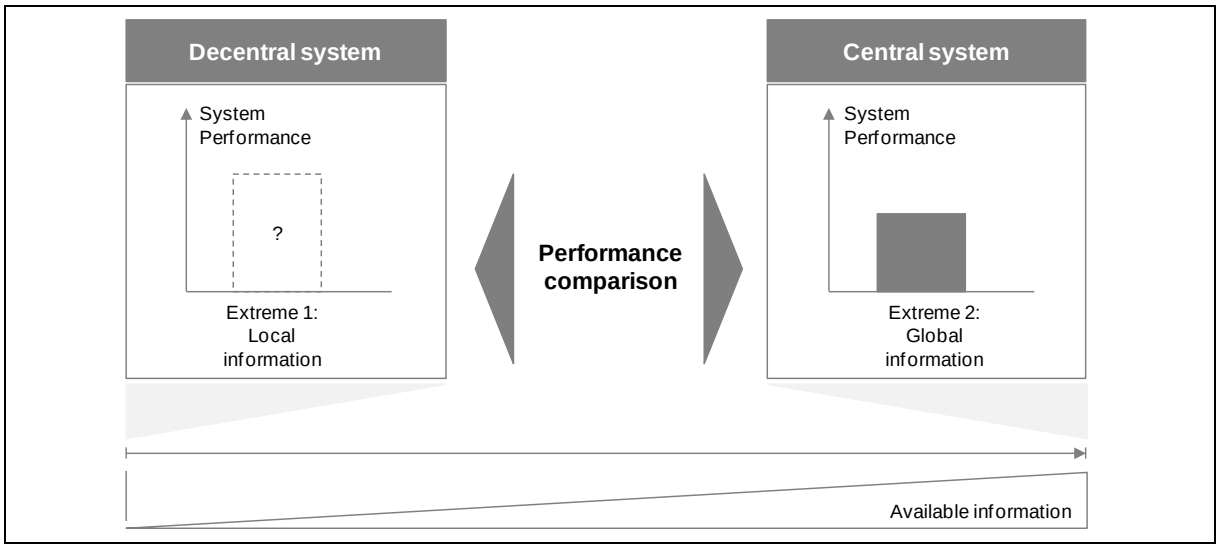



Figure 2: Focus of this thesis

Central control with global information is used as a benchmark. Thereby the two extremes of information availability are compared (see figure above). It is investigated how the known information quality influences the performance of the system and which performance level can be achieved with local information only. The applicable trade-offs (information vs. performance) are analyzed with respect to efficiency, i.e. the requirement of additional vehicles and vehicle movement. This shall

evaluate the hypothesis of DE KOSTER & VAN DER MEER (1998) who already stated in 1998 that decentral systems are very simple but less efficient than central systems. The findings will allow a judgment of the applicability of decentral systems in real-world scenarios. It might be most efficient to use an information quality between the two extremes. All control strategies are tested and evaluated using the AutoMod simulation software with a generic test layout. The system is scalable and able to represent different degrees of system complexity. The amount of vehicles can be varied. Several series of simulation experiments are carried out with different parameter settings. Disturbance scenarios are considered to test the robustness of the implemented control algorithms.

## 1.2 Outline of the thesis

The outline of the thesis concludes this introductory chapter. Following that, Chapter 2 uses several classification schemes and definitions to clarify the exact scope of the thesis. Starting with a very wide definition of logistics, the important concepts “material handling” and “transport systems” are derived. The chapter is also used to clarify our understanding of a decentral control system and to distinguish the three control strategy types which can typically be found in intralogistics transport systems. Additionally, simulation is introduced as a tool for analyzing complex material handling systems. Key performance indicators for measuring the system performance are defined.

The succeeding chapter reviews the existing scientific literature on decentral control in intralogistics transport systems. The latest findings are presented with respect to the three control strategy types of dispatching, routing and intersection control. The review is used to illustrate the four shortcomings which have been carved out for decentral systems (see Chapter 1.1).

Afterwards, the generic test environment is introduced. Its basic functionality and the implemented AutoMod simulation model are described conceptually.

Chapter 5 contains a detailed description of the control strategies which have been implemented. If they are considered valuable for the understanding of decentral control systems, “lessons learned” during the implementation process are highlighted.

The next chapter focuses on the simulation methodology. It explains the major input parameters and how they are combined for the different simulation experiments. The input data sets are introduced and their core characteristics are briefly discussed. Additionally, the statistical approach for calculating the performance indicators based on the output data is described.

The simulation methodology is the foundation for presenting the results of the experiments in Chapter 7. The different scenarios are analyzed with respect to throughput and average service times. Additional vehicle requirements and vehicle movement figures are derived. Some further analyses illustrate adaptive capabilities and other characteristics of the decentral system. The following chapters contain the general limitations of the thesis approach and the conclusions which can be drawn.

The last chapter summarizes the results and points to future research questions.

# 2 Scope of this study

By defining and distinguishing several important concepts, this chapter shapes the scope of the thesis. Starting with a very broad view on logistics, each section will narrow down the subject.

## 2.1 Logistics and intralogistics

The overall topic of this thesis is logistics. There exist numerous definitions of this term. From a very universal perspective, logistics can be defined as the design and control of logistics systems. It is the science of developing and operating the required information and communication systems as well as the physical components for solving a certain logistical problem (see FLEISCHMANN 2003). VAHRENKAMP (2007) names four core activities of logistics:

- Store
- Transport
- Distribute
- Collect/pick

Besides those core functionalities additional tasks such as handling, processing, testing and packaging are part of logistics activities. Each logistics system contains several of these logistics tasks. A chain of tasks can be understood as a logistics process that transforms logistics objects from an input state to an output state.

The first important distinction to be made with respect to logistics is between inter-company logistics and intralogistics. While the former refers mainly to transportation between companies and their facilities, the latter is used for describing all logistics activities within a certain facility. We use an exemplary network model of an assembly plant for clarifying the differences. In a first step we look at the inter-company network (see figure below). For simplification we only consider the material flow and neglect information processes. The network shows the assembly plant in its very center. The plant is supplied by a number of other companies. Some of them deliver directly to the plant and other inbound shipments are consolidated in a warehouse which might be managed and owned by a logistics

service provider. On the outbound side the same situation can be found. A number of customers is supplied directly from the plant and an external warehouse is used to serve smaller demands or those in remote areas.

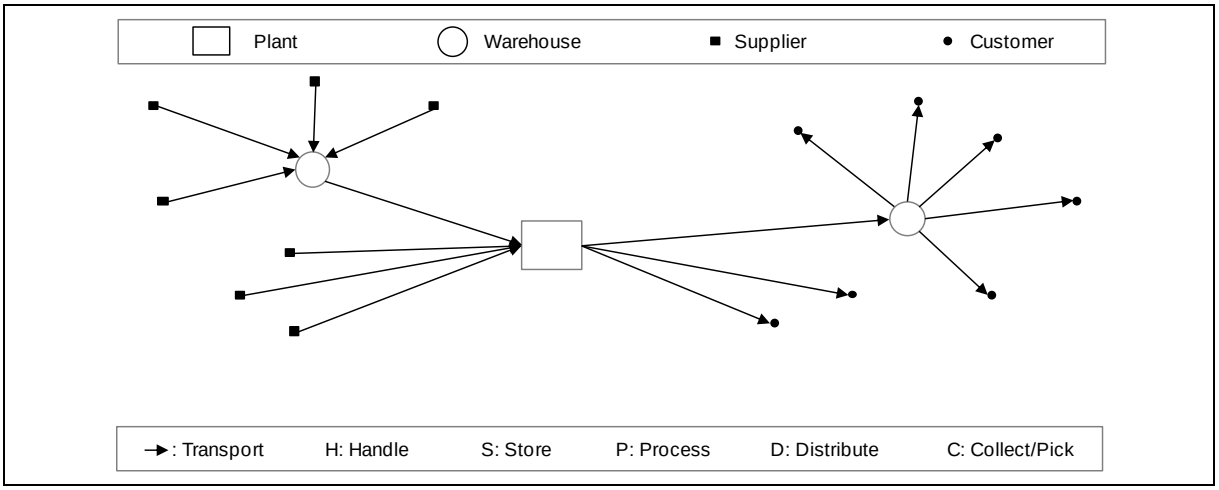



Figure 3: Exemplary inter-company logistics network of an assembly plant

The intralogistics activities become visible when we do not look at the arcs, but at the nodes of this network. Within the facilities, i.e. plants and warehouses at each node, an own intralogistics network is in place. The figure below shows a simplified version of the intralogistics activity network of the assembly plant. It connects the receiving process via the inbound warehouse to the production. After the production processes have been completed, the goods are either directly brought to the loading process or are stored in the warehouse. The figure also suggests intralogistics networks for one of the warehouses and one supplier within the inter-company network. This shall indicate that each of the facilities within the network has its own intralogistics network.

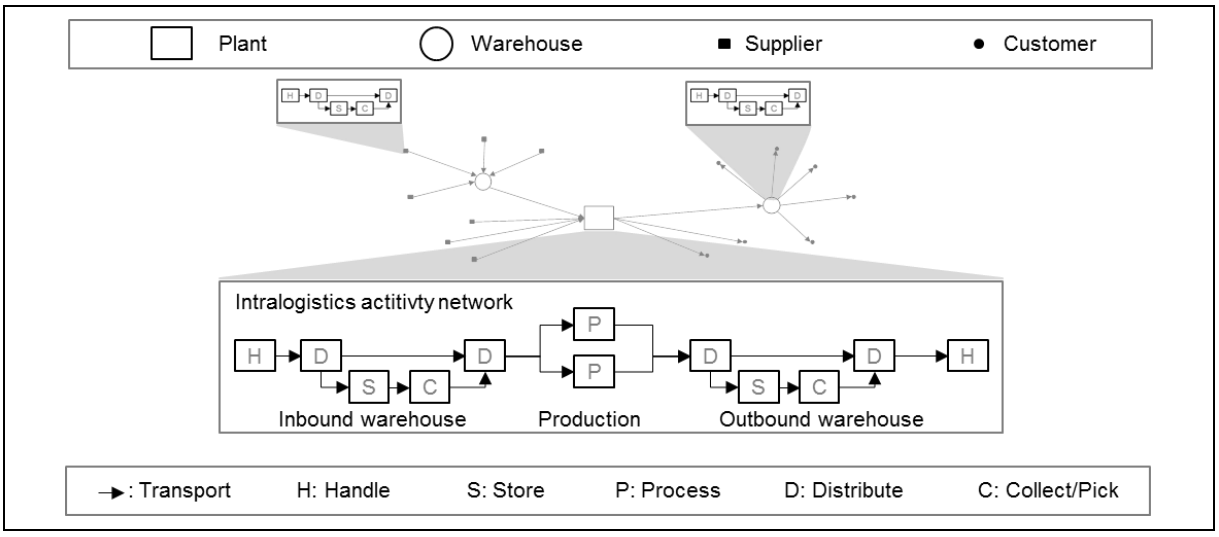



Figure 4: Simplified intralogistics activity network of an assembly plant

For the rest of the thesis we will only consider intralogistics networks. This means we will look at logistics as defined above, but only from the perspective of operations within a certain facility. The assembly plant and the warehouses are only examples. Intralogistics networks can be found at several stages of logistics operations and have very different characteristics. Baggage handling systems at airports, cross-docks within distribution networks or port container terminals are other examples.

## 2.2 Material handling and material handling systems

Having clarified our understanding of logistics, i.e. of intralogistics, we take our definitions one step further. In total, there are three different perspectives for looking at logistics activities (TEN HOMPEL ET AL. 2007):

- Economics
- Informatics
- Engineering

Economics takes a very broad view and considers logistics as an approach for managing the development and operational execution of efficient logistics processes. Informatics mainly focuses on systems that enable the information flow within logistics networks. Engineering looks at the technical facets of logistics. This perspective is important for intralogistics. The design and control of the technical components and subsystems of intralogistics systems is usually referred to as material handling. It focuses on the material flow on an operational level.

Returning to the example of the assembly plant, material handling includes the handling of goods in the inbound warehouse (e.g. unloading, material reception, material put-away, retrieving), the production area (line supply, removal of empties) and the outbound warehouse (material put-away, picking, loading trucks). It does not only refer to the execution of processes, but also to the design of the technical components which are required for those processes. In this context forklift trucks, racks or picking systems might be necessary systems.

The technical systems which are required to perform the material handling activities are called material handling systems. There are various forms of material handling systems. They consist of a general infrastructure, technical material handling equipment, personnel, a control system and a communication system. Depending on the application either manual systems or mechanized systems are in place. Mechanized systems which can be controlled automatically are called automated material handling systems (see LE-ANH 2005).

In summary, material handling within the context of this thesis can be defined as the design and control of the necessary manual or mechanized material handling systems for carrying out the core logistics functionalities on the operational level of intralogistics. Automated material handling systems are the core subject of this thesis.

## 2.3 Transport systems

Based on the definition of material handling systems, we can more specifically look at this system type. There are various approaches for classifying material handling systems (LE-ANH 2005). We use the core logistics functionalities for our differentiation. Among the most important system types are:

- Storage systems
- Transport systems
- Sorting systems
- Distribution systems
- Picking systems
- Packaging systems

Each of the systems is named according to its main functionality. This thesis focuses on automated transport systems. They are used to transport goods from sources to sinks within facilities. We will

refer to the transported entities as loads. Alternatively, the terms transport order and transport request might be used. Each load comprises exactly one unit of transported goods. It might be thought of as a pallet which contains one or more articles.

Although the core task of a transport system is to solve a transport problem, those systems may serve other purposes at the same time. During a transport the system always has some kind of buffering or storage functionality. A transport system might in addition be connected to a storage system or other types of systems. The usage of transportation systems for picking activities is very common. Being aware of those combinations, we will consider systems in which transportation is the main material handling purpose.

As for material handling systems, there are also several approaches for classifying intralogistics transport systems. We follow the approach of TEN HOMPEL ET AL. (2007) and use the technical design as the criterion for our classification (see figure below). The interested reader is referred to the book of TEN HOMPEL ET AL. (2007) to find more information on the different system types and their characteristics.

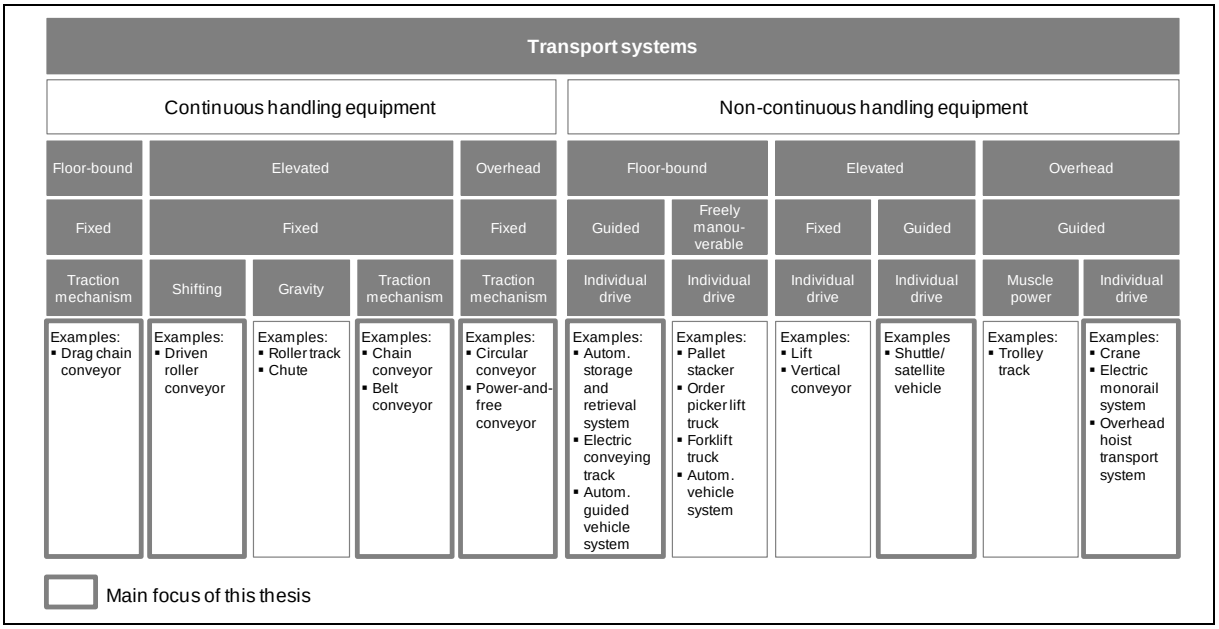



Figure 5: Classification of transport systems (compare TEN HOMPEL ET AL. 2007)

The classification is used to shape our understanding of intralogistics transport systems. Following the focus which is highlighted in the figure above, two system types are mainly considered:

- Continuous transport systems which use traction mechanisms or shifting
- Non-continuous transport systems with guided vehicles that have an individual drive

Following a definition of GUDEHUS (2005) the first system type is named a conveyor system and the second is referred to as a vehicle system. In conveyor systems the transport network itself is driven, while in vehicles systems each vehicle has an individual drive and moves on a passive network. There are various different versions of both system types. In the following sections we will review some examples and further clarify our understanding of transport systems in the context of this thesis.

Conveyor systems are considered first. For these systems two different types have to be distinguished for the purpose of our analysis. There are systems that use carriers for load transport and systems

which work without carriers. As our goal is to evaluate dispatching strategies only systems that use carriers are relevant. Baggage handling systems as they are used at airports can illustrate the difference. These systems can be designed in two different ways when conveyor systems are used. Either the baggage is transported using belt conveyors or plastic tubs are used as load carriers. The latter system type is the focus of this study. The dispatching of empty tubs has an impact on system performance. System behavior of a carrier-based conveyor system can be influenced whenever a decision point is reached. Decision points are loading stations, unloading stations, merges and switches (see figure below). Loading stations are the sources of the system where loads have to be picked up. The loads need to be delivered to the unloading stations. Merges and switches are required to determine the route choice and right of way prioritization of vehicles while they move in the transport network.

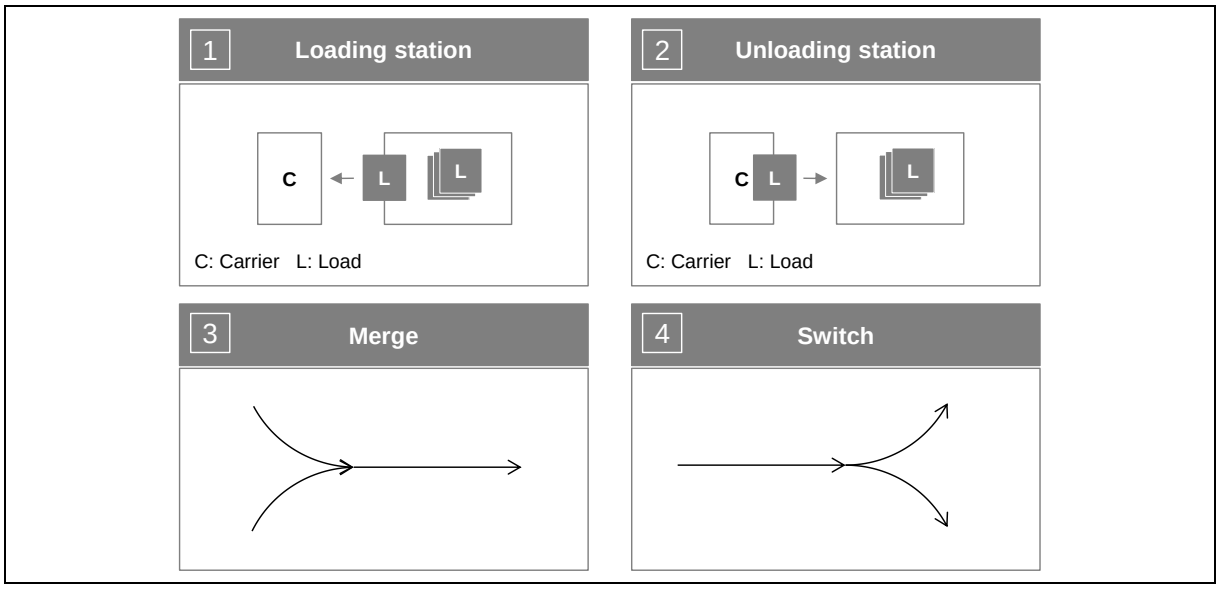



Figure 6: Decision points in a carrier-based conveyor system

Two different lines of thought are possible. Either the decision competence is given to the carrier which communicates with the decision point to technically implement the decision or the decision point itself makes the decision. Having a local focus, the decision points are definitely able to collect more information from all passing vehicles than a single vehicle can perceive in the system. Therefore we consider the decision points to have the responsibility for decision making. The second perspective can usually be found in literature about autonomous decision making of logistics objects. It puts the vehicles in the center of the decision making process. This is not a contradictive perspective. An autonomous vehicle would also make decisions based on the local information which it perceives at the decision points. If it is allowed to locally communicate with the decision point, both use the same information. Therefore it does not matter in the end if the vehicle or the decision point is given priority. The most important point is that decisions are made based on local information.

Following this understanding of empty carrier decision making, it becomes evident that our thoughts are not only limited to baggage handling systems which were mentioned as an example. All systems which use carriers to transport loads will have to answer similar questions.

Next, we will consider the second transport system type which is considered relevant to our dispatching analysis. In vehicle systems the same dispatching tasks as in carrier-based conveyor systems need to be solved. The vehicle-based systems can show very different technical characteristics. We therefore only highlight the most important systems with respect to this thesis.

Electric conveying track systems are the first example. They might be used as an alternative to conveyor systems in baggage handling. Destination-coded vehicles (DCV) transport the baggage in the track network. They have an own drive and travel on tracks which are used for energy supply. Each vehicle is therefore an independent unit. Electric monorail systems and overhead hoist systems feature the same system characteristics. The main difference is that an overhead track is used. Electric monorail systems can be found in various different warehouse and manufacturing environments. The overhead hoist systems are a special transport system type for wafer production facilities.

Automated guided vehicles (AGV) are another example for vehicle-based transport systems. The main difference compared to the systems which were mentioned above is the guidance system which defines the transport network. In AGV systems either physical guidance lines or virtual guidance can be used. Neither one supplies energy. Physical guidance lines can be inductive, optical or magnetic. Virtual guidance uses software than combines position information and an internal environment model to determine the behavior of the vehicles.

The conceptual track layout of vehicle systems does not differ excessively from conveyor systems. There are the same four types of decision points and decision making will be very similar to the processes in conveyor systems. Although the vehicles are driven and could actively decide about their next actions, the decision points can still be thought of as active components that influence the system behavior. The same conclusions as for the conveyor systems apply. However, there are slight differences in AGV systems. Loading and unloading station still exist, but merges and switches are not real technical components which can make decisions. In case of physical guidance systems the guidance lines can still be used to influence the vehicle behavior in so-called block sections. Those help to prevent collisions and deadlocks at merges. But as soon as only virtual guidance is used, there are no technical means to physically influence the vehicle behavior at merges and switches. The decision making process is completely in the hands of the vehicles and their internal software.

During the remainder of this thesis, the transport systems are considered from a very general perspective. Although we do not explicitly refer to technical specifications, they can always be thought of as either a conveyor system with carriers or a vehicle-based system. We will use the term vehicle, but this could also refer to any other kind of carrier, e.g. a plastic tub. Additionally, we use the concept of loading stations, unloading stations, merges and switches. As mentioned, the latter two terms become indistinct in AGV systems. We assume that either an inter-vehicle communication or a specific control unit at intersections can fulfill the same tasks as switches and merges do in other systems. A fifth type of decision point is introduced. Storage locations (sometimes also called “dwell points”, “depots” or “home locations”) are loop sidings which are used to park idle vehicles (see figure below). The arrival at a storage location can initiate decision processes. Storage locations are often located close to loading stations to improve the vehicle availability at those stations. According to HU & ENGBELU (2000) there are two basic assumptions which can be made for the handling of empty vehicles. They can either circulate in the transport network due to limited space for parking vehicles (see e.g. LE-ANH & DE KOSTER 2004) or be parked at storage locations with sufficient capacity. We follow the latter approach and assume an adequate size of the storage locations.

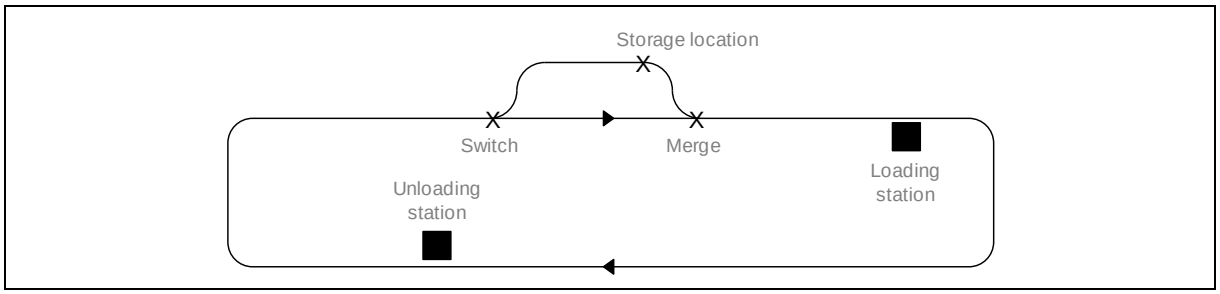



Figure 7: Decision points in a single-loop path layout with loop siding

Our focus is on transport systems with many vehicles. This is an area where little research has been contributed to. The term "many vehicles" denotes systems where the number of vehicles is much larger than the number of stations (see LE-ANH & DE KOSTER 2004). The exact number of used vehicles does always depend on the systems size. But we want to create and analyze systems which require up to several hundred vehicles.

## 2.4 Designing transport systems

With our understanding of transport systems in mind, we can now look at the major factors which determine their performance. From a design perspective there are mainly three aspects which need to be considered:

- Layout
- Vehicles
- Control system

The design of a transport system is an iterative process. An initial version of the system is developed and afterwards the three mentioned factors are changed and adjusted until the required system performance is achieved. Knowledge about the interdependences between the different factors is the prerequisite for a proper understanding of the system characteristics and the system behavior.

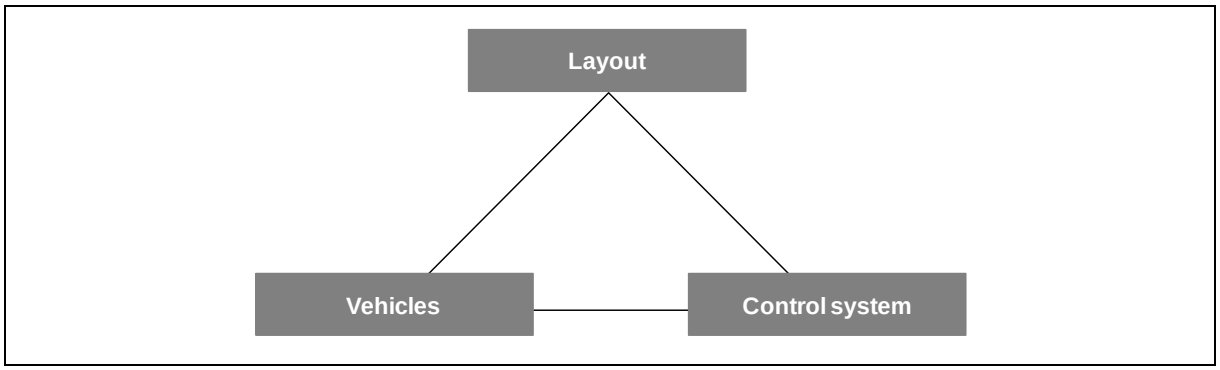



Figure 8: Interdependent design aspects of a transport system

The following sections give a brief introduction to each of the design aspects and contain the assumptions which are relevant for this thesis. For more details about system design, especially with respect to AGV systems refer to QIU ET AL. (2002), VAN DER MEER (2000) and VIS (2006).

**Layout**
The layout design is often the first step in the design process. Although there are a number of techniques available which can support the layout design (see LE-ANH 2005), they are usually only applicable for a very specific type of design problem. Therefore, design decisions are often made based on experience and common sense instead of using a systematic approach (VAN DER MEER 2000). Designing the layout covers several design decisions:

- Course of the track, i.e. the path layout with switches, merges and storage locations
- Position of loading and unloading stations
- Characteristics of the track itself:
    - Unidirectional flow vs. bidirectional flow
    - Single lane vs. multiple lanes
- Technology of the track layout, e.g. what kind of conveyors are used or which guidance system for automated vehicles

This thesis uses a single lane, unidirectional path layout with several loading and unloading stations. The layout is assumed to be predefined. This means the first of the three factors which determines the system performance is kept fixed and not varied during the analysis.

**Vehicles**
The vehicles are the second factor which needs to be considered when developing a transport system. Firstly, the vehicles need to be designed. Their capacity has to be determined and technical factors such as velocity, acceleration etc. need to be selected. This design step refers to the technological side of the vehicles used.

In a second step, the required number of vehicles needs to be calculated. This is usually done after the layout has been developed. Depending on the system, the vehicles can be expensive and account for a large share of the overall investment (e.g. in an AGV system). The aim is therefore to find the minimum number of vehicles which achieves a certain performance requirement. The layout and the implemented control strategies are the major drivers for the required number of vehicles. Due to the complexity of the calculations, the number of vehicles can only be estimated in most conventional layouts (VAN DER MEER 2000).

For the purpose of our analysis we do not consider details regarding vehicle technology. However, the following assumptions are required as they affect our evaluations:

- The technical vehicle characteristics (acceleration etc.) are set according to the default values of the used simulation software (see Chapter 6.1.4 for details)
- Each vehicle is able to transport only one load at a time (single load vehicle)
- All vehicles show 100% reliability
- No additional downtimes (battery management, maintenance etc.) have to be considered
- Vehicles prevent collisions by slowing down and stopping when they approach another vehicle, i.e. vehicles can sense forward

During the analysis the number of vehicles in the system is used as an input variable. It is examined how its variation affects the system performance in combination with the control system.

**Control system**
While the layout and the vehicle design refer to the hardware components of the transport system, the control system design process creates the software that is required to run the system. In the control system the logic for the system operation is defined. It contains the decision rules that ultimately determine the behavior of the system and its performance. While the required number of vehicles mainly influences the initial investment of a transport system, the control system drives the operating costs. It is responsible for energy consumption and maintenance requirements. Among the frequently mentioned control strategies are:

- Scheduling / dispatching
- Routing
- Parking
- Collision prevention
- Deadlock prevention and/or resolution

Depending on the system, some specific control strategies, e.g. battery management for AGV systems are required. We conclude this sub-chapter with some additional assumptions which are relevant for the design of the control system:

- Each transport order consists of one load
- Vehicles are not used for buffering or storing loads (e.g. early baggage storage in baggage handling systems, see HALLENBORG (2007) for details)

The next sections take a detailed look at the structure of control systems and control strategies.

## 2.5 Control systems for transport systems

Having identified control systems as one of the three major factors which influence the performance of transport systems, we will now take a detailed look at this concept. After a brief introduction to the general functionalities of control systems the different structural design approaches are introduced. The last sub chapter distinguishes three control strategy types which are used for controlling transport systems.

### 2.5.1 Basic concepts

Roughly speaking, control systems in automated intralogistics transport systems use sensors to perceive the system status and make decisions to influence the system behavior by triggering actuators. The sensors can either simply observe the system (e.g. optical sensors which determine if a tub has arrived in a certain area of the system) or be represented by more intelligent components which provide a certain feedback signal whenever an action has been completed. An example for the latter could be a vehicle which actively provides the information that it has passed a certain reference point. In this case the vehicle itself would use some kind of sensor, process its signal and communicate only the result to the overall control system. The implications of different decision making hierarchies will be discussed in the remainder of this section.

The actuators are technical components, mostly drives, which are triggered by the control system. They perform actions which the control system found necessary to influence the current system status. An actuator could for example be the drive which controls the direction that a switch sends arriving vehicles to.

Sensors, actuators and the control systems have to be connected by an appropriate communication infrastructure which is able to achieve the required data interchange rates (see e.g. TEN HOMPEL & SCHMIDT (2005) for different interface standards and data connections or DIN 19226 for a general classification scheme of control systems).

As already mentioned, the control system is responsible for decision making in response to a certain system status or a certain event which occurred in the system. It can be understood as a software program which is running on suitable IT hardware. The decision making process requires a certain set of control strategies. These strategies are algorithms which determine an action based on a number of input variables. The control system operationalizes the strategies. Before we define three different strategy types, we look at another aspect which influences the functional characteristics of a control system: its overall structure.

### 2.5.2 Central vs. decentral structure of control systems

The structure of a control system describes its hierarchy of decision making. Material handling literature defines mainly three different control system structures. The main distinction is made between central and decentral control systems. In a central system decisions are made by one central component. This component controls all other subsystems which it is connected to. The central control component has global knowledge about the current system status and should be able to make optimal decisions. The disadvantages of central systems (e.g. limited flexibility, single point of failure) have already been discussed above. In a decentral system the decision making competence is distributed to a number of components. Each of those components has a smaller functionality than the core component in a central system and individually controls a certain part of the system. All the control components are connected to each other in a heterarchical system. This means they are all on the same hierarchy level and accordingly, there is no hierarchy. JÜNEMANN & BEYER (1998) state that purely decentral systems only show limited functionality and require a higher-ranking coordination mechanism. This is a compromise solution which tries to combine the advantages of central and decentral control. Systems which use this kind of coordination component are referred to as distributed systems (see figure below).

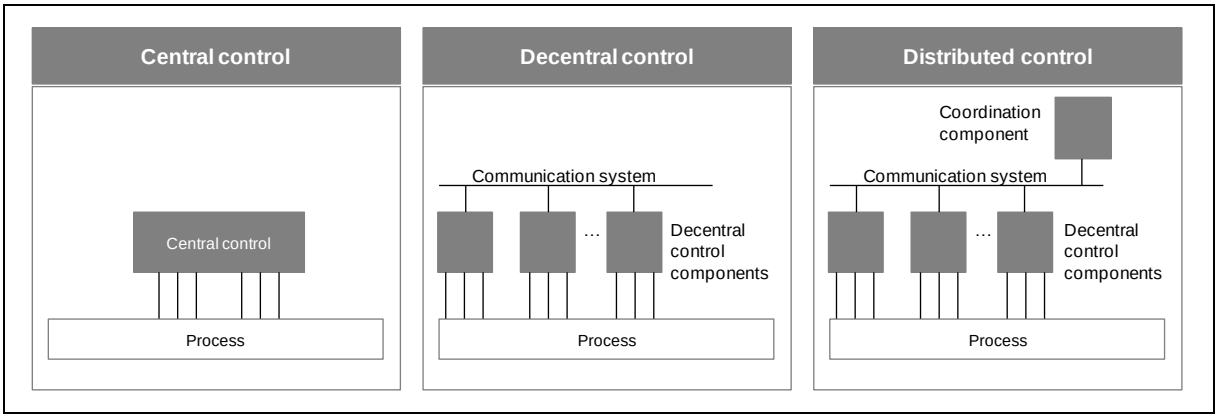



Figure 9: Basic structure of control systems (compare JÜNEMANN & BEYER 1998)

Centrally and hierarchically structured systems are the most common forms of control systems today (NIEKE 2010). Their hierarchical structure is a consequence of the single central decision unit. A layer model summarizes this control structure (compare VDMA Norm 15276) and is shown in the figure below. It illustrates the risk of a single point of failure. The transport system control unit is usually a subsystem of e.g. a warehouse management system. Its hierarchical structure is part of an overall control system hierarchy.

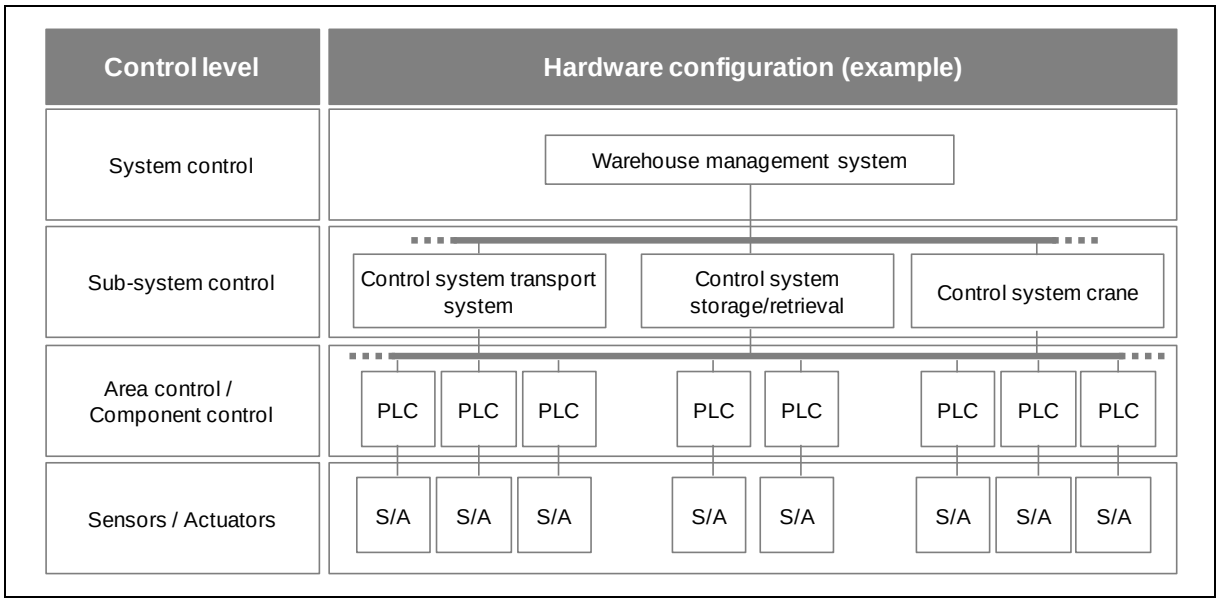



Figure 10: Hierarchical structure of a central control system (compare KLEIN 2008)

As we have already argued and will stress during our literature review, most of the decentral systems which have been proposed in the scientific literature do not fully explore the initial meaning of this term. In order to further elaborate this topic and to describe which structural configurations are applied, we introduce an additional concept. Following HOFMEISTER ET AL. (2010) there is global and local information in a control system. Local information is limited to the perception of the decision-making entity. This means that there is no information exchange in the system. Each entity only knows what it can perceive locally and has its own routines to process this information and to make decisions. On the other hand, an entity has global information when there is a communication infrastructure which can be used to acquire information from all parts of the system.

Using this definition of information quality to supplement the structural classification of control systems helps to stress our hypothesis of inconsistent terminological use and limited exploitation of the decentral paradigm. Due to its structural appearance, a central control system always uses global information for its decisions. The communication network facilitates global information exchange. And as the information is available anyway, it should be used for decision making. Global information should be expected to yield the best system performance. Central systems would be placed in the lower left corner of Figure 1 which has been introduced in the first chapter.

Many of the decentral systems which are proposed by various scientists are implemented decentrally. But the decision processes are not restricted to local information. The main idea of decentral systems is to create self-organizing systems. Their most important property is the emergence of a global order from local decisions and interactions (see HEYLIGHEN (2001) for a summary of characteristics of self-organizing systems). In addition, self-organizing systems are said to offer additional properties such as adapativity and robustness. For all these reasons it is tempting to transfer the ideas of self-organization to complex logistics systems. But it is often forgotten that self-organizing systems also have negative features. They show non-linear behavior. Causes and effects are not proportional. There are feedback loops which on the one hand enable self-organization (KRATZKY 1990), but on the other hand make such systems extremely hard to control.

Those negative features lead to deviations from the initial idea of decentral control. The system implementations are required to mitigate the negative features of self-organizing systems. Three different deviations can be distinguished.

Firstly, there is the class of distributed systems which we have already introduced above. They use a central mechanism to coordinate the actions of the decentral control components (see also TARAU 2010). Those coordination activities usually lead to an information exchange. By exchanging local information, global knowledge about the system is created. The coordination mechanism can only make a useful decision if it has global knowledge about all the individual decentral control components. The usage of distributed control systems does not fully explore the promising potential of decentralized systems. The coordination mechanism can be considered a single point of failure itself. Distributed systems would be assigned to the upper left corner in Figure 1. They generally have a distributed structure with slight global elements and aggregate local information to global knowledge.

Other systems without central coordination do not limit the actions of the system entities to local interactions. In many cases they are allowed to communicate with all other entities. This leads to an aggregation of local knowledge and shifts the systems from the upper right corner of Figure 1 to the upper left. Similar to the distributed systems, the potential advantages of decentral systems are not fully utilized when this approach is applied. Instead, an extreme communication overhead is created due to the system-wide information exchange. The required communication limits the applicability to complex networks.

Finally, the concept of distributed problem solving should be mentioned in this context (see BOND & GASSER 1988). Distributed problem solving names an approach where a complex problem is not solved by a single unit, but it is broken down into less complicated tasks which are solved by decentral components. The individual solutions are then aggregated to the global solution. Some systems also try to follow this approach which shows the biggest deviation from the decentral paradigm.

Due to the inconsistent usage of the term “decentral control” and the limited exploration of the decentral idea in existing systems, we follow a different approach. Our idea is to create a system which only uses local information and in which information aggregation is not allowed. Although we expect an efficiency loss, we want to quantify this loss and determine if it justifies the other advantages which decentral systems promise. Such a system does not have a hierarchical structure anymore. It can be called heterarchical. We want to compare the two extremes of a completely central and decentral system to answer the question whether the latter is a realistic approach or structural compromises (e.g. distributed systems) are always necessary. We define local information as information which is available at the point where the decision is made. The maximum radius from which a network node may acquire information are the paths which lead to its predecessors and its successors. No communication between the network nodes for exchanging information is allowed.

### 2.5.3 Control strategy types

After the discussion of the general functionality of control systems and their structural shape, the next step is to look at the rules that are used for decision making. Three different strategy types can be distinguished when looking at automated intralogistics transport systems:

- Dispatching
- Routing
- Intersection control

These strategies contain all other control aspects which have been mentioned in Chapter 2.4. Special requirements as battery management are not considered for simplicity. We only give a brief introduction to each strategy and state relevant assumptions. The strategies will be explained in detail during the literature review.

**Dispatching**
This control strategy type deals with two different aspects. Firstly, it is responsible for the assignment of loads to vehicles (and vice versa). Secondly, it covers the control of idle vehicles when no loads have to be transported.

Instead of dispatching many publications use the term scheduling. Scheduling requires pre-arrival information (LE-ANH & DE KOSTER 2006). Once the transport demands are known, a planning process is initiated to find the optimal vehicle-load assignment. In addition, the vehicle routes can be planned to avoid collisions. However, scheduling is often not applicable in practice. On the one hand, pre-arrival information is in many cases not available and makes the planning process impossible. Additionally, the scheduling algorithms cannot cope with the stochasticity and complexity of load arrivals in most transport systems. Real-time decision making would not be possible due to continuous system changes and the complexity of the calculations (VAN DER MEER 2000). Dynamic scheduling approaches which use a rolling planning horizon have not been completely adopted to intralogistics applications yet and therefore heuristic dispatching rules are the best choice in dynamic environments (LE-ANH 2005). Due to these limitations and as the availability of pre-arrival information is highly improbable in many transport systems, we limit our analysis to dispatching in online systems. In online systems transport information is not available before a load physically enters the system (LE-ANH 2005).

**Routing**
Once a load is assigned to a vehicle (or vice versa) or it has been decided to send a vehicle to a storage location for parking, the routing function starts its work. It is required to define the optimal route from source to sink. “Optimal” may refer to the shortest route or other criteria as the shortest route is not necessarily the fastest. Therefore adaptive routing algorithms need to take additional factors (traffic etc.) into account. The importance of routing depends on the transport network and the number of alternative routes it offers.

**Intersection control**
This strategy type refers to the control of switches and merges. At each switch it has to be decided for arriving vehicles which of the available directions to take. This aspect is normally already covered by the routing strategy. Additional control strategies for distributing the system’s load and preventing deadlocks might become necessary in decentrally controlled transport systems.
Merges are responsible for giving priority to one of the two directions when flows are combined. The applicable rules are rather simple. For example a “first-come first-served” (FCFS) logic can be used for controlling the right of way.

### 2.5.4 Robustness and adaptivity

This sub-chapter is concluded by a brief discussion of adaptivity and robustness. Both terms are named in many publications as the major qualitative advantages of decentral control systems. But their exact meaning remains unclear in most cases. This problem seems to arise from the fact that the terms are on the one hand frequently used in everyday language, and on the other hand to describe phenomena in many different scientific disciplines, e.g. informatics, control theory and medical sciences.

During this study we will analyze how disturbances influence the performance of transport systems. In order to give the reader a precise understanding of what we mean when talking about robustness and adaptivity in this context, we provide a definition of both terms. Following GRIBBLE (2001, page 1) and the theory of complex systems, we define:

- *Robustness* as "the ability of a system to continue to operate correctly across a wide range of operational conditions, and to fail gracefully outside of that range".

According to AY (2006) robustness can be achieved by adaptive mechanisms that compensate changes of the environmental conditions and failures of parts of the system. Therefore we define:

- *Adaptivity* as the ability of a system to adapt to changing influential conditions.

This means that robustness refers to the quality of the system performance. It judges the outcome and evaluates whether expected system characteristics have been achieved. Robustness is very hard to measure. On the one hand, it is extremely problem-dependent. On the other hand, the specific user expectations determine if a system can be called robust (see SASTRY & BODSON 2011). Adaptivity refers more to the functionality and the mechanisms for adaptation than to the outcome quality. The adaptvitiy of a system can be evaluated based on its robustness.

We believe that those definitions are an appropriate background for our discussion of disturbances. However, it should be noted that control theory (as many other scientific disciplines) has a slightly different understanding of the terms adaptive control and robust control. The interested reader may refer to the relevant literature (e.g. IOANNOU & SUN 1995).

## 2.6 Measuring the performance of a transport system

The preceding sections have introduced the control of intralogistics transport systems as the core topic of this thesis. Now, an appropriate methodology for comparing control strategies and system configurations needs to be derived.

### 2.6.1 Performance criteria

In order to properly assess different system configurations a number of key performance indicators (KPI) have to be defined. These have to highly aggregate the system data in order to limit the complexity of the analysis and to allow a quick judgment about the performance of the system. The KPI should enable a fast understanding of problems which might exist in the system without physically observing the system operation. This is especially important in this thesis as many different system configurations are compared.

Numerous criteria can be applied to measure the performance of a transport system. Some examples are:

- Travel time
- Waiting time
- Conformance to due time
- Vehicle utilization
- Load throughput

All of those criteria can be measured with respect to their maximum, minimum or average. For some applications also the underlying distributions of the performance criteria might be relevant. Quantiles can be used to characterize them.

The desired performance does always depend on the characteristics of the system which is analyzed. While it might be necessary to minimize the waiting times of arriving loads in one system, another application might make maximization of vehicle utilization desirable. In baggage handling systems for example, it is important that the maximum transport times does not exceed a certain maximum because a departing flight might be missed otherwise (HALLENBORG 2007). Quantile values are often used to define the performance requirements in this case.

We have defined dispatching as the major focus of this study. The specific goal set of this control strategy can be interpreted as controlling the system vehicles in a way that:

- Maximizes the throughput of the system
- Minimizes the time that each transport order spends in the system
- Minimizes the vehicle empty travel time

In our analysis we will only compare system configurations that achieve the same throughput. Thereby we can exclude this aspect from performance comparisons. In addition, we try to minimize the number of KPI for assessing the second and third aspect of dispatching.

We follow the definition of VAN DER MEER (2000) and consider the average service time as the key criterion. The service time equals the load waiting time (from system entrance to pickup) plus its transfer time, i.e. the time between pickup and delivery. The service time contains the average load waiting time which is a commonly used performance criteria. We consider the overall time that a load spends in the system to be the most important performance indicator and therefore want to simultaneously analyze the transport time. Minimizing the waiting time implies short queue length and in the case of constant laden transport times also low vehicle empty travel time. The latter aspect is becoming more and more important as it influences energy consumption.

For some further analysis, we will take a detailed look at the utilization of the vehicles. While it is quite simple to define the delivery status of a vehicle, it is very hard to exactly distinguish which share of its activities it spends on retrieving and going to park. The exact definition of both activities depends on the used dispatching strategy. Therefore a meaningful comparison is difficult. Whenever we consider vehicle activities, we will therefore limit our attention to the time that vehicles physically spend at a storage location. This is the only figure which can be defined similarly for all dispatching strategies.

In addition to the achieved service time, we consider the required number of vehicles and the distance they need to travel. Those two factors influence the required investment and the operating costs of the system. They can be interpreted as indicators of the system efficiency.

While some of the mentioned KPI can simply be read from the system specifications, others will need to be measured based on the operation of the transport system. The following sections illustrate different approaches to measuring the performance of a transport system and derive the usage of a simulation model as the only applicable approach for our research.

### 2.6.2 Approaches for analyzing material handling systems

There are various approaches which can be used when assessing the performance of a material handling control system. According to the specific situation, the best approach needs to be selected (DANILUK & CHISU 2010, LAW 2006). The figure below shows the usage of the real-world system as the first option for analyses. This is of course a very intuitive approach and promises the most realistic results. But there are also disadvantages. First of all, this approach is expensive. It will happen very

rarely that a company can afford to do real-world tests of a system. In addition, tests might contain risks, e.g. collisions of the vehicles. Depending on the length of the test run it might also be questioned if the considered state of the system is really representative and covers all questions which need to be answered before implementing the control systems. Especially when a new system is built, the real-world systems might not even be built at the time when the tests for the control system have to be completed. Different systems configurations should be tested before finalizing the design.

For all those reasons, the common approach is to use a model of the system for analysis and testing purposes. The term "model" can be used with two different meanings. The model can either be a physical model, e.g. a miniaturization of the real-world system. This kind of model is used in many engineering disciplines for cost-efficient analyses. On the other hand, mathematical models can be used. This is the type of model which is used in most cases for the development of control systems in material handling. A mathematical model represents a real-world system. A set of logic and quantitative relationships is used to build the model. Afterwards the input variables of the model can be modified to analyze how the system reacts to those changes (LAW 2006).

This study analyzes a generic transport system which does not exist in reality. It is therefore impossible to use a real-world model. The test environment needs to be modeled. We chose a mathematical model as it is more flexible than a physical model and allows us to assess various different aspects. In addition, the mathematical model is less expensive and can be created with less effort. As the applicability of analytical models for the given case is limited, simulation was chosen as a tool for all experiments. The following sub-chapters review both approaches with their respective advantages and disadvantages.

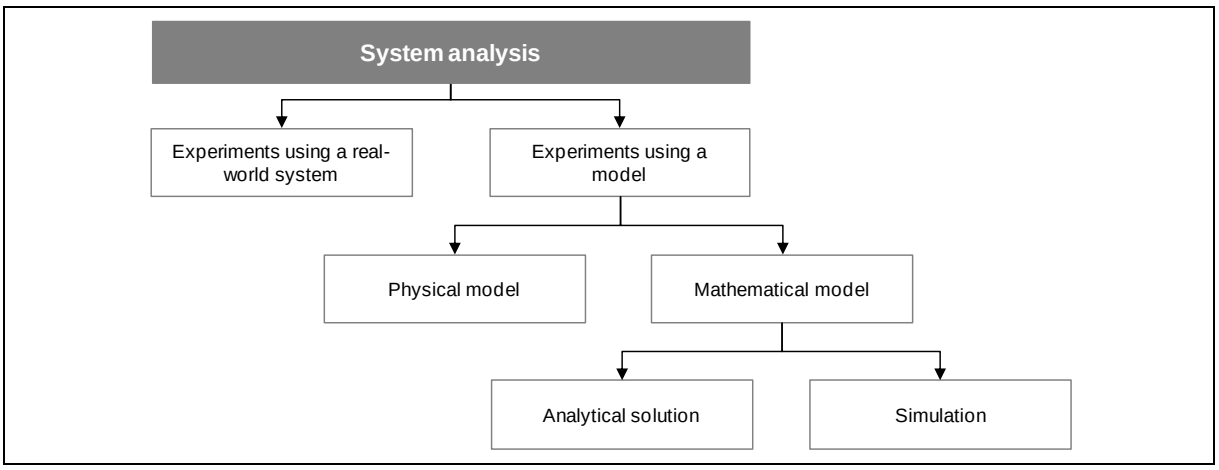



Figure 11: Methods for system analysis (compare DANILUK & CHISU 2010)

### 2.6.3 Analytical solutions

Analytical models are mathematical models which allow for finding an analytical solution. This means an exact, optimal result can be calculated based on the relationships which the model contains. Only models with a comparably simple structure can be solved analytically. And even with a simple structure, computing the optimal solution might become very time-consuming (LAW 2006).

Many of the commonly known planning problems in logistics and operations research can nowadays be solved with analytical methods (e.g. the vehicle routing problem). If the computational effort is too high, heuristic approaches have been developed to achieve goods results with a reasonable cost/benefit ratio (see e.g. NEUMANN & MORLOCK (2002) for an introduction to linear programming and other techniques).

The main advantage of the analytical models is that they allow universal statements about a system (SCHOLZ-REITER ET AL. 2008). However, optimization techniques like linear programming are not able to take the stochasticity and dynamics of real-world logistics systems into account. They are rather limited to planning problems with a longer time-horizon and need stable, deterministic input data to be applied.

SCHOLZ-REITER ET AL. (2008) have therefore reviewed some additional modeling techniques for usage in dynamic logistics environments. They distinguish event-driven and flow-oriented methodologies. Most logistics systems are modeled with event-driven techniques as their behavior is based on distinct actions of logistics entities. In contrast, flow-oriented approaches use differential equations to model logistics systems. Although the resulting mathematical model can be solved rather quickly, the use of continuous flows is not intuitive for most logistics systems. A general assessment of the different modeling methodologies is summarized below:

- Event-driven techniques
  - Petri-Nets
    Petri-Nets are a graphical language for system modeling which uses a supplemented mathematical theory. They can map causal links and concurrency. Petri-Nets can be used to analytically prove certain functional system properties (e.g. that the system is free of deadlocks). There exist several extensions of the concept which have been developed to improve the value of applying Petri-Nets.
  - Queuing theory
    This technique can be used to analytically evaluate a certain class of stochastic systems. It puts the concept of queues in the centre of all thoughts and enables the calculation of several performance indicators for the logistics entities which are served in the queue network. Thereby, for example mean and variance of waiting times in queues can be calculated. The disadvantage is that the analyzed systems need to fulfill a lot of structural requirements to make the mathematical models of queuing theory applicable. The analyzed systems in many cases do not represent real-world applications. Additionally, the theory cannot predict the future state of the system.
  - Max-Plus Algebra
    In contrast to queuing theory, the Max-Plus Algebra considers deterministic systems. Using an own algebra with a “Max” and a “Plus” complexor, it can model recursive dependencies in systems and derive mathematic structures which are very similar to linear equation systems. The applicability of the methods from linear algebra is the biggest advantage. Max-Plus Algebra enables the analysis of dynamic system properties and interrelations.
- Flow-oriented techniques
  - Control Engineering
    This approach introduces the idea of feedback-based control into the modeling of logistics systems. The transfer of these ideas to the logistics domain is complicated and limits their applicability. However, the usage is seen as a promising field of future research and can help to analyze temporal changes to the system.
  - Dynamic systems
    The ideas of dynamic systems have mainly been developed in the field of mathematics. The main focus is system properties as stability and the general description of dynamic systems. The analysis of output data is not the major goal.

The figure below summarizes the different approaches and states some analyses they can be used for. None of the approaches which have been discussed so far allow an extensive evaluation of the system characteristics with respect to the typical logistics KPI which we have introduced above. Either the analytical approaches cannot deal with the structural complexity and the dynamics of the material handling systems or they are not able to measure those indicators at all. For some of them other restrictions apply. Therefore it is inevitable to introduce additional techniques. The figure below already contains simulation as an alternative approach for systems analysis. The next section will therefore review the opportunities that a simulation approach offers. It should be noted that we refer to simulation as an approach for modeling and controlling a system. In addition, there are simulation approaches for some of the analytical techniques. Simulation in this context refers to using simulation models for solving the analytical problem, not to modeling the system itself (NEUMANN & MORLOCK (2002) refer to this as deterministic simulation).

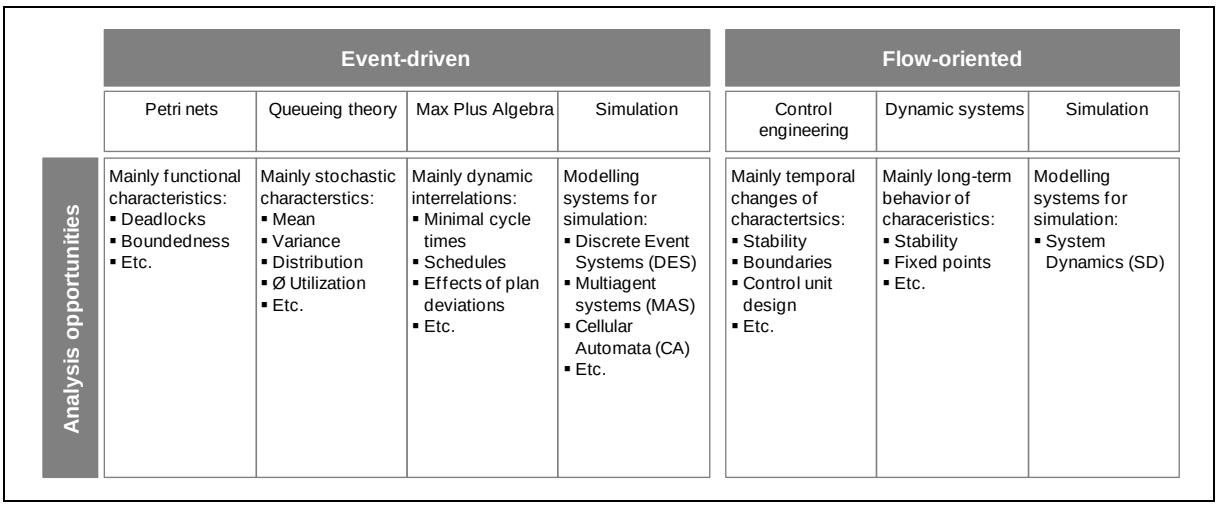



Figure 12: Mathematical approaches for system analysis (compare SCHOLZ-REITER ET AL. 2008)

### 2.6.4 Simulation

Simulation uses computers to numerically evaluate the mathematical model of a certain system and the generated output data to estimate the true model characteristics (LAW 2006).

Following the classification of SCHOLZ-REITER ET AL. (2008) discrete and continuous simulation models can be distinguished (see figure above). The former are used to represent systems that change their status at certain points in time, while the later are used for systems where the state variables change continuously. For an internal carrier-based or vehicle-based transport system a stochastic discrete event simulation model is most suitable. Stochastic means in this context that the models contain random input variables.

As mentioned above, simulation models have the disadvantage of not allowing the derivation of universal statements about a system. The results of simulations are only valid for the system configuration which has been analyzed. However, a simulation study is the only feasible approach for our research objectives due to the following reasons (also compare what has been said about analytical models above):

- Analytical models are not able to reflect the dynamics of intralogistics operations and should be used in systems where a long and stable planning horizon is considered (SCHOLZ-REITER ET AL. 2008). We consider dispatching systems without a planning horizon. The resulting dynamics can only be modeled with a simulation model.

- Analytical models can only be applied efficiently to small systems with few vehicles. Otherwise the computational effort becomes too high (QUI ET AL. 2002).
- Analytical and related modeling techniques are not able to support an analysis regarding the KPI which we have defined.
- Simulation allows experiments and comparative analysis of different control systems before the actual system is built or only with a virtual system.

Simulation models are an appealing approach to system analysis during planning, implementation and operation of a control system. However, care needs to be taken when creating simulation models. The effort needed for building a valid model should not be underestimated. The results which the model creates can only deliver reasonable results if the model is a good representative of the real-world system. Additionally, the quality of the input data is of extreme importance. The results can only be as good as the quality and validity of the input data (DANILUK & CHISU 2010). Certain statistical procedures need to be followed when carrying out simulation studies. The results are only estimations of the system characteristics.

Summarizing, this thesis uses a stochastic discrete event simulation model to compare the performance of central and decentral dispatching strategies in intralogistics transport systems. A series of simulation experiments will be necessary for analyzing the output data with respect to the defined KPI.

# 3 Literature review

Having introduced the main scope of this thesis, this section reviews the existing literature. The three control strategy types which were mentioned above are used as a framework. As already argued, it will become evident that the routing strategies are the most frequently mentioned concept when it comes to decentral control. Each control strategy type is analyzed according to two dimensions. Firstly, its broader meaning and the theoretical background are explored. Afterwards proposed decentral implementations will be assessed regarding the three remaining shortcomings of decentral control systems:

- Violations of the decentral paradigm
- Limited system size due to technical limitations
- Insufficient performance measurement

Following the dispatching strategies, routing and intersection control rules are discussed.

## 3.1 Dispatching

Most publications about dispatching strategies deal with AGV systems. The major research activities on this topic started in the 1980s and many of the fundamental classifications which are still valid have been developed during the last two decades of the 20th century (see e.g. MAXWELL & MUCKSTADT 1982). Starting with manufacturing systems and warehouses, attention has especially turned to port container terminals during the last few years. Another important area of research is the control of overhead hoist systems in semiconductor wafer fabrication (see e.g. KOO ET AL. 2005). While there are also publications that try to use optimization techniques for dispatching, we will limit our attention to those which work without pre-arrival information and can be used in extremely dynamic environments. The term dispatching is also used in other contexts, e.g. in the control of power systems or production planning. The meaning of the concept is different in those contexts and confusion should be avoided.

As mentioned above (see Chapter 2.5.3), we will refer to dispatching in two dimensions. The first is assigning vehicles which have just completed a job to new loads and assigning arriving loads to vehicles. The second task is to decide about a vehicles' destination when it just completed a job and no loads are available for assignment. The AGV literature distinguishes both questions. Only the first one is referred to as dispatching while the second is usually named vehicle positioning. Generally, we feel that the definition of dispatching should include both activities as both deal with the question of further usage of an empty vehicle and need to be considered subsequently in a real-world implementation. Our understanding of dispatching does therefore cover the load-vehicle assignment and empty vehicle positioning.

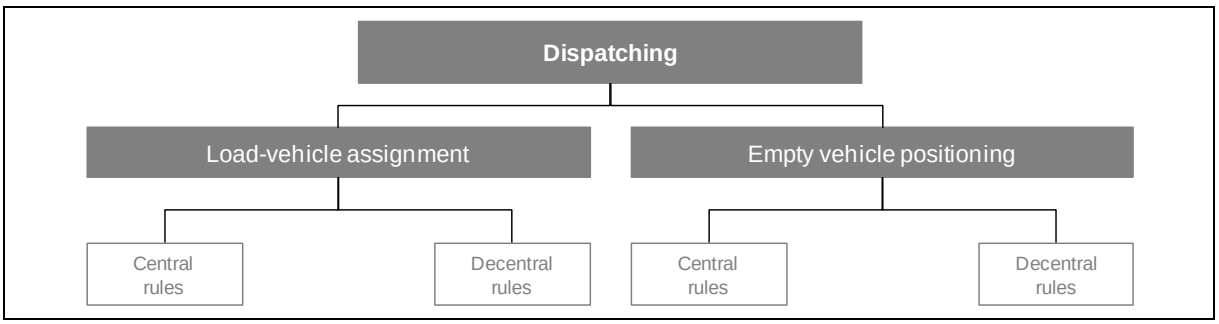



Figure 13: Classification of dispatching tasks

**Load-vehicle assignment**

Load-vehicle assignment is not only limited to situations when a vehicle delivers a load. In total there are three different actions which can trigger a dispatching process (see LE-ANH 2005):

- Arrival of a new load in the system
- A vehicle delivers a load to its destination
- A vehicle reaches its parking location

These triggers reflect the two perspectives which EGBELU & TANCHOCO (1984) define for classifying dispatching rules. The first trigger is linked to load-initiated (or workstation-initiated) rules. This means that an arriving load is responsible for selecting a vehicle. If there are idle vehicles, they are prioritized according to a certain logic and the "best" vehicle is claimed by the load. The logic for prioritizing the vehicles is essentially the dispatching rule. Several different criteria can be applied for ranking available vehicles (see the description of the dispatching rules below).

The last two triggers on the list refer to the second perspective on dispatching rules. They require vehicle-initiated dispatching rules (see also KOO & JANG (2002) for a comparison of the two rule types). Idle vehicles apply a certain logic for choosing one of the waiting loads.

In real-world systems load-initiated rules are usually combined with vehicle-initiated rules. This is necessary to prevent a system stopping operation. When for example an arriving load cannot find an idle vehicle once it has entered the system, the simplest approach is to let the load be claimed by the first vehicle which becomes idle. If this combination of both rule types was not implemented, the load would have to continuously or at least periodically search for vehicles, slowing down the system response time and performance.

Before we give an overview of central and decentral dispatching rules, it is necessary to take a look at the kind of system models which the AGV community deals with. Only the knowledge of those

models can enable a clear understanding of what central and decentral control dispatching strategies really are in this context. LE-ANH (2005) distinguishes three different AGV system types:

- Conventional systems
- Single-loop systems
- Tandem systems

While the first system type can refer to virtually any real-world transport system, the latter two form problem type classes which are commonly accepted in the AGV community. Single-loop systems consist of only one guide path loop with several loading and unloading stations. Several vehicles travel in this loop which leads to simple traffic control and dispatching requirements. The congestion probability is low (see LE-ANH 2005 for a comparison of the system characteristics). Tandem systems are single loop systems which are connected by transfer points. Only one vehicle is used in each of the loops. This makes traffic control and dispatching simple. There is no congestion risk as only one vehicle is used in each of the loops. Compared to those two system setups, the dispatching and traffic control requirements in conventional systems are far more complex. Multiple vehicles have to be controlled and there is a high probability of congestion.

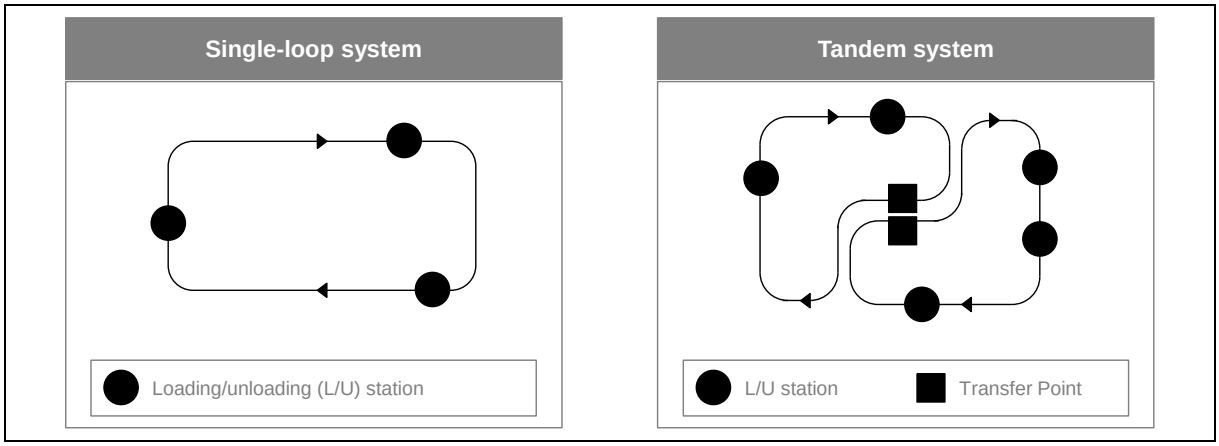



Figure 14: Standard AGV layout types

Having the different topology types in mind, we can now turn to central and decentral dispatching rules. VAN DER MEER (2000) describes a centralized control system as a knowledge-based system which continuously monitors the systems status. At every point in time it knows about all the loads which are waiting at loading stations. It is aware of the position and status of all vehicles in the systems. In order to acquire this knowledge, the central controller has to communicate with all source, sinks, storage locations and vehicles in the systems.

Numerous central dispatching methods which rely on global information for assigning loads to vehicles have been developed. There is a whole range from simple to complex rules (see LE-ANH 2005, LE-ANH & DE KOSTER 2004). They are either load-initiated or vehicle-initiated:

- Single attribute dispatching rules:
  These rules only use one criterion for making the dispatching decision. Examples are distance-based rules. This class of dispatching rules comprises some of the simplest and best-known rules, e.g. the shortest-travel-time-first (STTF) or shortest-travel-distance-first rule (STDF). Other subclasses of the single attribute dispatching rules are workload-based and time-based dispatching.

- Multi-attribute dispatching rules:
  These rules combine several different factors for making dispatching decisions and might achieve better results than the single attribute dispatching rules.
- Hierarchical dispatching rules:
  They are often used in manufacturing systems and use the added-value as a decision criterion.
- Rules which use vehicle reassignment or a look-ahead period:
  BOZER & YEN (1996) introduce a new idea into the dispatching research community. They do not limit the vehicle-load assignment to a nonrecurring event. Using their modified shortest-travel-time-first (MOD STTF) and bidding-based device dispatching ($B^2D^2$) it becomes possible to reassign vehicles according to a certain set of rules. MANTEL & LANDEWEERD (1995) propose "thinking-ahead" and include information about loads which will be available shortly in the dispatching decision. They distinguish between rules with and without a time horizon.

All of these dispatching rules are heuristics. They do not calculate a global optimum, but rather try to make a decision in a short time. As already mentioned, all rules require global knowledge as their assignment philosophy requires a global comparison of different vehicle-load assignment options in the system. DE KOSTER ET AL. (2004) perform a study to compare several central dispatching rules with and without pre-arrival information in three real-world scenarios. They consider a distribution center, a production plant and a transshipment terminal with 5, 11 and 25 vehicles in those systems. The tested distance-based dispatching outperforms the other tested rules with respect to the average load waiting time. Their performance might be worse regarding the maximum waiting time. The availability of pre-arrival information can significantly improve the dispatching performance. The same authors state in 2005 that there is not one universal dispatching rule that performs best for all cases. The best rule for a specific case rather has to be found by a detailed analysis (LE-ANH & DE KOSTER 2005). CLAUSEN ET AL. (2011) choose such an approach and analyze the consequences of using different vehicle-initiated and load-initiated dispatching rules in combination with different layouts of a transshipment terminal. Their conclusions focus on the best layout shape rather than on the comparison of the dispatching performance. The number of forklifts which are part of their vehicle fleet is not explicitly stated.

Having examined the most important central dispatching approaches, we will now turn our attention to decentral dispatching. Decentral dispatching is discussed less frequently than central dispatching. Additionally, the literature discusses decentral dispatching usually in very simple systems. Mostly, single-loop systems or tandem systems are considered. Those are comparably easy to control. In some cases even analytical models can be applied for optimization (VAN DER MEER 2000).

BARTHOLDI & PLATZMAN (1989) analyze a single-loop system and evaluated the performance of a first-encountered-first-served rule for this scenario. They were able to derive an expected performance level analytically and verify it with a simulation. Inspired by these results several authors tried to develop new approaches during the following years. The aim is to develop extremely simple layouts or to transform conventional layouts into combinations of simple, i.e. single-loop layouts. The resulting layouts should achieve the same performance as the conventional layout but be far easier to control. SINRIECH & TANCHOCO (1993) develop methods which can help to construct a single loop system for a certain given environment or cut an existing single loop into segments which are then each served by only one bidirectional vehicle (SINRIECH ET AL. 1996). Although the authors do not explicitly state this, they use some kind of sequential dispatching which is similar to the first-encountered-first-served rule. BOZER & SRINIVASAN (1992) and ROSS ET AL. (1996) also follow a simplification approach when they introduce their idea of tandem layouts. Although all those authors spend a lot of effort on

performance measurement and comparing their systems to the conventional version, it needs to be mentioned that the considered conventional layouts have a very simple structure themselves. As only a few vehicles are used in the system, these ideas provide limited benefit for the systems which are in the scope of this study. We want to focus on complex conventional systems with many vehicles.

Besides the mentioned decentral approaches which have their roots in the AGV literature, the concept of agents and multiagent systems started to influence the control literature in the late 1990s. Several authors tried to develop agent-based systems in order to exploit the advantages of the decentral paradigm.

In 2002 BERMAN & ERDAN proposed the usage of decentralized control systems in a computer-integrated manufacturing environment. They favor the usage of vehicle-initiated dispatching rules compared to workstation-initiated rules as they judge the latter to have higher requirements regarding the communication overhead. The concentration on vehicle-initiated rules is considered to be sufficient as in the manufacturing context the vehicles are highly utilized and the system is rarely in an idle state. They implement a multi-attribute dispatching rule based on the distance to the workstation and the due time of the product. The vehicles collect information from all workstations before making a dispatching decision. It becomes obvious, that this approach does not fulfill the requirement of using local information only. In combination with routing rules, a conceptual test environment with up to two vehicles was developed.

FAY & FISCHER (2004) develop a control system for DCV in a baggage handling application. They propose a multi-agent dispatching system based on the contract-net protocol. The vehicles make offers for each new load. Those offers are based on the distance between the current vehicle position and the pick-up location. The distance figure is combined with the current utilization of the route to prioritize and select one of the offers. But by using the distance as a decision criterion this strategy uses global information. FAY & FISCHER carry out a simulation study which uses a section of a real-world baggage handling system and historical input data. However, the experimental simulation settings remain unclear and the system does not contain more than 15 vehicles.

The control methodology developed by WEYNS & HOLVOET in 2008 also makes use of the multi-agent perspective. The aim of the authors is to test the feasibility of decentral control systems for AGV. They refer to dispatching as “transport assignment” and develop two different rules which they call FiTA and DynCNET. The first approach is based on the idea that transport agents (of loads which are waiting to be picked up) and the AGV agents both emit fields in their local virtual environment. The vehicles combine the effects of the fields they receive to calculate some kind of field gradient which they follow. It should be noted that the authors state that this kind of system is currently not feasible as the engineering concepts are not available. In addition, they propose another algorithm which is an adaption of the contract net protocol. Transport agents and AGV agents both have a certain radius in which they search for vehicles or loads. The assignment procedure works similarly to the contract net protocol. It is not clearly stated which criteria the vehicles use to provide their proposals. Compared to the standard contract net protocol only one modification has been made. The vehicles are allowed to switch tasks after the initial assignment. WEYNS & HOLVOET (2008) use simulation experiments to analyze the performance of their dispatching rules. In a real-world layout with 56 loading and 50 unloading stations, they use 14 AGV to show that their 2 new dispatching rules perform better than the original contract net protocol regarding the average waiting times.

SEOW ET AL. (2010) provide a study which deals with decentral dispatching in a multi-agent system, but is not related to internal transport systems. They use a taxi operator as a case study to develop a decentral dispatching system. The taxi case is very similar to the questions which internal vehicle-

based systems have to answer. A simulation model with up to 20 taxis is used for comparing a central with a decentral approach based on customer waiting time and empty travel time. The conceptual model for this system contains communication among all taxis and a central dispatch center which bundles the incoming customer requests. Therefore global information and a central coordination mechanism are in place.

**Empty vehicle positioning**

When it comes to the second aspect of dispatching – the control of empty vehicles or "vehicle positioning" – literature becomes even less extensive. The simple question to answer is where to park an idle vehicle in case no new loads are available for assignment once a vehicle has completed its current job. The major objective of vehicle positioning is to find parking locations which help to minimize the empty vehicle travel time and the response time of the system once new loads are available for pickup. Idle vehicles are parked to avoid increasing the overall traffic and influencing the transport of loads. Empty vehicle positioning is of high relevance for the overall system performance. Its significance grows with the load profile variability. There are several approaches which are proposed in AGV literature (LE-ANH 2005):

- Central zone positioning rule: There is one central zone in the layout where vehicles are sent when they are idle.
- Distributed positioning rule: This rule is generally similar to the central zone positioning rule. But instead of one central zone, there are multiple zones in the system. The vehicles are distributed to these zones.
- Circulatory loop positioning: One or more loops in the system are designed for empty vehicle circulation.
- Point of release positioning: This rule is the simplest approach. It means that idle vehicles stay where they have completed their last job.

The first approach is only an option in very simple layouts where the loading stations are located close to the central parking zone. Otherwise the response times would be considerably long. Parking at the point of release will in many cases lead to congestion, if no additional parking zones are integrated in the layout. As already mentioned, instead of assuming circulatory loop positioning, this study follows the distributed positioning approach. The future demands in the systems are by definition unknown. Therefore the best strategy is to balance the number of vehicles at the different storage locations in the system.

As stated above, it is extremely difficult to distinguish when an empty vehicle is going to park and when it is moving to retrieve. Both terms depend on the used dispatching strategy. In a similar fashion it is hard to define exactly when a vehicle is empty and when it is idle. An empty vehicle that has just completed a delivery job might directly be sent to a source for picking up a new load. In this case it is empty and leaves to retrieve the assigned load. Thus it is empty, but not idle. It would only be idle if it was sent to a storage location. The situation gets more complicated if a dispatching decision does not explicitly send a vehicle to a source or storage location but the decision is either postponed or depends on random influences. Due to this ambiguity which is discussed in detail in Chapter 5.2 for each of the introduced dispatching strategies, we use the terms "empty" and "idle" synonymously for all vehicles that are currently not transporting a load.

Similar to what has been said above about dispatching, most of the AGV literature only considers very simple problems when dealing with empty vehicle positioning. In many cases single-loop layouts are considered. Sometimes systems with only one vehicle are analyzed. Generally, it is assumed that there are loop sidings in the systems for parking vehicles. Using either the maximum vehicle response time,

the mean vehicle response time or an even vehicle distribution as optimization objective, algorithms for selecting home locations are developed (see e.g. CHANG & EGBELU 1996, KIM & KIM 1997). The aim is to develop mathematical models for solving the positioning problem analytically. Therefore very restrictive system characteristics have to be assumed. GADEMANN & VAN DE VELDE (2000) analyze a static version of the problem. This means that they examine systems in which all vehicles are always idle at the same time. Only HU & ENGEBELU (2000) consider a conventional layout and try to make less restrictive assumptions about the demand patterns. However, their integer programming model is not easily applicable to real-world cases.

Using analytical models to solve the problem of empty vehicle positioning will always require global information and very limited structural layout complexity. In many cases pre-arrival information about the demand patterns are necessary. These restrictions do not fit to the characteristics of systems which are to be analyzed in this thesis. But especially for more complex systems with a large amount of vehicles much theoretical background does not exist. Two publications on the topic are summarized below.

In 2004 LE-ANH & DE KOSTER made an attempt to analyze dispatching rules for systems with many vehicles. They do not explicitly refer to the term "vehicle positioning" in this context, but rather include it in their overall dispatching process. Besides three of their own dispatching rules, they include the best rules which TALBOT (2003) found in his research. The dispatching rules are compared in two rather simple layouts. One of the layouts contains 2 loading and 2 unloading stations. The other layout contains 4 stations of each type. Loading and unloading stations are situated next to each other. The number of switches in the system equals the number of loading stations. Each switch is located in front of one of those stations. Between 60 and 100 vehicles are used in total. The main assumption for the study is that only very few vehicles can be parked in the system and that idle vehicles generally have to circulate until they find a new task. Using different balanced and unbalanced load scenarios the following dispatching rules are compared:

- Modified shortest-travel-distance-first rule
  In contrast to the original version of the rule, its modified version allows reassignment and cancelation. This means that even after a vehicle has found a closest load and is approaching it, it may stop and pick up a load which it encounters waiting at any location along its route. The formally assigned job needs to be canceled.

- Entrance control (EC) dispatching rule (TALBOT 2003)
  This rule is based on the idea of calculating a net-stock level as a decision criterion. The net stock level $s_i^{net}$ is calculated for each source station $i$ at time $t$ as follows:

$$s_i^{net}(t) = v_i^{curr}(t) + v_i^{app}(t) - w_i(t) \tag{3.1}$$

  In this formula $v_i^{curr}$ is the number of vehicles which are currently at the station $i$, $v_i^{app}$ is the number of vehicles which are currently approaching station $i$ and $w_i$ is the number of loads which are waiting to be picked up at this station. When a vehicle arrives at the decision point (switch) which is corresponding to station $i$, the net stock level at station $i$ is calculated. If the net stock level is currently below a certain threshold value $s_i^{thr}$ the vehicle proceeds to station $i$. If it is above the threshold, it is routed to the other direction. Having delivered a load at a certain sink, the idle vehicle circulates in the system until the described procedures lead it to a loading station.

- Multi-attribute dispatching rule (Multi-Att)
  The multi-attribute rule combines the net-stock parameter $s_i^{net}$ with the distance $d_{vi}$ from the current location of vehicle $v$ to the station $i$ by using the weight factors $\beta$ and $\gamma$, where $\beta + \gamma = 1$:

$$\omega_{vi} = \beta * \hat{d}_{vi} + \gamma * \hat{s}_i^{net} \tag{3.2}$$

  But LE-ANH & DE KOSTER do not simply adopt the net-stock level from the EC rule. Both values $d_{vi}$ and $s_i^{net}$ are normalized by taking the minimum and maximum values of stock level and distance for all stations $i$ into account.

$$\hat{s}_i^{net} = \frac{s_i^{net}(t) - min_i s_i^{net}(t)}{max_i s_i^{net}(t) - min_i s_i^{net}(t)} \; ; \quad \hat{d}_{vi} = \frac{d_{vi}(t) - min_i d_{vi}(t)}{max_i d_{vi}(t) - min_i d_{vi}(t)} \tag{3.3}$$

  At a decision point the specific $\omega_{vi}$ for all loading stations is calculated. The station with the smallest $\omega_{vi}$ is selected. At the next decision point a new calculation is carried out and the destination assignment is repeated. The circulation of idle vehicles works similarly to the EC rule.

- Modified multi-attribute dispatching rule (Multi-Mod)
  This rule works similarly to the Multi-Att rule. The only difference is that a power coefficient $\delta$ is included in the formula for calculating the decision criterion:

$$\omega_{vi} = \beta * \hat{d}_{vi} + \gamma * \left(\hat{s}_i^{net}\right)^{\delta} \tag{3.4}$$

The results from the simulation experiments show that the modified version of the shortest-travel-distance-first rule does not perform well in this type of system. It is outperformed by the other rules. Compared to the last two rules, the EC rule has the disadvantage of being dependent on choosing the threshold value $s_i^{thr}$. Additionally, the last two rules are less sensitive to different load scenarios, i.e. load arrival rates and load arrival patterns. From the perspective of our study, it needs to be said that the first and the last two rules all use global information, either based on distances or because of using minima and maxima across the whole system. Talbot's rule is the one which needs the smallest amount of information and comes close to our understanding of decentral control. But it requires the definition of the threshold value and is outperformed by the global approaches. Generally, the layout needs to fulfill certain requirements regarding the position of the decision points in order to enable the usage of those rules. The circulation of idle vehicles is a prerequisite for their applicability.

HALLENBORG (2007) provides the only other discussion regarding empty vehicle control which we could find in literature. He considers a real-world baggage handling system which uses plastic totes and develops an agent-based control system. It is explained how simulation and emulation can be used but the paper does not contain any performance measurement or detailed discussion of simulation results. It is rather a conceptual description. The main idea for dispatching empty vehicles is to a certain extend similar to the approaches which were described in the last paragraph. The main difference is that no circulation of empty vehicles is assumed. Each vehicle is assigned a storage location once it has completed a job. It remains unclear if this is the only dispatching decision or if there is also some load-vehicle assignment involved. For dispatching the travel time to a specific storage location and the current fill level are considered. The relative fill level $s_i^{rel}$ of storage location $i$ is calculated as the ratio between the current number of totes at the storage location $v_i^{cur}$ and its

maximum capacity $s_i^{max}$. The fill level is then converted into a priority $z_i$. This priority is afterwards multiplied by the current travel time $d_{vi}$ of vehicle $v$ to storage location $i$ to calculate the decision criterion $\omega_{vi}$.

$$z_i = \begin{cases} 2{s_i^{rel}}^2 & 0 \leq s_i^{rel} < \frac{1}{2} \\ 1 - 2\left(1 - s_i^{rel}\right)^2 & \frac{1}{2} \leq s_i^{rel} \leq 1 \end{cases} \tag{3.5}$$

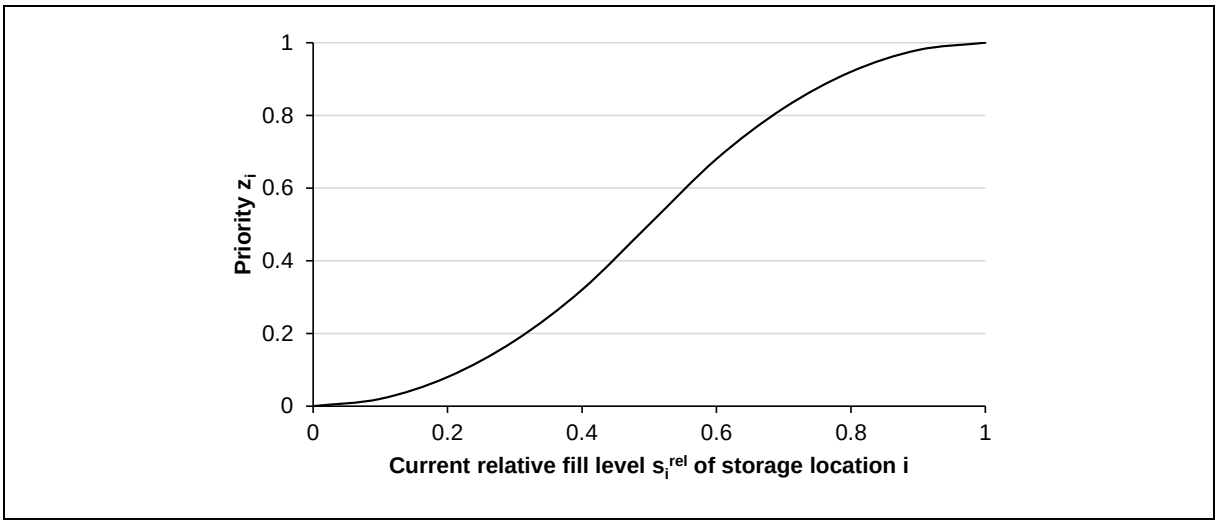



Figure 15: Function for calculation of priority *z*

The dischargers sends the empty tote to the storage location with the smallest value of $\omega$.

$$\omega_{vi} = z_i * d_{vi} \tag{3.6}$$

It is obvious that this comparison requires the usage of global information. The same is true for the determination of the dynamic travel time.

After the dispatching strategies, the next sub-chapter reviews the existing theoretical background for routing rules.

## 3.2 Routing

The term routing is used to describe control procedures which make vehicles in a transport system find a route from a source to a desired destination. Depending on the used routing protocol this procedure does not only involve finding a route, but also operationally guiding the vehicle from source to destination. If more than one route is available for a certain source-sink combination, the selection of one path for a certain transport is the second aspect of routing.

We start this section by introducing a general classification scheme for routing strategies which was developed by LAU & WOO (2008). The following paragraphs mainly show their findings. Afterwards we take a specific look at routing in AGV systems and BHS systems. The latter leads us to a comparison of routing in logistics and communication networks. Finally, several decentral routing approaches are analyzed.

The first and major distinction of routing protocols is based on the opportunity to update the best paths. Static routing only calculates the best paths once. Afterwards those paths are used without taking the current network status into account. The best paths can for example be computed by applying the Dijkstra algorithm (DIKJSTRA 1959) to all origin-destination pairs in the network model. The algorithm needs a criterion for determining the best path. Usually the shortest distance is used for this purpose. Besides calculating a complete routing table which contains the best route for each source-sink relationship, static routing can also be implemented by using a certain set of simple routing rules.

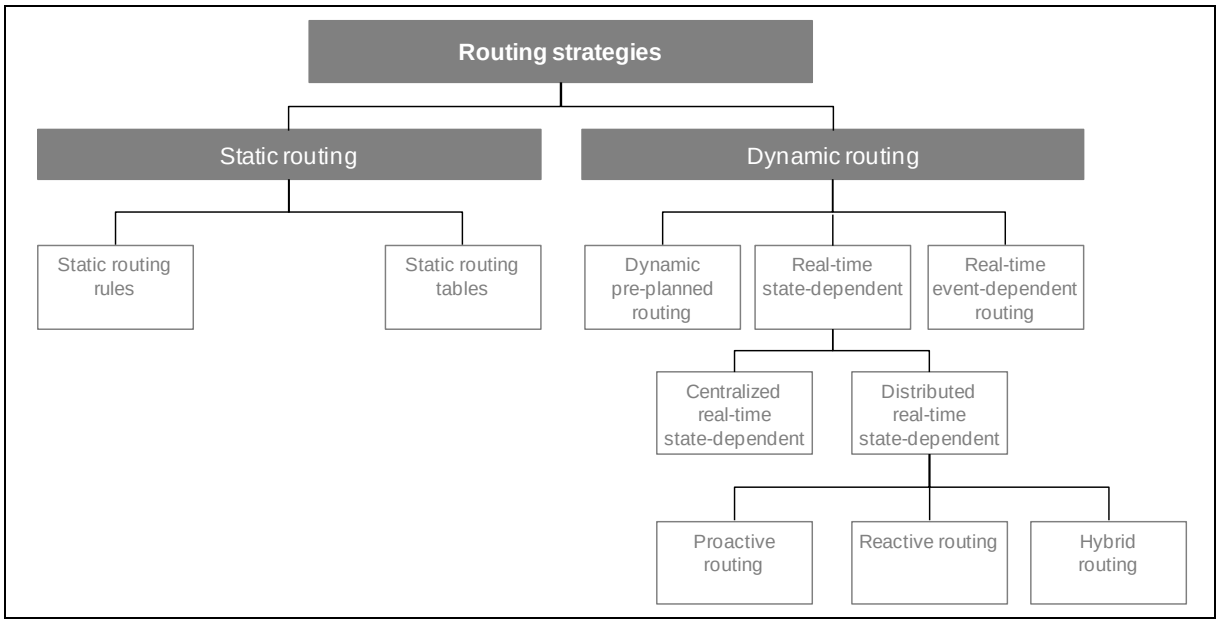



Figure 16: Classification scheme for routing strategies
(compare LAU & WOO (2008) and ABOLHASAN ET AL. (2004))

Dynamic routing strategies try to overcome the weakness of static strategies and take the current network status into account. The simplest approach is to define the time for routing table updates in advance. Depending on the traffic patterns or layout changes the calculation of the routing table is repeated. This method is referred to as dynamic pre-planned routing. It requires anticipation of the future changes to pre-plan the optimal time for updating the routing tables.

Real-time state-dependent and event-dependent approaches take the thought of adaptation to the current network status even a step further. Updates of the routing tables are not pre-planned but may happen continuously. Event-dependent routing relies on a set of alternative routes. Depending on the network status one of these routes is selected. An example strategy would always select the shortest path unless the utilization of this route is above a certain threshold or a path breaks down. In contrast, state-dependent routing uses information exchange to determine the currently best route based on the status of the network. A centralized version of state-dependent routing has global knowledge of the network and is extremely responsive. Distributed algorithms require a higher degree of information exchange, but are said to be more robust to disturbances (see LAU & WOO 2008). The routing tables are not kept by one central entity but locally by the different nodes in the system.

The distributed protocols are mainly used in mobile ad-hoc networks. Three different forms of distributed state-dependent routing can be distinguished (see ABOLHASAN ET AL. 2004):

- Global/Proactive routing uses periodic updates to continuously maintain an up-to-date view of the network. Examples are the optimized link state routing (OLSR) or the global state routing (GSR).
- On-demand/Reactive routing schemes do not proactively maintain a network understanding, but are executed only when a routing request is received. Examples of this algorithm type are adaptive distance vector routing (ADVR) and dynamic source routing (DSR).
- Hybrid routing schemes combine features of proactive and reactive protocols. The overall effort for route discovery is in most cases reduced by limiting the radius for the proactive network updates to near-by nodes and the usage of reactive routing for finding paths to remote nodes. Most of the hybrid protocols use zones to define the radius for the proactive updates.

**Routing vs. scheduling**

Publications on routing which are relevant for the topic of this study and were mentioned in the classification scheme above, mostly come from the field of logistics or communications networks. We will now look at intralogistics routing approaches and derive the reasons for considering communication protocols in this context.

The traditional AGV literature usually considers routing and scheduling concurrently as two of the main control tasks. But the scheduling term is not used especially consistently and the distinction to routing remains unclear. It is a general consensus that routing means to find a route from source to destination. Some authors have additional requirements and consider routing to be responsible for constructing routes which are free of congestion, conflicts and deadlocks (see e.g. QIU ET AL. 2002). Other authors (e.g. TAGHABONI & TANCHOCO 1995) see the latter requirements as a task of the scheduling procedure which is used. The confusion can on the one hand be explained by the technical characteristics of AGV. As we have seen above, depending on the used guidance technology, the avoidance of collisions and deadlocks might not be an easy task on the operational level. Therefore it seems reasonable to assign this task to the scheduling strategies. This is applicable in systems where all transportation requests are known before the system starts operation and optimization techniques can be applied to find the best routes which provide the highest system performance without interferences (deadlocks, collisions etc.). The problem at hand is then similar to a pick-up and delivery problem with time window (see LE-ANH 2005). But on the other hand, scheduling is in many systems not possible as no or only limited pre-arrival information is available. Dispatching strategies have to be used. In this case the task of avoiding collisions and deadlocks which is relevant in AGV systems would need to be accomplished by the routing strategy. Sticking to the requirements of providing conflict-free routes, several planning types have been developed which try to cope with this requirement in constructing feasible routes and assume that the routing activities are carried out after the dispatching decision has been made (see e.g. QIU ET AL. 2002):

- There are static methods which block entire paths for the time a vehicle is expected to travel on them. Using this methodology, conflict-free routes can be constructed.
- Time-window-based methods grant access to certain paths based on rolling planning horizons.
- Dynamic methods do not plan ahead but construct the route incrementally while the vehicle is travelling by considering the current utilization of a path when making the decision about which node a vehicle should travel to next.

The former two methods put more emphasis on the planning aspect while the latter one is a more reactive approach. The approaches which focus on planning can be applied in rather small layouts with few vehicles and stable demands. But conventional networks with many vehicles make it hard or impossible to pre-calculate all routes, especially when no pre-arrival information is available (see LE-ANH 2005). Therefore many real-world AGV systems rely on a set of simple rules which help to prevent any kind of conflict in the system (OLMI 2011). In a similar way, we do not put the goal of achieving collision-free routes in the context of routing but will discuss local rules for this task in the context of intersection control in the last section of the literature review.

**Transferring ideas from communication networks to logistics**

As stated, most approaches from the AGV literature are not really applicable to the type of systems which we want to consider in our study. However, there is also research on more complex systems available. Several approaches are mentioned in the next section. A general trend which can be observed in this context is that more and more researchers are realizing the huge structural and behavioral similarities between logistics networks and communication networks. Several research projects were launched to explore the potential benefits of the "Internet of things" (see e.g. TEN HOMPEL ET AL. 2006 or GÜNTHNER ET AL. 2008). Logistics entities travel in logistics networks like data packets in communication networks. As a lot of research has already been accomplished on routing in complex communication networks, the idea of transferring these insights to logistics networks is extremely appealing. Ad-hoc communication networks which change their configuration quickly and constantly are especially important areas of research. They provide analogies to the dynamics in complex logistics systems. But besides all similarities, there are also several distinctive characteristics which need to be considered and do not make the development of decentral logistics routing protocols an easy task (see SCHOLZ-REITER ET AL. 2006 and TEN HOMPEL ET AL. 2010). JOHNSTONE ET AL. (2010) who develop a dynamic routing policy which uses learning agents or SCHOLZ-REITER ET AL. (2006) who propose a concept for a general logistics routing protocol are examples for the transfer of communication protocols to the logistics domain. We will provide two other examples below. As the knowledge of the communication protocols OLSR and DSR is essential for understanding these, we will give an extremely brief and simplified description of their functionality. The interested reader is referred to the initial publications of JACQUET ET AL. (2001) and JOHNSON & MALTZ (1996).

The proactive protocol has the advantage that a route is immediately available when required. On the other hand dynamic source routing creates less communication overhead:

- Optimized link state routing
  The basic idea of link state protocols is that the network nodes exchange information about their neighbors. Each node senses its current neighbors and uses a message to broadcast this information in the whole network. Having received this information from all network nodes, each node is able to maintain its own graph of the network and calculate a routing table with the shortest paths. The neighbor information is updated periodically to keep the local routing tables up to date. The optimized link state protocol uses the concept of multipoint relays to reduce the communication overhead for the information broadcast.

- Dynamic source routing
  This routing strategy has been used in various wired and wireless environments. To implement dynamic source routing two different routines are required: route discovery and route maintenance. Each source node keeps a cache of known routes (with an expiration date). If a known route expires or a packet needs to be sent to an unknown destination, the source

triggers the route discovery process. A route request is broadcast and forwarded from node to node according to a certain set of rules. Once the route request reaches its target, the latter returns a route reply which contains all the nodes that need to be visited. In communication networks the hop count is more relevant than the distance. When using dynamic source routing, there are no periodic updates of the routes as in other routing protocols. Therefore a routine is required which checks if the routes in the cache are still valid. This route maintenance is implemented as a hop-by-hop acknowledgment or an end-to-end acknowledgment. If the acknowledgment process fails, an error message is returned to the source and the routes in the cache have to be updated.

**Decentral and distributed routing**

Having introduced the classification scheme for routing algorithms and having highlighted some of the major routing research topics, we will now focus on the analysis of decentral routing algorithms. Looking at the classification scheme, it becomes evident that instead of "decentral" only the term "distributed" is used. LAU & WOO (2008) have chosen this name correctly because without acquiring additional knowledge about the network from other nodes, no meaningful local routing decision can be made. Routing without global knowledge about the network can never guarantee that a certain destination is reached. This limitation has to be kept in mind when assessing the contributions of several authors below. Although these authors might use the concept of "decentral" routing, it does not correspond to the local meaning which we require in this thesis.

There are different approaches for implementing the decentral or distributed routing algorithms. The next paragraphs illustrate the most influential ideas. We start with authors that follow the idea of multi-agent systems and afterwards state approaches for transferring communication network models to logistics. Finally, we take a look at two optimization techniques by reviewing ideas in the context of model predictive control and ant algorithms.

One of the first attempts to come up with a decentral routing algorithm was made by TAGHABONI & TANCHOCO (1995). They developed an incremental route planner for AGV systems which does not select a route at the starting node, but rather decides the next node to travel to during the journey. Their conceptual system can be seen as one of the first agent-based routing systems. The authors also evaluate the performance of their strategy in two simple layouts and come to the conclusion that a complete route planner works better in complex layouts. From our perspective the weakness is that local and global information are combined to make the routing decisions.

The following authors explicitly adopt the concept of multi-agent systems for the development of their routing strategies. LAU & WOO (2008) introduce a dynamic routing algorithm for automated material handling systems. It uses a zone control logic and comprises a route discovery process which is based on message broadcast, a multi-attribute route selection function and a fault management function. The implementation of the selection function remains unclear. The developed routing strategy outperforms several other strategies which the authors compare in a simulation study of a generic loop-based layout. It can be defined as a distributed approach. In a decentral implementation local information is aggregated.

HALLENBORG (2007) tries to develop a multi agent control system for a conveyor-based baggage handling system. He focuses especially on the design of the agents and their communication. There is one route agent that is responsible for route selection when a route is requested by another agent in the system. This means that the approach does not completely follow the decentral approach and there is still global knowledge involved. The authors claim to use a static routing strategy and an incremental

route choice approach. But the exact functionality as well as the logic of the route agent for choosing one of the available routes remains unclear.

HOFMEISTER ET AL. (2010) develop a concept for a routing strategy which uses a label correction algorithm for decentral route updates. They claim that this reduces the communication overhead. The routing is to a certain extent based on ideas from the AGV literature which were mentioned above. It pre-plans routes and uses this information to consider future path utilization in its route selection. The distance to all destinations in the system is used as a decision criterion and therefore global information is required. The whole paper is conceptual and no performance evaluation is presented.

The following two examples from FAY ET AL. (2008) and TEN HOMPEL ET AL. (2010) show concepts that try to transfer the ideas of routing in mobile ad-hoc networks to logistics systems. While the former authors use an adjusted version of the link state protocol, the latter make reference to dynamic source routing. Although logistics and communication networks have a similar structure the operating conditions are different and modifications of the initial network protocols are required. In order to make sure that a source-sink route is definitely found, both routing strategies aggregate local information to global knowledge. The requirement of using local information is not fulfilled.

Besides finding a certain route it is also necessary to balance the load if there exist several routes to one destination. TEN HOMPEL ET AL. (2010) implement a traffic count methodology. They propose using a utility function which models the trade-off between the standard travel time on a route and the current traffic on this route. The function is not explicitly stated. FAY ET AL. (2008) use a probability function. It uses the expected travel times on alternative paths to the current destination as a decision criterion. The aim is to use the fastest path as the preferred solution, i.e. with highest probability. If paths with more or less equal travel time are available, the load should be distributed equally to them and even paths with rather long travel time have to be chosen once in a while in order to collect information about the current network status, i.e. the current travel time. For making this route choice decision the overall travel time from the current node to the destination via the different alternative paths has to be known. This means local information has to be aggregated.

FAY ET AL. (2008) apply a real-world simulation model to test the functionality of their algorithm. As the computational performance is not sufficient for simulating the complete baggage handling system, they only consider a certain section. This reduces the number of vehicles in the system from 400 to 14. The only results which are shown consider a disturbance scenario. The performance loss after the breakdown of a path segment is bigger for the central routing than for the decentral routing strategy. TEN HOMPEL ET AL. (2010) also use a baggage handling system simulation to test their multi-agent control system. But several adjustments are necessary and they therefore do not analyze the logistical performance as the results could not be transferred to a real-world scenario. They focus rather on the amount of messages which is transmitted during the simulation runtime.

**Optimization techniques and their applicability**

The following two approaches to decentral routing complete this section of the literature review. They are both optimization techniques rather than reactive control strategies. However, they should be mentioned in this context. The summary of the ant optimization algorithm can especially provide further insights. It is often mentioned in the context of decentral control and was the inspiration for the local dispatching strategy which will be introduced in Chapter 5.2.5.

TARAU (2010) proposes a control system for the route choice task in a DCV baggage handling system based on the idea of model predictive control. She develops a central, a decentral and a distributed version of the control strategy. Due to many assumptions and mathematical details the core

functionality of the local optimization algorithm is hard to verify. The distributed version of the algorithm allows communication with other nodes and performs better than the local decision rule. Two different systems are compared with a simulation study. The bigger one contains 4 loading stations, two unloading stations and 9 junctions. The input data is a custom-made sequence of bags. The analysis of the output data focuses rather on the computational effort of the algorithms than on logistics performance indicators. Solving mathematical problems seems to be more important than logistical problems. But the approach remains the only one that really focuses on local decision making.

During the 1990s a whole class of heuristic optimization techniques was developed. The initial idea of the Italian scientist Marco Dorigo (see e.g. DORIGO ET AL. 1996) to transfer the foraging behavior of ants to solving combinatorial optimization problems has been adopted and improved by many researches during the following years. In the late 1990s this whole class of optimization algorithms was referred to as Ant Colony Optimization (ACO). All those algorithms use an analogy to real-world ant colonies and try to imitate their behavior. Foraging ants deposit pheromones along their chosen path. Additionally, they choose from several paths based on the current pheromone concentration. Paths with a high concentration have a higher probability of being chosen. Due to the quicker return of ants on the shortest path, its pheromone concentration increases quicker than the concentration on other paths. In return, it is more likely that the next ant will chose the shortest path again. The idea of communicating indirectly by modifying the environment has been name stigmergy in the scientific community (see DHILLON & VAN MIEGHEM 2007). Although there are different versions of ACO, they all rely on the stigmergy concept.

AntNet was developed to apply ACO for routing in communication networks. Its procedure is explained briefly as it is the basis for one of the decentral dispatching algorithms which is introduced in Chapter 5.2.5 (see DI CARO & DORIGO (1998) for a detailed description of the AntNet algorithm):

- Artificial ants start at regular intervals from each node in the network. Each ant tries to find the shortest path to a randomly assigned destination node.
- At each switch, the ant selects its next node based on local, private and heuristic information. It stores its chosen route nodes and additional information, e.g. the travel time.
- Once it arrives at its destination the ant travels back to the starting point of its journey. It takes the same path back that it took to get to its destination. During the journey the local knowledge of each visited node is updated based on information which the ant collected and the quality of the followed path.
- The artificial ant dies once it returns to the source node.

The core elements for routing the artificial ants and for making the direction decision (step 2) are probability tables at each switch. They contain a probability for choosing one of the neighbor nodes when the current destination is to be reached. A direction is taken based on a combination of this probability and the queue length of the next link. Once an ant has completed its trip, it gives feedback about the quality of the found solution and the probability tables are updated accordingly. Various different feedback functions have been developed and tested. It should be noted that the artificial ants are complementary to the data packages which are routed through the communication network.

Several ACO algorithms have been proposed in the literature but only a few contain performance analysis. DHILLON & VAN MIEGHEM (2007) try to fill this gap by comparing the performance of the AntNet algorithm with the Dijkstra algorithm.

In logistics literature, it is often argued that ideas of swarm intelligence and ant behavior should be used for routing in logistics networks. But it needs to be considered that the AntNet algorithm has been developed as an optimization algorithm. Artificial ants are used in a network to explore it and find the shortest paths. From a conceptual point of view the ants would therefore not be representations of the vehicles in a transport system. This is not the intention of the initial algorithm. Each vehicle uses artificial ants for optimization and decision making (see e.g. CLAES ET AL. (2011) for an application in a traffic network). An IT infrastructure would be required to run the ant algorithm for each vehicle. Those optimization runs are computationally expensive. This can be especially problematic when only limited planning data is available and the real-time requirements are very strong.

Due to these limitations a second line of thought is possible. Deviating from the initial intention of the algorithm, the vehicles could also be directly represented by the ants. As the vehicles are not as numerous as the artificial ants, they would discover the network much slower. The results of this routing strategy cannot be expected to reach the same performance as the first implementation opportunity. Additionally, an application of this rule requires dealing with many disadvantageous features of the AntNet algorithms. For example stagnation effects need to be mitigated in order to allow discovery of new shortest paths when the status of the network changes (see SIM & SUN 2003).

Generally, the applicability of the algorithm also depends on the network structure. For example, baggage handling systems have a rather low density of connections when being compared to communication networks. The usage of the ant-based routing is therefore less appropriate (HALLENBORG 2007).

In this section we have provided a general classification of routing strategies. Looking at different applications in logistics and communication networks, it has been found that a purely decentral routing strategy cannot be developed.

## 3.3 Intersection control

A review of existing control approaches for intersections completes our literature review. The two relevant intersection types are switches and merges. At first sight, the control of intersections does not seem to have a major impact on the system performance. But during our research we noticed that the understanding of intersection control should not be limited to the pure control of traffic flow. Intersection control can also have an impact on the requirements of conflict-free routing and avoiding congestions and deadlocks. It can find answers to some of the questions which were raised for routing and scheduling.

There are only a few papers which specifically deal with intersection control. It is usually only considered as one of many sub-problems within the development of control systems. But for decentral control systems intersection control gains importance. Especially in systems which do not allow pre-planning of schedules and routes, many problems have to be solved on the operational level and the control of intersections becomes a relevant control factor. Its reactive nature can replace the planning activities which are not feasible in many situations. During the next paragraphs we will first introduce a general classification of intersection control rules which was published by GUDEHUS (2005). Afterwards we will take a look at existing implementations and the possibilities for influencing conflicts, congestion and deadlocks with intersection control. The classification scheme distinguishes two types of rules:

- Right-of-way rules at merges
- Parallel operation strategies at switches

Looking at the first class of strategies, a very simple approach is to use a FCFS logic for assigning priorities to vehicles which arrive at a merge. The vehicle which arrives first is allowed to occupy the merge and vehicles which arrive afterwards have to wait until the merge is cleared. The vehicles are processed in sequence of their arrival. Two other rules always give priority to one of the inlets of the merge. One rule is referred to as "simple right of way". This means that vehicles arriving from the inlet with low priority get directly processed whenever the gap between two subsequently travelling vehicles on the priority lane is big enough. In case of an "absolute right of way" rule, the vehicles on the lane with low priority always have to stop and wait for a sufficient gap.

For the switches the main question is how to distribute the arriving vehicles to the successor nodes. GUDEHUS (2005) assumes for his classification that all successor nodes have the same properties and abilities. A first strategy type sends the vehicles periodically to one of the successor stations. An extended version of this strategy does not only follow a periodic approach but additionally groups the vehicles into batches. The last relevant strategy uses the utilization of the successor stations as a criterion. It sends the vehicles always to the station with the lowest utilization. While the applicability of the first two strategies is limited to environments where the assumption of equal station properties holds, the latter one points to the direction of our understanding of switch control. Based on local information, i.e. the utilization of the successor node, a decision about the next node can be made.

Looking at the classification scheme, it is obvious that intersection control is the first strategy type which can easily adopt a decentral perspective. Decisions are only based on location information and can be used to locally coordinate the system activities. A higher degree of centralization would require more complex structures and does not necessarily promise a higher system performance. Local decision making enables the usage of simple decision criteria. Instead of complicated planning procedures a pure rule-based control paradigm can be followed.

MIN ET AL. (1999) use such a rule-based approach for traffic coordination in a mobile robot environment. They use the mobile robots to represent AGV. Besides rules for maintaining a safe distance to other vehicles and avoiding obstacles, they apply a FCFS rule at merges. Trials with three robots are carried out. WEYNS & HOLVOET (2008) also focus on avoiding collisions at merges. They develop a concept for an agent-based system. The agents use hull projections to coordinate their actions. The agents interact based on local virtual environments. Using these as a communication mode, they choose a path and request approval by sharing their hull projections for this path with other agents. Only when the other agents approve the route choice, do they continue their journey. Otherwise they either wait or select another route. This approach has a lot of requirements regarding the communication technology which is used. A related concept has been developed by ARORA ET AL. (2000) who introduce a junction control approach which is based on game theory. They assume that there are technologies in place which enable a vehicle to sense its own distance to a junction and the distance of other vehicles which are approaching the same junction. FRAZZOLI ET AL. (2005) propose a control rule which is based on reserved regions which vehicles claim to prevent collisions. The concept assumes that the vehicles are able to sense the position of other vehicles within a certain radius.

Summarizing these theoretical concepts, it can be said that merge control is primarily used to prevent collisions on an operation level. These rules can supplement or replace the collision-free route planning which has been introduced above. Most of the concepts require high communication overhead or sophisticated sensors for perceiving the environment. Only the approach of MIN ET AL. (1999) could be implemented with limited technical means. In addition to avoiding collisions, merge control also has an impact on the performance of the system. This aspect is barely mentioned in the publications.

After merge control, we look at switches. There is nearly no literature on this specific topic, the reason being that the control of switches is usually considered as an aspect of the routing algorithm. The switches simply select one of the directions based on the routing tables. But looking at the general classification of GUDEHUS (2005) above, it becomes evident that the switches could also be used to influence the load distribution in the system. Thereby they have an impact on congestion and the overall performance of the system. Based on the current utilization of the successor links or successor nodes, a switch can decide which path an arriving vehicle should follow. A prerequisite is of course that only directions which allow the vehicle to reach its desired destination are considered. Balancing the utilization is possible with very limited local knowledge. But this local distribution does not necessarily achieve a global optimum. Otherwise global information would need to be included in the decision making process. Several authors which try to integrate this load balancing approach in their routing algorithms have already been mentioned above (e.g. TAGHABONI & TANCHOCO 1995, FAY ET AL. 2008, TEN HOMPEL ET AL. 2010).

In addition to avoiding collisions and distributing the system load, merges and switches are also important for avoiding deadlocks in decentrally controlled logistics systems. The term "deadlock" has its roots in informatics but can easily be transferred to various system types. It becomes relevant when tasks in a system are carried out concurrently. Therefore the importance of deadlocks grows with the degree of decentral planning and local decision making. A deadlock occurs when a group of tasks or processes is waiting for an event which can only be triggered by one of the members of this group. In this situation the systems locks itself and cannot proceed. In logistics systems this situation usually occurs when two or more vehicles block each other and none of them can continue. COFFMAN ET AL. (1971) defined four conditions which cause a deadlock when they are present concurrently:

- Mutual exclusion condition:
  All tasks claim exclusive control of the resource which they control.
- Wait for condition:
  Tasks are keeping resources which have been assigned to them while waiting for additional resource.
- No preemption condition:
  Resources cannot be removed from their current tasks until the tasks have completed their usage.
- Circular wait condition:
  There is a chain of tasks and the tasks in the chain have to wait for resources which are currently used by other tasks in the chain.

LEHMAN (2006) states that deadlock handling aims at making sure that at least one of the four conditions is never fulfilled during the operation of a system. As the first condition usually applies in technical systems, research focuses on the latter three. This means that systems can either be designed in a way that tasks always claim all resources they need at the beginning (condition 2) or release all allocated resources if they are not granted the missing resources (condition 3). Another option is to introduce a linear order for the resources (condition 4). Generally, three different deadlock handling strategies can be distinguished (see LEHMANN (2006) for the classification and the following example).

We use the case of two vehicles which have to coordinate their actions for crossing two bridges from opposite sides as an example:

- Deadlock prevention: Rules are introduced which make it impossible that all of the four conditions above are present at the same time. In the example a unidirectional traffic is introduced on both bridges.
- Deadlock avoidance: This strategy aims at avoiding deadlocks by recognizing them when they would be the consequence of the next action. Rules which trigger a dynamic resource allocation in these situations are used. In the example traffic lights are introduced which allocate the bridge resource to one of the vehicles at a time.
- Deadlock detection and resolution: This strategy does not try to avoid deadlocks but tries to find strategies which detect and resolve deadlocks once they occurred. In the example one of the vehicles needs to drive backwards when the deadlock occurs on the bridge.

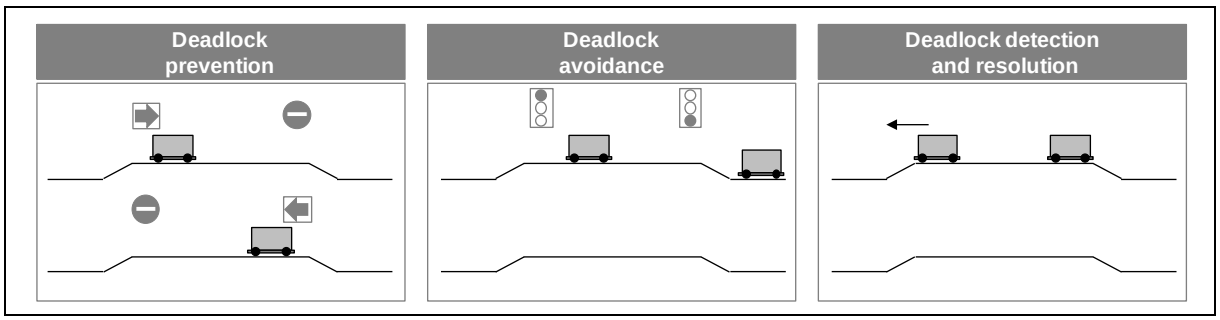



Figure 17: Deadlock handling strategies (compare LEHMANN 2006)

The applicability of the different strategies depends on the system under consideration. Deadlock detection and resolution is hard to implement in dynamic logistics systems. Technical limitations complicate the deadlock resolution. Vehicles can usually not travel backwards. Therefore deadlock prevention and avoidance is emphasized in logistics systems. Four different approaches can be distinguished on an operational level (see VAN DER MEER (2000) and LE-ANH (2005) for an overview and reference to other authors):

- Forward sensing of vehicles
- Balancing the system workload
- Traffic control at intersections
- Zone planning

The first approach assumes a certain sort of intelligence for the vehicles. They have technical means to sense their environment and can prevent deadlocks by identifying the positions of surrounding vehicles. The coverage of this approach is limited and it might happen that a deadlock can no longer be prevented when a vehicle senses the risk. The second approach has already been discussed above. The switches in the system can be used to distribute the load in the system. An even distribution reduces the risk of an accumulation of vehicles and thereby the risk of a deadlock. Mitigating the risk of a deadlock might in this case have an influence on the achievable system performance.

The control of traffic at the intersections refers in the first place to the strategies which have been discussed above for merges. By only allowing a certain number of vehicles to enter a merge at the same time, the risk of mutual interferences and blockages is reduced. The zone planning approach is

related to this topic. It does not limit its attention to merges, but divides the whole system into zones. Afterwards only a defined number of vehicles are granted access to those zones at the same time. The limited number of vehicles in the same zone helps to prevent deadlocks. A careful implementation would only allow one vehicle per zone. One example system that is related to zones is the control system of the Flexconveyor (see MAYER & FURMANS & 2010). This system contains an algorithm that avoids deadlocks in circular configurations of the Flexconveyor modules. The main idea is that a module denies a load access to the loop when it detects that it is the last idle module within this loop. The loop can be considered a zone. The different modules exchange information about their individual status. This aggregation of local information allows each module to make judgments about the current deadlock risk. The algorithm has been specifically developed for the Flexconveyor system and its characteristics. Although the main idea could be transferred to many logistics systems, the scalability and technical feasibility in bigger systems remains unclear. The considered test systems consist of less than 20 modules. In general, the aggregation of local information is the first deviation from the local principle that we have identified for intersection control. It is obvious that a local approach can never reach the same performance as global knowledge of the deadlock risk. But intersection control seems to allow an application of local decision rules for deadlock prevention and avoidance. The loss of efficiency needs to be evaluated.

## 3.4 Concluding remarks

The literature review verified the hypotheses we claimed during the introduction of this study. Generally, there exists a lot of literature on decentral control of material handling systems. Routing is the topic which is tackled most frequently. Not all facets of dispatching are covered and the impact of intersection control seems not to be completely recognized yet.

Dispatching is the core topic of our study. When it comes to vehicle-load assignment, scientific literature mostly proposes the usage of global heuristic rules. Truly local strategies have only been developed for very simple systems. All the approaches that follow the multi agent paradigm aggregate local information to global knowledge when making dispatching decisions. Positioning of empty vehicles is even more problematic. There are only a few publications which propose control strategies for this topic in complex networks with many vehicles. Among the proposed strategies, only one deals with distributed vehicle positioning. All the other strategies assume loop circulation. The only available control strategy that uses distributed positioning requires global information.

Similar conclusions can be drawn for the routing algorithms. A goal-directed routing algorithm which makes sure that a vehicle definitely reaches its desired destination cannot be developed without global knowledge. Therefore all the proposed algorithms follow the distributed paradigm rather than the idea of decentral control. This is true for logistics and communication networks. The transfer of the ant algorithm to logistics networks is not as trivial as often stated. Depending on the implementation strategy the transfer will either require a high computational effort or performance losses and negative side-effects need to be expected.

Compared to dispatching and routing, intersection control has the biggest potential for decentral decision making. Rules-based control strategies are already discussed in literature. However, the capabilities of these strategies need to be evaluated further when decentral systems are developed. Care should be taken not to intuitively include an information exchange into the algorithms.

In general, many of the mentioned publications are of a rather conceptual nature. They simply describe how a system should work. But they do not prove the feasibility or measure the performance. Whenever a simulation model is used for verifying the characteristics of the control system, the model

complexity is very low. This does not allow confirmation of the feasibility of application in real-world scenarios and supports our hypothesis of technical limitations.

Routing and dispatching are often related to each other. Many dispatching rules require routing mechanisms as they are based on travel time. Travel times can only be determined when the underlying routes are known. Although this study focuses on dispatching, it needs to consider routing mechanisms at the same time to account for the interdependences.

Besides the communication requirements, decentral control also has an interesting impact on the amount of information which need to be stored in a system. While it is possible to prevent redundant information in central systems, there will always be a certain level of redundant information in distributed systems (see HOFMEISTER ET AL. 2010).

Having completed the literature review, the next chapter introduces the generic test layout which is used for the simulation experiments in this thesis. The conceptual implementation of the AutoMod model is discussed briefly.

# 4 The test environment

## 4.1 System description

For testing, evaluating and comparing the performance of different control strategies a generic simulation model is used. It shall answer the questions which were raised above. The basic test layout is a transport system which consists of one loop and a shortcut (see Figure18). The loads enter the system at the sources (Q1, Q2) and have to be transported to the sinks (S1, S2).

The transports are performed by vehicles which move on unidirectional driveways that connect all the stations. Idle vehicles can be parked at two storage locations (E1, E2). All stations are located on loop sidings of the main path in order to avoid blockage effects.

The system allows for testing of all control strategies which were mentioned. Questions about dispatching can be answered as several sources and sinks are included. The storage locations allow an evaluation of the empty vehicle positioning aspects. Routing can be tested based on the shortcut. Finally, intersection control could be evaluated based on switches and merges of the system. More complex layouts can be created by connecting several mirrored duplicates of the initial system. The generation of different layout scenarios for this thesis will be explained in Chapter 6.1.1.

Based on the test system advantages it is proposed to use the test system in other studies which treat of similar control strategy questions. This makes results in the scientific community more comparable and the overall learning curve becomes steeper.

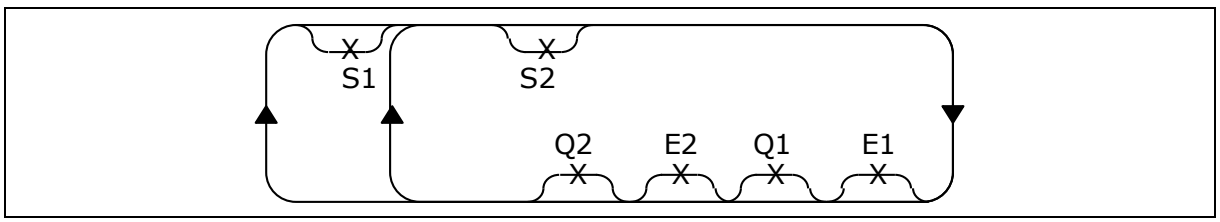



Figure 18: Basic test system

After the introduction of the test environment, the next sections summarize the key characteristics of the simulation model implementation.

## 4.2 Conceptual description of the AutoMod implementation

The simulation model has been implemented using the AutoMod simulation software. The transport system is represented by an AutoMod path mover system. This system type was chosen because it contains various pre-defined functions which can be used to manipulate the behavior of the vehicles. However, it turned out during the ongoing implementation process that those functions did not provide the required flexibility. Especially for the control of empty vehicles, the AutoMod standard procedures interfere with the customized implementation and restrict the functionalities. It was therefore necessary to develop a self-contained logic for controlling the vehicles. This logic uses the AutoMod infrastructure and some of its functions but detaches the actual control of the vehicles.

The customized vehicle control system uses control points to represent the different stations (or decision point types) which were introduced above (see Chapter 2.3). The core idea is to send the vehicles from one of those control points to the next until a destination is reached. This approach allows the highest degree of flexibility. The behavior of laden and unladen vehicles can be influenced whenever required. The functionality of forwarding vehicles is implemented in the procedure pTravel. Two types of loads are required:

- Loads of type “full” – They represent the actual transport orders which need to be transported from sources to sinks.
- Loads of type “empty” – Every time a vehicles travels unladen, it transports a load of type empty.

The introduction of these two load types is necessary to have complete control of the vehicles. Once a load has been picked up by a vehicle, the load acts like a passenger in a taxi and gives the vehicle directions. It is easier to influence the vehicle behavior indirectly via the load than by using the vehicle functions.

At the beginning of each trip a vehicle, i.e. its passenger load, gets assigned a number of attributes. Those attributes are derived from the dispatching process and define a destination for the vehicle. This destination information is afterwards used in the procedure pTravel to route the vehicle.

The required attributes are stated below:

- aiDestination – This is the destination index of the load. It represents a sink for loads of type full and a source or storage location for loads of type empty. If aiDestination is set to 0, the vehicle selects its next control point randomly from the successors of each switch. Based on aiDestination the remaining three attributes are set.
- aiPickupLocation – If <> 0 the vehicle is currently in the retrieve mode and either looking for a load or is already claimed and travelling to pick up the claimed load.
- aiDeliveryLocation – If <> 0 the vehicle is currently transporting a full load and moving to deliver it to a sink.
- aiParkingLocation – If <> 0 the vehicle has been assigned to a storage location and is currently travelling to this location.

While the first attribute is used to actively make the vehicles choose a direction, the latter three are mainly used to record statistics. As the whole AutoMod logic for vehicle control is being replaced, the standard AutoMod reports do not contain valuable data and activity tracking needs to be customized.

As already stated, the loads are sent from one control point to the next. There are six different control point types in the system which are used for this purpose. They represent decision points. Each station is assigned a consecutive number between 1 and $m$, where $m$ is the number of sources in the system. By definition the number of sources equals the number of sinks and the number of storage locations in the test system. The following paragraphs introduce the different control point types which are relevant for dispatching and briefly describe the related processes:

- q_(1…*m*):
  These control points represent the sources. Loads enter the system at these locations. Depending on the dispatching strategy vehicles wait here until a load becomes available. Arriving vehicles deliver an "empty load". They wait until a "full load" is assigned, retrieve it and continue their journey to deliver it.
- s_(1…*m*):
  These control points represent the sinks. The loads are transported here from the sources. Once a vehicle delivers a "full load", an "empty load" is automatically created. The "empty load" claims the vehicle which just delivered the load and is now its new "passenger". The dispatching strategy assigns the load a new destination.
- etsin_(1…*m*) and etsout_(1…*m*):
  Each storage locations consist of a set of two control points. They represent the entrance and the exit of the storage locations. The exit is mainly relevant for vehicle control. The entrance is only required to record the time a vehicle spends in the storage location. Vehicles deliver an "empty load" to the exit station if they were moving to park. Having delivered the "empty load", a copy of this load is created and picked up by the vehicle. The vehicles wait at the exit station until they have claimed a load at a source or have been claimed by a load. Then they continue their journey. If they have been reassigned while travelling or waiting in the storage location (in a queue behind the first load which is waiting at the exit station), the vehicles do not stop at the exit, but continue their journey directly without getting a new "empty load".

The different processes are summarized in the figure below. The only process which has not been mentioned yet is the direct return from a sink to a source. In this case the storage location is not visited. The described processes are simplified. There are various exceptions which are part of the simulation model but are not required for this conceptual description. One example is the release processes which will be introduced during the next chapter.

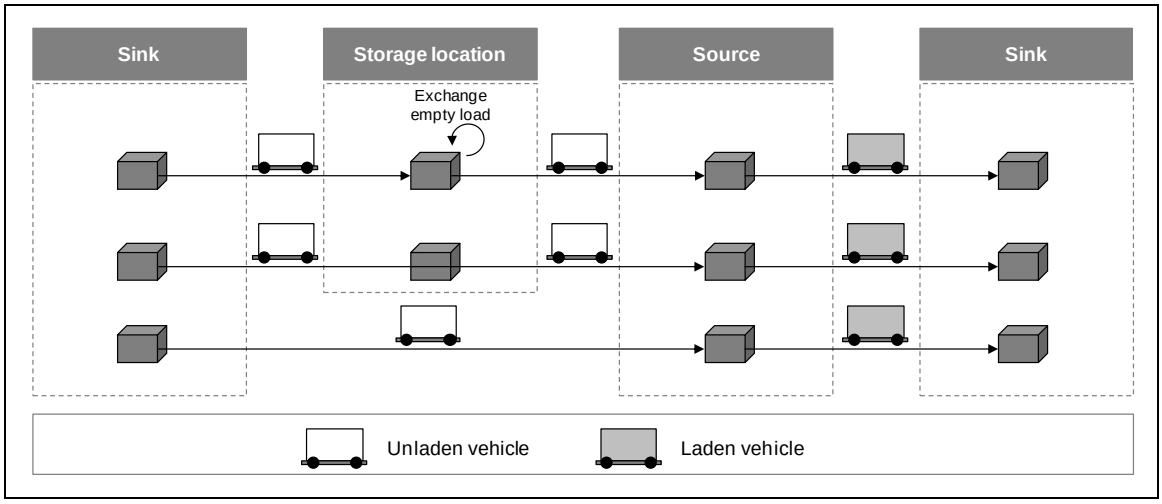



Figure 19: Standard vehicle processes

Two other control point types are required to implement the routing process and intersection control. These control points are always used in sets of three at switches and merges. Similar to the control points which have been mentioned above, each set gets a consecutive number between 1 and *o*, where *o* is the number of switches in the system. Based on the layout the number of merges equals the number of switches:

- d1_(1…*o*), d2_(1…*o*), d3_(1…*o*):
  The control point with set index 1 is put at the inlet of the switch (see figure below). The other two control points are put at the outlets. A routing table is maintained at each switch. The table defines the next node to choose when traveling to a defined destination on the shortest path. Based on the aiDestination attribute of an arriving load, the routing table is used in the pTravel procedure to define the outlet that the vehicle has to proceed to. This process implements the routing algorithm. During the description of the control strategies (see Chapter 5.2) exceptions from this standard procedure will be defined.
- m1_(1…*o*), m2_(1…*o*), m3_(1…*o*):
  Similar to the switches, three control points are used to control the merges. A vehicle claims the merge resource when it arrives at the inlet and releases it once it passes m3_(1…*o*). Using a counter with capacity 1 for each merge, an arriving vehicle either has to stop when the merge is already occupied by a vehicle or can proceed directly to the outlet. With this approach a FCFS logic is implemented for giving right of way.

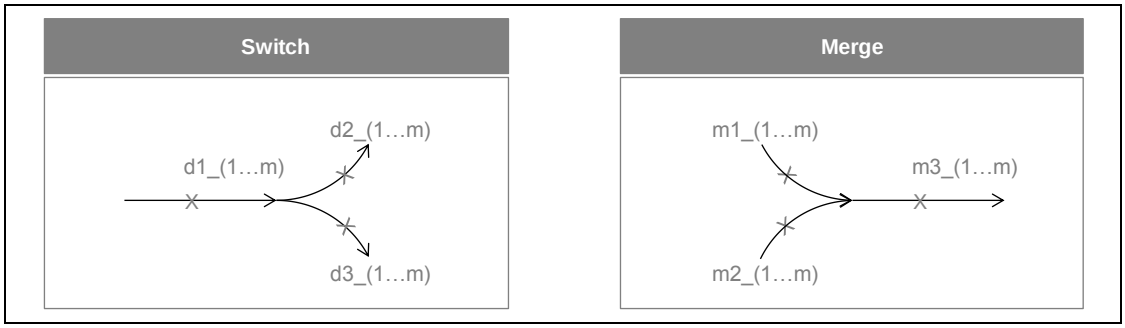



Figure 20: Control points at switches and merges

The last control point type does not have any impact on the control of the vehicles with respect to the implemented control strategies. All control points which are named as “h_” and a consecutive number just serve as a tool to implement a “non-accumulating” path mover system. The reasons for using “non-accumulating” systems will be explained below.

Several standard AutoMod vehicle functions are used to implement the dispatching control logic. The functions are modified in order to fulfill the dispatching processes as they are described in Chapter 5.2. The core functionality is integrated in the following functions and procedures:

- Load available function
  A load calls this function once it enters the movement system and before being available for a vehicle claim. Therefore it is possible to make the load actively claim a vehicle in this function. This is the place where all load-initiated dispatching activities are implemented. It needs to be noted that the function is used for “empty loads” and for “full loads”. Therefore attributes for controlling both load types can be set in this function. In case an arriving load does not find an idle vehicle, it is added to the list of waiting vehicles.

- Task search procedure
  This procedure is executed by a vehicle which has just completed a job. Each time a vehicle delivers a load, the task search procedure becomes accessible. It is used to implement all vehicle-initiated dispatching strategies at sinks, sources and storage locations. According to the different dispatching strategies the vehicle searches for loads in the lists of waiting loads and claims a load (if available).
- Passing station function
  Each time a vehicle arrives at a control point, the passing station function is called. It is called before the pTravel routine is executed and can be used to maintain several attributes of the vehicles. It is used to track and record vehicle statistics and the number of vehicles which are currently waiting at the storage locations. For some of the dispatching strategies core functionalities are implemented in this function. For example, forecast and feedback-based dispatching use the function to record the known dispatching destinations. Feedback-based dispatching additionally uses it for setting the next destination.
- Station finished function
  As its name suggests, this function is called when a vehicle leaves a control point. Similar to the passing station function it is used to maintain several statistics attributes. One of the most important is the update of the list with vehicles that are currently approaching the control point. Similar to the passing station function some dispatching activities also take place when the function is called. Feedback-based dispatching uses it for updating the dispatching tables.

Besides the standard AutoMod functions, the calculation of the routing tables is the most important custom function. It uses the Dijkstra algorithm to calculate the shortest path successor for each combination of system node and relevant destination. Relevant destinations are sources, sinks and storage locations. Whenever required for dispatching decisions, a second function provides the length of paths between any combination of source and sink based on the calculated routing table and the control point distances.

## 4.3 A note on the suitability of the chosen simulation model implementation approach

As already mentioned above, the initial choice of a path mover system in AutoMod led to additional effort in implementing the vehicle control system. The whole control logic had to be customized in order to make it flexible enough and tailor it to the needs of our experiments. During these experiments some additional disadvantages of this system type were discovered:

- The runtime performance decreases with the number of vehicles in the system. The vehicles continuously sense their environment for collisions. These sensing activities require many calculations and slow down the simulation. In order to maintain an acceptable runtime for the models, the system default had to be changed from an “accumulating” to a “non-accumulating” type. This improves the runtime performance but requires that control points with a capacity of 1 are placed at a distance which equals the length of a vehicle. These are the control points which were named “h”. Vehicles are sent from one control point to the next one. They can only continue their journey when they are able to claim the next control point. Therefore vehicle collisions are prevented based on the control point distance and the standard vehicle-based collision control that is computationally expensive can be turned off. The additional control points only slow down the compiling, building and start process of the simulation model.

- The memory requirements increase with the number of vehicles. For systems with more than 2,500 vehicles we experienced problems on standard business notebooks.

For these reasons we recommend the usage of a conveyor system for future experiments which involve complex layouts and more vehicles. The performance of the AutoMod conveyor system can be expected to be better and the implementation effort should be at about the same level as for the path mover system. Maybe the effort is even a bit lower as there is no AutoMod logic which needs to be "outflanked".

An additional remark shall illustrate another aspect that has been found to be a major driver of memory requirements. Waiting loads have a huge impact in this context. Therefore the load generation in the system which does not achieve the required throughput has to be stopped in order to avoid the memory running low.

After these rather technical aspects and a brief description of the implemented AutoMod system, we will now focus on the actual implementation of the different control strategies.

# 5 Implemented control strategies

This chapter starts with introducing the assumptions for the development of the tested control system. Afterwards the implemented control strategies are described in detail. The development process for some of the decentral strategies has taken several iterations. First ideas were developed, implemented and tested. Often modifications were necessary afterwards. The following descriptions only contain the final implementations. But key findings from the development process which are relevant for a general understanding of decentral control systems and which can contribute to future implementations are summarized.

## 5.1 Assumptions

For the development of the control strategies several assumptions are required. The main subject of this thesis is dispatching strategies. To apply these strategies to an overall control system, we assume firstly that there is a routing algorithm in place which provides a static routing table at each switch. The decision whether the routing algorithm is implemented as a central, distributed or decentral process is not in our scope. We simply suppose that vehicles can be routed from every location within the network to a destination once the dispatching decision has been made. The following discussion of dispatching strategies and their destination assignment only considers empty vehicles. For all laden vehicles the standard routing procedure is applied which brings the vehicles from their pickup location to the defined delivery location.

For the control of merges we assume a FCFS prioritization. During the implementation process additional routing and intersection control strategies became necessary. Those are explained in the next subchapters.

The core requirement for dispatching is that all node and vehicle decisions are limited to local information. There is no information exchange within the network. Only local communication can be used. All entities are allowed to process the information they perceive in their local environment. There is no a-priori information available for dispatching. The only exception is the relationship between sources and their corresponding storage locations. There is a storage location located in front

of each source (see Figure 18). These node sets (sources and storage locations with the same index) know about each other and are able to communicate. The exception is necessary as the sources need to call vehicles from the storage locations whenever new loads arrive. Otherwise the system could not start its operation. This call-off is the only allowed inter-node communication.

The vehicles in the system are equally distributed to all storage locations for the start of the simulation. Only one transport request can be assigned to a vehicle at a time, i.e. no planning of vehicle tasks is possible. Reassignment is not allowed for any of the dispatching rules. This means, once vehicle and load have been assigned to each other, this match is not changed or split up again[1].

All loads are processed on a FCFS basis. This has implications for the load-initiated dispatching rules. A load which enters the system first checks if there are already waiting loads at its current source location (a list of waiting loads is used in AutoMod for each source). In the case where there are waiting loads, the arriving load is added to the list of waiting loads and the load with the longest waiting times is triggered to look for vehicles according to the different dispatching rules. If the load finds a vehicle, it is removed from the list of waiting loads. In the case where there are no waiting loads, the arriving load looks for a vehicle itself. In case it does not find a vehicle, it is added to the list of waiting loads. Whenever we talk about “arriving loads” in the following sections, we refer to the actions which are triggered by the load with the longest waiting time. This might either be the load which has just arrived (if no other loads are waiting) or a load which has already been waiting and whose actions were triggered by the arrival of the next load.

The lists of waiting loads are also relevant to the vehicle-initiated dispatching rules. The vehicles always search these lists for waiting loads. In case a load is available, the load with the longest waiting time is assigned and removed from the list of waiting loads.

Similar to the loads, the idle vehicles are also assigned on a FCFS basis. Each control point in the system has a list of vehicles which are currently approaching it. A vehicle is added to the list once it leaves the predecessor and removed when it arrives at the control point. The list of approaching vehicles contains the vehicles in sequence of their expected arrival at the control point. Arriving loads search those lists when they look for a vehicle. When loads find vehicles (according to one of the dispatching rules) they always claim the available vehicle with the earliest expected arrival at the specific control point. Only vehicles which have not been assigned a load yet are available for dispatching. Once a load claims a vehicle, the latter is no longer available for dispatching. According to the modeling approach all vehicles which are travelling between storage location entrances and exits are available for dispatching (unless they have already been claimed). However, it should be noted that the vehicle-initiated dispatching rules are only triggered when a vehicle has physically reached a control point, i.e. a sink or the exit of a storage location.

## 5.2 Dispatching

### 5.2.1 Core questions for the development of local dispatching rules

As there are no appropriate decentral dispatching algorithms available for the type of system that is subject of this thesis, some fundamental thoughts are required. They can afterwards be used for developing new decentral dispatching algorithms. Generally, dispatching seems to be an easy task. For the load-vehicle assignment it needs to be decided which load a vehicle that just completed a job is assigned to next. And for loads which enter the system the decision about choosing an idle vehicle has

[1] This is only relevant for D2 and D5 as the vehicle assignment process is postponed to the moment of vehicle arrival at the source for D1, D3 and D4 (see Chapters 5.2.2, 5.2.4, 5.2.5)

to be made. In a case where there are currently no loads in the system, vehicle positioning has to select a storage location for an idle vehicle. These core questions remain unchanged when limiting the decisions to local information. In this situation the challenge is to find information which allows meaningful decision making.

From the perspective of the loads which enter the system, there are not many alternatives. When limiting their decision to local information, they can either look for vehicles at the source or at the corresponding storage location. All other options would either involve the aggregation of information or information exchange. Therefore all the following strategies will mainly follow this approach for load-initiated dispatching.

The same conditions apply when an idle vehicle arrives at a storage location. It can only check if there are loads available at the corresponding source. If no load is available, the vehicle has to wait to be claimed. This is the first aspect of vehicle-initiated dispatching.

The second vehicle-initiated decision has to be made at sinks when the vehicle delivers a load. Due to the restriction of local information, no information about loads which are waiting at sources or the fill level at different storage locations is available. However, a dispatching decision is required. There are several options for designing the selection process of a dispatching destination:

- Option 1
  The most restrictive design would assume that the vehicle does not have any valuable information for making dispatching decisions and therefore has to continue travelling randomly. No dispatching destination can be assigned.
- Option 2
  The second option is to assume that the vehicle has perceived some kind of network information during the transport of the load. This information can be used to select the dispatching destination.
- Option 3
  Taking the idea of option 2 one step further, the available information is not limited to the knowledge of the vehicle. Historical information which was collected locally during the operation of the system can be used for a forecasting process. Based on the forecast a dispatching decision is made. Either the vehicles or the sinks can collect this information. It seems more beneficial to give this task to the sinks as they are most probably faster in acquiring information from the overall network.
- Option 4
  The last design option is to store information externally in the network. A feedback process can be used for updating this information.

The options 1 to 4 were the major lines of thought for developing the decentral dispatching rules which are introduced in the following sections. Each of the rules tries to operationalize one of the ideas with a suitable dispatching algorithm. In order to have a benchmark for assessing the performance of the decentral rules, a central rule which uses global information has been implemented. It is described in the last of the following subchapters.

As introduced in Chapter 3.1, there are three different events which mainly trigger the execution of the dispatching algorithms:

- A vehicle delivers a load to a sink. After completion of the job it searches for a new task.
- An empty vehicle arrives at a source or at a storage location.
- A new load arrives in the system and looks for a vehicle.

These three aspects are covered during the description for each of the dispatching strategies to state the load-initiated and the vehicle-initiated perspective.

### 5.2.2 Random dispatching rule (D1)

**Idea**
This rule is extremely simple. It refers to the first of the design options which were mentioned above and assumes that the vehicles which have just completed a job do not have any valuable information about the network. They cannot judge the current situation at sources and storage locations. Therefore the vehicles travel randomly after having completed a delivery job.

**Implementation**
A vehicle that delivers a load at a sink is not directly assigned to a new task as it does not have any information about the current network status. It continues its journey empty without a transport request assigned. As it does not have a fixed destination yet, it continues travelling randomly until it has reached a source or a storage location. Once an empty vehicle arrives by chance at a source, it has access to the transport requests which might be waiting at the source. In a similar way a vehicle which arrives at a storage location has access to the transport requests which might be waiting at the corresponding source. In a case where there are no loads available, the vehicle waits until it is requested by an arriving load at the corresponding source. Until the vehicle arrives at a source or a storage location, it cannot actively search for loads. It becomes available for loads after having arrived at source or storage location.

Loads that enter the system only have access to vehicles which are waiting either directly at that source or at the storage location corresponding to that source. In case no vehicle is available the load has to wait for the arrival of an idle vehicle. The vehicle will then initiate the assignment process, either at the source or at the storage location.

This dispatching strategy is an example of the difficulties with determining exactly the status of a vehicle (see Chapter 3.1). A vehicle which travels randomly can either be considered as being in the retrieve mode or in the go to park mode. It can either be considered empty or empty and idle. Similarly, it is hard to define the exact meaning of the load-vehicle assignment and the empty vehicle positioning share of the dispatching rule. Load-vehicle assignment is limited to decisions at sources and storage locations. Empty-vehicle positioning is accomplished by the random routing strategy as no dispatching destinations are assigned.

**Major lessons learned and changes to initial implementation**
The following table summarizes the major findings from the development and implementation process. It shows the problems which were observed and reduced the performance of the system. Next to each problem the implemented solution shows how the problem was solved.

| | Problem | Implemented solution |
|---|---|---|
| 1 | • The loads experience long waiting times before being picked up. The load-vehicle assignment for vehicles which are called from the storage locations takes place at the time of the call-off, i.e. before the vehicles physically arrive at the source. Therefore it may happen that a randomly travelling vehicle which arrived in the meantime but had not been assigned a load, blocks the pickup process of the assigned vehicle which arrives from the storage location. | • The assignment process is postponed until a vehicle actually arrives at a source. Vehicles are called from the storage location, but at this point in time they are not assigned to a specific load. Only vehicles which already have arrived at the source can be assigned to loads. Due to the possible arrival of randomly travelling vehicles, it might happen that more vehicles than initially required and called are available at the source. |
| 2 | • Vehicles arrive at sources and storage locations that currently do not have a vehicle demand or generally a low demand. These vehicles strand at those locations and are not available to support the transport process at other locations. | • A release process is implemented which checks in fixed time intervals if the current vehicle waiting time at a source exceeds the threshold $t_{sou}^{rel}$ or exceeds the threshold $t_{sto}^{rel}$ at a storage location. The time interval for the check is set to the minimum of $t_{sou}^{rel}$ and $t_{sto}^{rel}$. If a vehicle is released, it continues travelling randomly in the network. |
| 3 | • Due to low demand there might be tailbacks at certain sources. More vehicles than required arrive when travelling randomly and might temporarily block parts of the system and decrease its performance. | • An additional strategy for load-dependent source approach (see Chapter 5.4.2 below for a detailed description) is developed.<br>• In combination with random dispatching this strategy only applies for the vehicles which are called from a storage location.<br>• Randomly travelling vehicles which try to approach a source at which all buffer positions are already occupied, are rerouted by being sent to the alternative outlet of the switch. |

Table 1: Lessons learned during the implementation of D1

**Initialization process**

The initialization process is not of major importance for this dispatching strategy. The vehicle release process is started at the beginning of the simulation. The empty vehicles therefore start directly to travel randomly in the system and are spread across the whole system when the first load enters the system.

### 5.2.3 Static destination rule (D2)

**Idea**

This approach assumes that each vehicle has certain, but very limited knowledge of the transport network. Each vehicle knows the storage location that it initially left from and is constantly assigned to this storage location. The vehicles do not acquire additional knowledge. In this study the number of vehicles in the system is always equally distributed to the storage locations during the initialization process, i.e. the same number of vehicles is assigned to each storage location.

**Implementation**

Vehicles which completed a job are not actively searching for a new transport request. This is the most important feature of this strategy. Once the vehicles have delivered a load to a sink, they simply travel back to their standard storage location. On their way to the storage location they are not accessible for waiting loads and are not searching for work. If a vehicle arrives at a storage location and there are waiting loads at the corresponding source, the vehicle claims the load with the longest waiting time.

The claimed load is removed from the list of waiting loads. All empty vehicles always travel via the assigned storage location when this strategy is in use. They never approach the source directly after having completed a job. If a vehicle arrives at a storage location and no load is available at the corresponding source, it has to wait until it is claimed by an arriving load.

Arriving loads can only claim vehicles at the storage location corresponding to their arrival source. If no waiting vehicles are available, the load has to wait and will be claimed by the next vehicle which arrives at the storage location. The static destination rule only applies load-vehicle assignment to the processes between sources and storage locations. All vehicles at the sinks are processed according to the very simple empty vehicle positioning approach.

**Major lessons learned and changes to initial implementation**
Major problems were not experienced during the implementation.

**Initialization process**
No special initialization processes required.

### 5.2.4 Forecast dispatching (D3)

**Idea**
For this dispatching strategy the sinks in the system locally collect and analyze the information they can acquire from arriving vehicles (see design option 3 above). The acquired knowledge is used to periodically forecast the expected vehicle demand at the sources in the system and to estimate the fill level of each storage location. Based on these projections the vehicles are dispatched to the different locations.

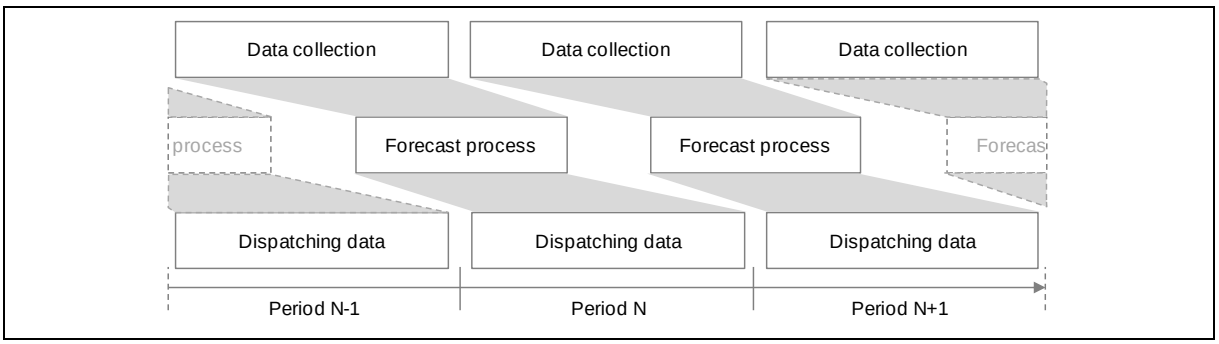



Figure 21: Conceptual design of the forecast dispatching procedure

**Implementation**
The overall dispatching strategy consists of 3 parallel processes:

- Data collection
- Forecast process
- Load-vehicle assignment and empty vehicle positioning based on projected dispatching data

Firstly, the data collection process needs to be considered. Each time a vehicle delivers a load, the sink reads the departure source of the arriving load.

Each sink $i$ ($0 \leq i \leq m$) maintains a list $\boldsymbol{Q}_i$ that contains the $j$ ($0 \leq j \leq 2m$) dispatching destinations $q_{ij}$ in the system that it knows about. Dispatching destinations are sources and storage locations. For the read source information it is checked if the source is already contained in $\boldsymbol{Q}_i$. If it is already known, the information is discarded. Otherwise the location is included in the list $\boldsymbol{Q}_i$ of known dispatching

destinations. Vehicles can only be sent to known locations. Therefore this procedure improves the knowledge that each sink has about the network with the number of arriving vehicles.

Based on the list of known dispatching destinations in the system each sink $i$ maintains a list $\boldsymbol{R}_i$ with the number of loads $r_{ij}$ that it received from each known source $j$ ($0 \leq j \leq m$) during a defined time period $N$. Each period is of length $t^N$. After the end of each period, the collected information is reset to 0.

In addition to the source information, the sink also checks if the arriving vehicle was called from a storage location before it picked up the delivered load. If this is the case, a second update procedure is started. Storage locations which are not known yet are stored and known ones are discarded in order to update $\boldsymbol{Q}_i$. In addition, the information about the fill level is read and used to afterwards locally estimate the current fill level of each known storage location. Those estimates $s_{ij}^{est}$ ($0 \leq j \leq m$) are maintained in a list $\boldsymbol{S}_i^{est}$ at each sink. Only vehicles which left from a storage location before picking up a load provide information about the storage location fill level. Vehicles which did not travel via a storage location or have been released from a storage location before picking up a load do not provide fill level information. In the first case, they simply do not have information about the fill level, and in the second case the vehicles might have been travelling randomly for a longer time and the information therefore would not be up to date. The arrival process is summarized in the flow chart below.

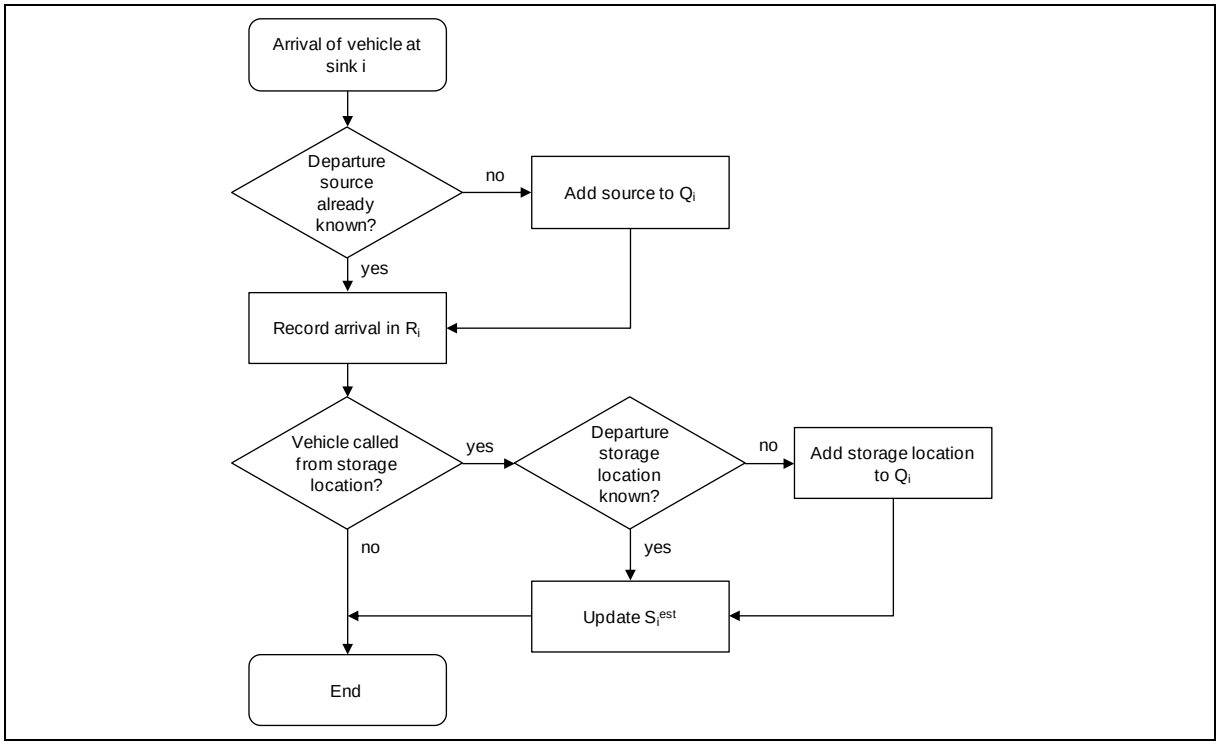



Figure 22: Arrival process of each vehicle at sink $i$ for D3

The forecasting process is the second part of the dispatching procedure. Based on the acquired knowledge exponential smoothening is applied at the end of each period $N$ to estimate at each sink $i$ the expected demand of vehicles $e_{ij}$ which each known source $j$ ($0 \leq j \leq m$) will have in the next period $N+1$. The amount $r_{ij}^{N}$ of vehicles which has been received from source $j$ during the last interval $N$ is used to update the forecast. The weighting factor $\varepsilon$ is defined on the interval [0, 1]. The forecast function is stated below and updates the list $\boldsymbol{E}_i$ of estimated demands:

$$e_{ij}^{N+1} = \varepsilon * e_{ij}^{N} + (1-\varepsilon) * r_{ij}^{N} \tag{5.1}$$

For any kind of estimation or forecast the latest information should be used. Regarding the fill level of a storage location the level which was reported by the last arrived vehicle can be considered the latest information. That is why there is no real forecasting process for the fill level. The current knowledge about the fill level is simply updated with each arriving vehicle (which was called from a storage location before pickup). The current estimation is the fill level which the last vehicle reported. If source $i$ receives the information from a vehicle $v$ which left from a storage location $j$, the current estimated fill level $s_{ij}^{est}(t)$ at time $t$ equals the fill level $s_j^v$ which the arriving vehicle provides:

$$s_{ij}^{est}(t) = s_j^v \tag{5.2}$$

There is only one exception. In case no vehicle from a known storage location has been received during the last forecast period $N$, the estimated fill level for this storage location is calculated by multiplying the latest information (from the period before) with the factor $\rho$ which is defined on the interval [0, $+\infty$] at the beginning of the next period $N+1$. An assumption is necessary if no information has been received from a certain storage location. No vehicle arrival means that there is either no demand and therefore no vehicle left from the storage location or there are no vehicles to serve an existing demand. It has been found that the latter case is usually covered by the source forecast (see above) and the former case is relevant to avoid an oversteering of the storage location, i.e. to limit the number of vehicles which is sent to a certain storage location from which currently no vehicles are leaving. Accordingly, in case no vehicle from storage location $j$ arrived at sink $i$ during the last period, the fill level $s_{ij}^{est}(t_N)$ which was known at the beginning $t_N$ of period $N$ is increased by multiplication with factor $\rho$ to calculate the fill level $s_{ij}^{est}(t_{N+1})$ for the beginning of the next period $t_{N+1}$:

$$s_{ij}^{est}(t_{N+1}) = s_{ij}^{est}(t_N) * (1+\rho) \tag{5.3}$$

This procedure only yields useful results when the current fill level is not 0. Therefore in the case where the current estimated fill level for a certain storage location is 0 at the end of a period $N$, the fill level estimation is set to 1 and updated based on this value. To avoid fill level estimations going to infinity, an upper border of 999,999 is set. Test runs have shown the performance impact of this upper border can be neglected. It should be kept in mind that each of the sinks has its own perspective on the system. The forecast process is executed concurrently at each sink $i$. The fill level estimation for the same storage location might be completely different at two sources.

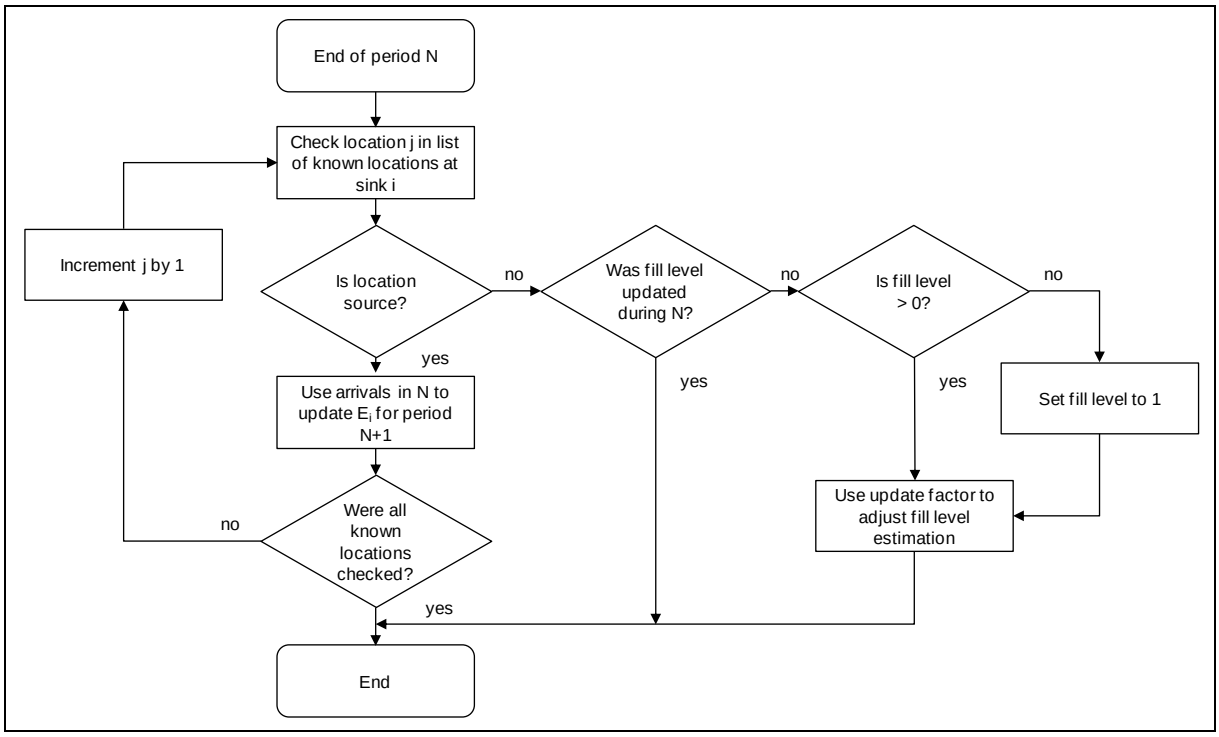



Figure 23: Forecast process at sink $i$ for D3

The forecasted data is the basis for the dispatching decision. Vehicles which just completed a job at a certain sink $i$ are sent to the known source $j$ with the highest estimated remaining demand $f_{ij}$ for the current period $N$. The remaining vehicle demand of each known source $j$ is calculated based on the estimated vehicle demand $e_{ij}^N$ for that period $N$ and the number of vehicles $g_{ij}^N$ which have already been sent to a specific source $j$ during this period. Thereby the remaining demand at source $j$ is reduced by subtracting 1 for each vehicle that has been dispatched to this source. The remaining demand and the number of sent vehicles are stored in the lists $\boldsymbol{F}_i$ and $\boldsymbol{G}_i$ at each sink.

$$f_{ij}\ (t) = e_{ij}^N - \ g_{ij}^N \tag{5.4}$$

If there is no source with remaining demand left for this period, the vehicle is sent to the storage location with the lowest estimated fill level. On its way to the storage location the vehicle is not available for load requests and does not look for jobs.

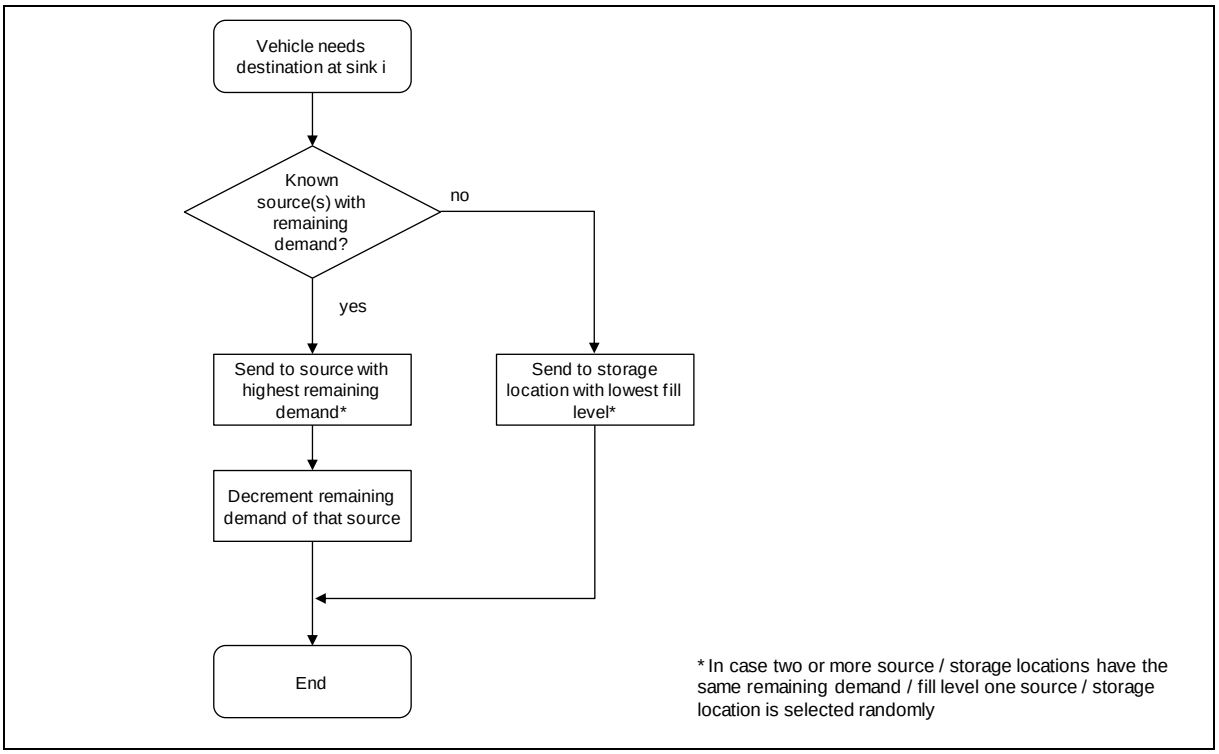



Figure 24: Dispatching process at sink $i$ for D3

As already mentioned, this dispatching rule has the effect that all sinks get to know the network step by step. Nevertheless, decision making is still based on local information. The sinks make decisions only based on their local perception. There is no complex decision making process which involves other nodes and thereby aggregates local information to global knowledge. No communication among the nodes is necessary.

Besides the dispatching process at the sinks, additional dispatching decisions need to be considered. A vehicle that arrives at a storage location can check the corresponding source for waiting loads. If there are no loads available, it has to wait. A vehicle which arrives at a source checks if there are loads waiting and otherwise simply waits for the arrival of a new load.

Arriving loads check if vehicles are available either directly at the source, i.e. are approaching the source and have not been claimed yet or at the storage location which corresponds to their current source location. If there are no vehicles available, the loads need to wait until a vehicle arrives.

Forecast dispatching limits the load-vehicle assignment to decisions at sources and storage locations. The sinks are responsible for empty vehicle positioning. Empty vehicle positioning does not only contain the selection of a storage location. It is also used to send vehicles directly to sources. But the source approach happens without explicit load assignment. The assignment process is not initiated before the vehicle physically arrives at the source.

**Major lessons learned and changes to initial implementation**

| | Problem | Implemented solution |
|---|---|---|
| 1 | • Similar to Problem 1 for random dispatching (see Chapter 5.2.2) | • Similar to Problem 1 for random dispatching (see Chapter 5.2.2) |
| 2 | • Similar to Problem 2 for random dispatching (see Chapter 5.2.2) | • Similar to Problem 2 for random dispatching (see Chapter 5.2.2) |
| 3 | • Too many vehicles are either sent to the sources or arrive there when travelling randomly. Due to low demand there might be tailbacks at certain sources. These temporarily block parts of the system and decrease its performance. | • An additional strategy for load-dependent source approach (see Chapter 5.4.2 below for a detailed description) is developed.<br>• Randomly travelling vehicles which try to approach a source at which all buffer positions are already occupied, are rerouted by being sent to the alternative outlet of the switch. |

Table 2: Lessons learned during the implementation of D3

**Initialization process**

The initialization process influences the achieved performance. It is important to enable a decent throughput level quickly after the start of the simulation. Otherwise the used termination criterion (see below in Chapter 6.2.4) cancels the simulation run. For the forecast-based dispatching strategy it has been found that the best solution is to start the vehicle release process directly at the beginning of the simulation. The vehicles start directly travelling in the system. Although they do not spread any knowledge as this is always limited to arrival information of laden vehicles at the sinks, this procedure makes sure that all vehicles are part of the release process once the first loads enter the system. The AutoMod implementation only includes vehicles in the release process that have already transported a load to make sure that a vehicle is available once the first demand occurs at a source and to spread reasonable network information. But according to the implementation it might happen that a certain vehicle is not part of the release process as it has not left its initial storage location at least once and therefore blocks it when there is no demand at the corresponding source. Randomly travelling vehicles will accumulate behind this waiting vehicle and are not available for transporting loads in the system. Forecast dispatching requires changing the standard implementation.

In the beginning of the simulation run a sink might only know about a source, but not about a storage location yet. In case the source does not have a demand anymore (remaining demand = 0), a vehicle is not dispatched to an explicit location, but continues to travel randomly. This is an extremely unlikely event.

### 5.2.5 Feedback-based dispatching (D4)

**Idea**

The development of this rule has been inspired by the AntNet algorithm which was introduced by DORIGO & DI CARO (1999). The AntNet algorithm has been derived from the initial version of the ant-based optimization algorithm and applies it to finding shortest paths in a network. Feedback-based dispatching uses the idea of stigmergy as the fourth design option for local decision making. As briefly introduced above (see Chapter 3.2), the AntNet algorithm is based on probability tables which are maintained at each switch. Artificial ants use those tables to decide which travelling direction to choose. Having arrived at their destination the ants use the overall travel time as a criterion to judge

the quality of the found path. They provide feedback about this path quality to all switches they passed on their route. Based on their feedback the probability tables are updated.

The AntNet algorithm was originally developed for routing. Dispatching applications have some characteristics which make modifications necessary:

- In contrast to routing, one optimal solution for dispatching cannot be found. The probability table at each switch needs to rather distribute the empty vehicles according to the current status of the network. As this status changes constantly, the tables need to be adjusted continuously.
- In material handling systems online decisions are required. Therefore it is impossible to use artificial ants to simulate the system and to determine an optimal solution. The vehicles in the system have to represent the ants of the algorithm directly (see discussion of this approach in Chapter 3.2).
- Besides positive feedback, penalties are required in the feedback functions, e.g. if a vehicle has been sent to a source where no load is waiting, a better solution for future vehicle distribution needs to be found.

Figure 25 summarizes how the bio-inspired dispatching algorithm works. It gives an overview of the approach. Afterwards the different steps of the algorithm will be explained in detail.

There is a probability table at each switch in the system. This table contains all known dispatching destinations, i.e. all sources and storage locations of the system that the switch is currently aware of. Each switch learns those locations from the vehicles which are passing by.

A vehicle which completes a job at a sink travels to the next switch. Arriving at that switch it selects one destination based on the probability table and continues its journey. This process is repeated each time the vehicle arrives at a switch until the vehicle arrives at a dispatching destination, i.e. a source or a storage location.

Once the vehicle leaves the reached dispatching destination, it provides feedback to all the switches which it has visited since its departure from the sink. The feedback process is based on the linear reward-penalty reinforcement scheme (see NOWE ET AL. 2006). It has the advantage that feedback is not limited to positive figures, but may also be used to decrease the probability value of the chosen destination. The feedback value $c$ is calculated based on the waiting time that the vehicle experiences at its dispatching destination. It is used to update the probability tables of all the switches along the route which the vehicle took since its departure from the sink. There are two different feedback functions (see figure and formulas below). If there is a reward, the probability for selecting the arrival destination based on the dispatching table is increased, in case of a penalty (negative $c$) it is decreased.

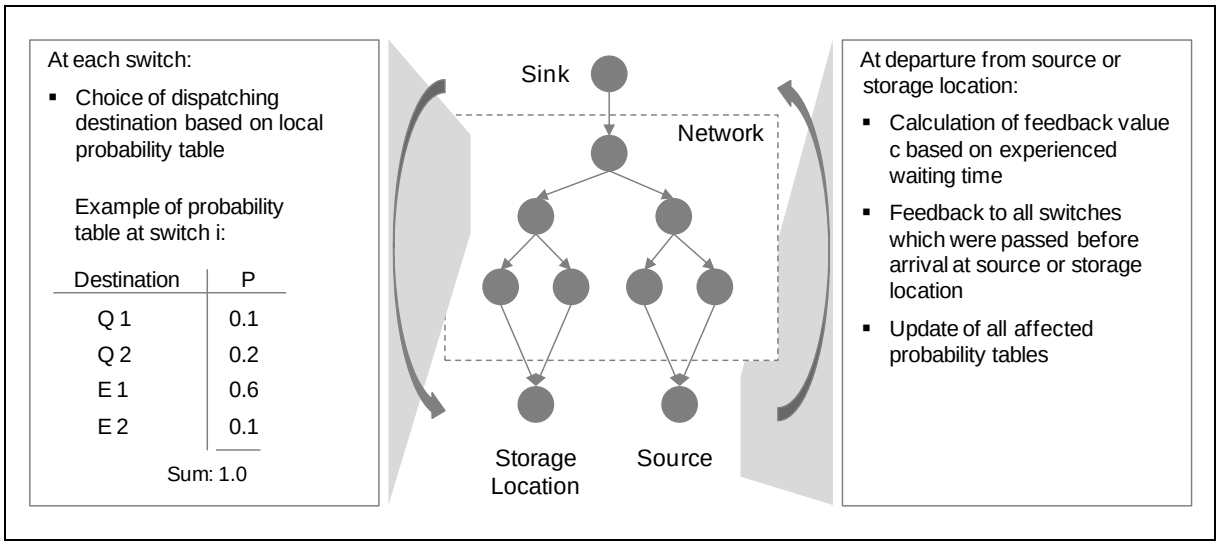



Figure 25: Feedback-based dispatching procedure

**Implementation**

The process for acquiring knowledge about the network is to a certain extent similar to the process which is used for forecast dispatching. Here, not the sinks but the switches are the relevant entities for the information collection as they are the points where the dispatching decisions are made. Therefore each switch $i$ ($0 \leq i \leq o$) maintains a list $\boldsymbol{Q}_i$ of relevant dispatching destination $q_{ij}$ ($0 \leq k \leq 2m$) that it knows about. Similar to forecast dispatching the relevant destinations can be sources or storage locations. The destination information is spread by the vehicles. In the initial version of the algorithm only unladen vehicles provided information after they had completed a transport. This information contained the location where the vehicle had picked up its last load and the storage locations it potentially departed from (similar to forecast dispatching). But first trials showed that laden vehicles should also spread information. They make the origins of their current loads available. In addition, vehicles that are called from a storage location spread the information about their designated pickup location. The switches read this information from the passing vehicles. If the location where the vehicle came from is already known, the information is neglected. Otherwise the location is included in the list $\boldsymbol{Q}_i$ of known dispatching destinations. Vehicles can only be sent to known locations. Therefore this procedure improves the knowledge that each sink has with the number of passing vehicles.

Based on the list of known dispatching locations a probability table $\boldsymbol{P}_i$ is maintained at each switch $i$. This table assigns a probability $p_{ij}$ to each known dispatching location $j$ ($0 \leq j \leq 2m$). We will refer to this table as dispatching table. The value $p_{ij}$ represents the probability that a vehicle which arrives at switch $i$ selects the known dispatch location $j$ as its next destination. The choice of the next dispatching destination happens locally at each divert. By using the probability table it is made sure that all dispatching locations are at least chosen with a small probability to get new information about their latest status. The sum of all probabilities in the dispatch table at each switch is always 1.

By using the dispatching table we create a vehicle-initiated dispatching rule which influences the vehicles' destinations while they are moving. The assigned dispatching destination is the input parameter for the routing algorithm. The destination might be changed several times as it is reconsidered each time a vehicle arrives at a switch. The number of switches at which the dispatching destination can be changed is defined by the parameter $b$, i.e. it can be varied between 1 and + ∞.

In addition to the vehicle-initiated dispatching rule at each switch, the same rules for dispatching activities at sources and storage locations as for D3 are in place. Once a vehicle arrives at a source or a storage location, it searches for waiting loads either directly at the source or the source which corresponds to the storage location at which it has just arrived. During its journey from sink to dispatching destination the vehicle is not available for loads and does not look for work.

Arriving loads on the other hand have the opportunity to search for idle vehicles either directly at the source or at the corresponding storage location. Vehicles and loads have to wait in case they can currently not find an entity to assign.

For the differentiation of load-vehicle assignment and empty vehicle positioning the same statements as for forecast dispatching mainly apply. Load-vehicle assignment happens at sources and storage locations. Empty vehicle positioning can refer to choosing a source or a storage location. But in contrast to forecast dispatching, the feedback-based algorithm does not make this decision at the sink, but repetitively at each switch. Vehicles approach sources and storage locations without direct load assignment. The process starts once a vehicle arrives at a source.

The dispatching table has to be updated continuously based on the network status. There are two different events which trigger an update. The first is learning a new dispatching location, i.e. adding a new location to $\boldsymbol{Q}_i$. The second is a feedback process which is executed once a vehicle leaves from its dispatching destination.

When a new dispatching location is discovered, the dispatching table is reset. This means that the same probability $p_0$ is assigned to each known dispatch location $j$.The probability $p_0$ is simply the reciprocal of the number of known dispatching destinations at switch $i$. This means the overall probability of 1 is equally distributed to all known locations, no matter which value they had before. As no information about the new destination is available, this approach assumes that an equal distribution is the best estimate to make. Every other approach would require the definition of a probability for the new location. This additional initialization parameter is to be avoided.

The second trigger to update the dispatching table is the departure of a vehicle from the source or storage location that it has been dispatched to. At the time of departure it can be calculated how long the vehicle had to wait at its dispatching destination. Either it had to wait for a load when it was sent to a source or it had to wait to be called in case it was dispatched to a storage location. In both cases the waiting time $t_v^{wait}$ of this specific vehicle $v$ is used to calculate the feedback value $c$ at sources and storage locations. The specific functions are given below. $c_{max}$ is the maximum feedback value which is defined on the interval ]0, 1]. Some additional parameters need to be specified:

- $t_j^{StWait}$ are the standard waiting times which are required for all operations at a source or storage location $j$ ($0 \leq j \leq 2m$). The standard waiting times contain “empty load” set down and “full/empty load” pickup time.
- $t_j^{AvWait}$ is the average waiting time at a storage location $j$ ($0 \leq j \leq m$) which is updated continuously after the departure of each vehicle.
- $t_{thr}$ is a threshold value for the waiting time at the sources.

The feedback function for a departing vehicle that waited for $t_v^{wait}$ at a source $j$ is given by:

$$c = \begin{cases} c_{max} & t_v^{wait} \leq t_j^{StWait} \\ c_{max} - \frac{c_{max}}{t_{thr}} t_v^{wait} & t_j^{StWait} < t_v^{wait} < t_{thr} \\ 0 & t_v^{wait} = t_{thr} \\ c_{max} \frac{t_{thr}}{t_v^{wait}} - c_{max} & t_{thr} < t_v^{wait} \end{cases} \tag{5.5}$$

The feedback function for a departing vehicle that waited for $t_v^{wait}$ at a storage location $j$ is given by:

$$c = \begin{cases} -c_{max} & t_v^{wait} \leq t_j^{StWait} \\ \frac{t_v^{wait\,2}}{\frac{1}{c_{max}} t_j^{AvWait\,2}} & t_j^{StWait} < t_v^{wait} < t_j^{AvWait} \\ 0 & t_v^{wait} = t_j^{AvWait} \\ c_{max} \frac{t_j^{AvWait}}{t_v^{wait}} - c & t_j^{AvWait} < t_v^{wait} \end{cases} \tag{5.6}$$

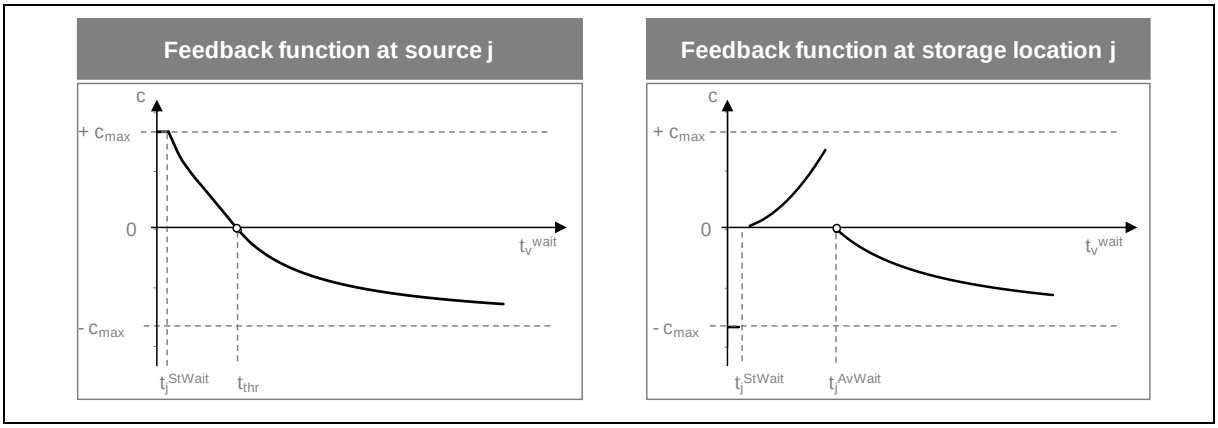



Figure 26: Functions for calculating feedback value $c$

The rational for using these functions was to distinguish which waiting times at sources and storage locations would be appreciated and which would not be desirable.

At a source, achieving the standard waiting time would mean that a load could directly be picked up upon vehicle arrival. No other vehicles interfered with the vehicle which was providing feedback and a load was available at the source. Above the standard waiting time the reward decreases until a certain threshold value is reached. This decrease can be necessary for two reasons. Either the vehicle had to wait because no load was available or because there were already other vehicles waiting in the queue. The threshold value equals the time a vehicle needs for load pickup in case all buffer positions on the loop siding in front of a source are occupied when the vehicle arrives and there are enough loads to be

picked up. The threshold value can therefore be estimated based on the physical system characteristics. Above the threshold it can be expected that the vehicle went to the source without enough loads being available and therefore a penalty is applied. The vehicle should either have been sent to a storage location or to another source.

For the storage location similar arguments are relevant. If the vehicle did not have to wait longer than the standard waiting time, there was no use going to the storage location. It should rather have moved directly to the corresponding source as a load was directly available when the vehicle arrived at the storage location. Waiting times above the standard waiting time need to be rewarded. In this case – based on local knowledge – it made sense to send the vehicle to the storage location. There was no load available at the corresponding source and the vehicle had to wait.

However, there might have been loads available at other sources in the network. Looking only at local information, this effect can only be considered indirectly. In case the waiting times become too long, it is necessary to propagate the information that there are currently no additional vehicles required at a certain storage location because only a few loads are available at the corresponding source. In contrast to the source, the threshold value cannot be estimated based on the technical system characteristics. To avoid the definition of another parameter, we have decided to use the average waiting time at the storage location for the decisions.

For calculating the average waiting time $t_j^{AvWait}$ at storage location $j$ ($0 \leq j \leq m$) we do not use the standard average formula but the mathematical model as it is used for example by DHILLION & VAN MIEGHEM (2007). The average waiting time is updated by the i$^{th}$ arriving vehicle according to:

$$t_j^{AvWait}(i) = t_j^{AvWait}(i-1) + \tau(t_v^{Wait} - t_j^{AvWait}(i-1)) \tag{5.7}$$

In this formula $\tau$ is a weighting factor that allows defining how reactive the calculation behaves with respect to the latest waiting time.

Several other feedback/reward functions have been tested, especially some with linear shapes. But although various alternatives could be applied, none of them could achieve the performance of the two feedback functions stated above.

Once the feedback value $c$ has been calculated, it is used to update the dispatching tables of all switches along the route which the vehicle took from sink to its dispatching destination. If the vehicle has travelled in loops and visited some of the switches several times, feedback is only provided once. In case the number of dispatching decisions was limited, i.e. a new dispatching destination has only been assigned at the first $b$ decision points after departure from the sink, feedback is nevertheless provided to all switches along the route, i.e. also to switches where no new dispatching locations have been assigned. Although no dispatching decision was made, the provided feedback is valuable knowledge for the visited nodes to update their dispatching tables and make a decision based on the latest network status.

Depending on the value of $c$ being either positive or negative, there are two different update procedures. The probability $p_{id}$ is assigned to the location where the vehicle is leaving from and the subscript $x$ stands for all other ($j$-1) probabilities. For each switch $i$ with $j$ known dispatch locations

the updated probabilities $p'$ are calculated based on the current probabilities $p$ as follows if $c$ is positive:

$$p'_{id} = p_{id} + c(1 - p_{id}) \tag{5.8}$$

$$p'_{ix} = p_{ix} - cp_{ix} \tag{5.9}$$

And with all the above said still being valid, the formulas for negative $c$ are:

$$p'_{id} = p_{id} - |c| p_{id} \tag{5.10}$$

$$p'_{ix} = \frac{|c|}{j-1} + (1 - |c|)p_{ix} \tag{5.11}$$

Generally, all vehicles are included in the feedback procedures at sources and storage locations. One exception are vehicles which are called from a storage location. As they were not directly dispatched to a source, they do not provide feedback when they leave the source. They already have provided feedback when leaving the storage location.

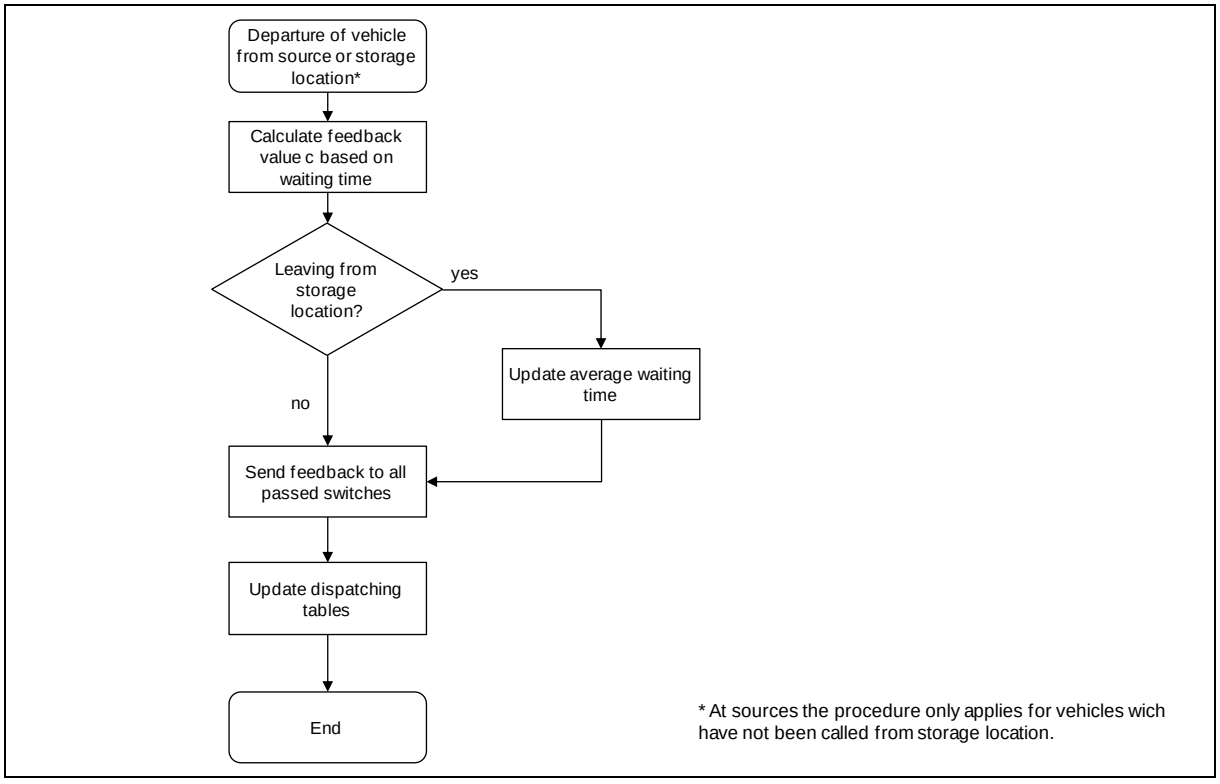



Figure 27: Feedback process for feedback-based dispatching

Another aspect which should be highlighted in this context is the feedback procedure for released vehicles. They are released because their waiting time is considered to be too long. Consequentially, they should provide the maximum feedback $-c_{max}$ when they are leaving from source or storage location. According to the functions shown above, this happens automatically at storage locations as the waiting time of released vehicles is set to 0. For the sources the function is extended by a release condition that leads to the desired behavior.

**Major lessons learned and changes to initial implementation**

| | Problem | Implemented solution |
|---|---|---|
| **1** | • Similar to Problem 1 for random dispatching (see Chapter 5.2.2) | • Similar to Problem 1 for random dispatching (see Chapter 5.2.2) |
| **2** | • Similar to Problem 2 for random dispatching (see Chapter 5.2.2) | • Similar to Problem 2 for random dispatching (see Chapter 5.2.2) |
| **3** | • Similar to Problem 3 for forecast dispatching (see Chapter 5.2.4) | • Similar to Problem 3 for forecast dispatching (see Chapter 5.2.4) |
| **4** | • Certain locations (sources or storage locations) receive high feedback values and vehicles travel to those frequently. The effect reinforces itself and only one source or storage location in the system is used as dispatching destination. The behavior is induced by vehicles which have visited a certain switch more than once and give multiple feedback. | • The vehicles store each visited switch only once, although they might have passed it several times. Thereby feedback is only given once to each switch on the route. |
| **5** | • Not all sources in the system are served initially. The system does not reach the required throughput. Especially in more complex systems, it can be observed that it takes a long time until all diverts know about all the dispatching destinations in the system. It could happen that sources with low demand were not known at the switches which are located in front of them. Therefore these switches would never send vehicles to those locations. | • As described above, the vehicles do not only spread network knowledge when they are travelling unladen, but also during the retrieve and delivery process. |

Table 3: Lessons learned during the implementation of D4

**Initialization process**

Similar to the forecast dispatching rule, the vehicle initialization process is important for our simulation experiments. In the case of the forecast rule, the release process starts immediately at the beginning of the simulation. This approach also delivers fairly good results for small systems when using the feedback dispatching rule. However, bigger systems (see layout scenarios in Chapter 6.1.1) cannot deal with this kind of initialization. Although blockages might occur, not starting the release process at the beginning of the simulation makes sure that vehicles are available when the first demand occurs at a certain source and that vehicles are not distributed all over the system. For bigger systems this is the only initialization strategy which leads to a quick and stable increase of the system throughput. It makes sure that the network knowledge is spread efficiently in the system.

The vehicles provide the first feedback when they leave a storage location for the second time. The first time they are in the storage location due to the initialization procedure and feedback would not be meaningful. The standard waiting times for sources and storage locations are recorded when the first vehicle travels these paths.

### 5.2.6 Central dispatching (D5)

**Idea**

The central dispatching rule shall be used as a benchmark for assessing the performance of the decentral rules. It uses global information. This means the whole network, all vehicle positions, available loads and storage locations with their current fill levels are known when the dispatching decision is made.

**Implementation**

Generally, the availability of all current network information would allow the calculation of the optimal solution for the load-vehicle assignment. However, it has been shown above that the calculation of the optimal solution is on the one hand computationally expensive. On the other hand, the found solution would only have an extremely short validity as the arrival of new loads immediately changes the system status and a recalculation would be inevitable. As continuous recalculations are not feasible in a real-world scenario, we follow the scientific community and apply a heuristic dispatching rule to our system. Among the commonly known rules, we select the shortest-travel-time-first and closest-vehicle-first approach which can be found in the AGV literature (see e.g. VIS 2006). As the literature review revealed, there is no dispatching rule that is superior. Rather a case-specific selection is required. As our goal is not an optimization of central dispatching, we choose two single attribute rules due to their simplicity and because they are commonly accepted. In some studies these distance-based rules were able to outperform other dispatching rules (see e.g. DE KOSTER ET AL. 2004). However, the rules also have disadvantages which became visible during our simulation experiments. They are responsible for load-vehicle assignment. In addition, we need a rule for empty vehicle positioning. As shown above, only the approach of HALLENBORG (2007) explicitly deals with the distribution of vehicles to several storage locations in the system. As this is one of the core questions of this thesis, we implement his approach. It has the additional advantage that almost no parameterization is required. The implementation is the first combination of load-vehicle assignment and empty vehicle positioning rules in systems with several storage locations that we are aware of.

A vehicle which completes a job at a sink has access to global network information. Therefore it checks all sources in the system for available loads. If there are loads available at more than one source, it claims the closest load (shortest-travel-time-first). As we use static routing (which will be introduced below in Chapter 5.3.2), the physical distance is used as decision criterion. Once a vehicle has claimed a load, this load is removed from the list of waiting loads. In contrast to the dispatching rules above (D1, D3, D4), there is a fixed assignment between load and vehicle. This means the vehicle definitely travels to the assigned pickup location. A re-assignment is not possible and the assignment process is not postponed to the arrival of the vehicle at the source. This is possible because based on the distance-based decision rule the vehicle will be the first empty vehicle to arrive at the source. In the case where two sources with available loads would have the same distance to the current vehicle position, one load would be selected randomly. This case does not occur in the test layout and is therefore only of theoretical relevance.

If no loads are currently available in the system, the vehicle is sent to a storage location. Among all storage locations in the system, one is chosen according to the procedure of HALLENBORG (2007). A detailed description of the selection process is given in the literature review (see Chapter 3.1). The three basic steps are briefly repeated:

1. Based on the current fill level of each storage location $i$ in the system its priority $z_i$ is calculated.
2. The priority of each storage location $i$ is multiplied with the distance $d_{vi}$ between the current vehicle position and the storage location $i$ to calculate the decision criterion $\omega_{vi}$.
3. The storage location with the smallest $\omega_{vi}$ is selected.

The multiplication in step 2 explains the shape of the underlying function for calculating the priority $z_i$ (see Figure 15, page 35). The function is used to disproportionally decrease the decision criterion for small fill levels. On the other hand high fill levels are increased disproportionally. Thereby more attention is given to storage locations with low fill levels.

In the case where two or more storage locations have the same and lowest decision criterion, one of those storage locations is selected randomly. This case is especially relevant, when more than one storage location has a fill level of 0. Vehicles which are travelling to their assigned storage location are available for dispatching when a new load enters the system. This means the vehicle status might change and instead of going to park, a vehicle might turn to a pickup location.

When a new load arrives in the system, it can search for vehicles in the complete network. The lists of approaching vehicles at all control points are considered. Among all available vehicles the load selects the vehicle which is closest to its current source location and available for dispatching, i.e. has not already been assigned to another load. The distance is calculated based on the physical distances in the system. As a simplification, the exact position of the vehicle is not considered for the calculation but the distance between the source and the control point which the vehicle is currently approaching. The vehicles between two control points are treated according to the FCFS principle. If the load finds a vehicle, it claims the vehicle. The vehicle is assigned to the load and not available for dispatching anymore. It proceeds to pick up the load.

Once a vehicle arrives at a storage location without having been assigned to a load during its journey, it checks all sources in the system for available loads. Among the available loads it selects the closest load, claims it and picks it up. Once the vehicle claims the load, the former is not available for dispatching anymore.

Looking at the concepts of load-vehicle assignment and empty vehicle positioning, there are differences compared to the dispatching strategies which have been discussed so far. The major distinction is the fixed load-vehicle assignment which might be triggered by the load or the vehicle. Except D2, all other strategies postpone the fixed assignment until a vehicle arrives at the source. In addition, empty vehicle positioning can be clearly defined and distinguished from load-vehicle assignment and other tasks for the central dispatching rule. The concept covers the choice of one specific storage location in this context. The vehicle is sent to park when no loads are available in the system. Afterwards the vehicle status can only be changed by a load-initiated rule or the arrival of the vehicle at the storage location. For the central rule the share of the activities which vehicles spend on retrieving loads and going to park could be precisely distinguished.

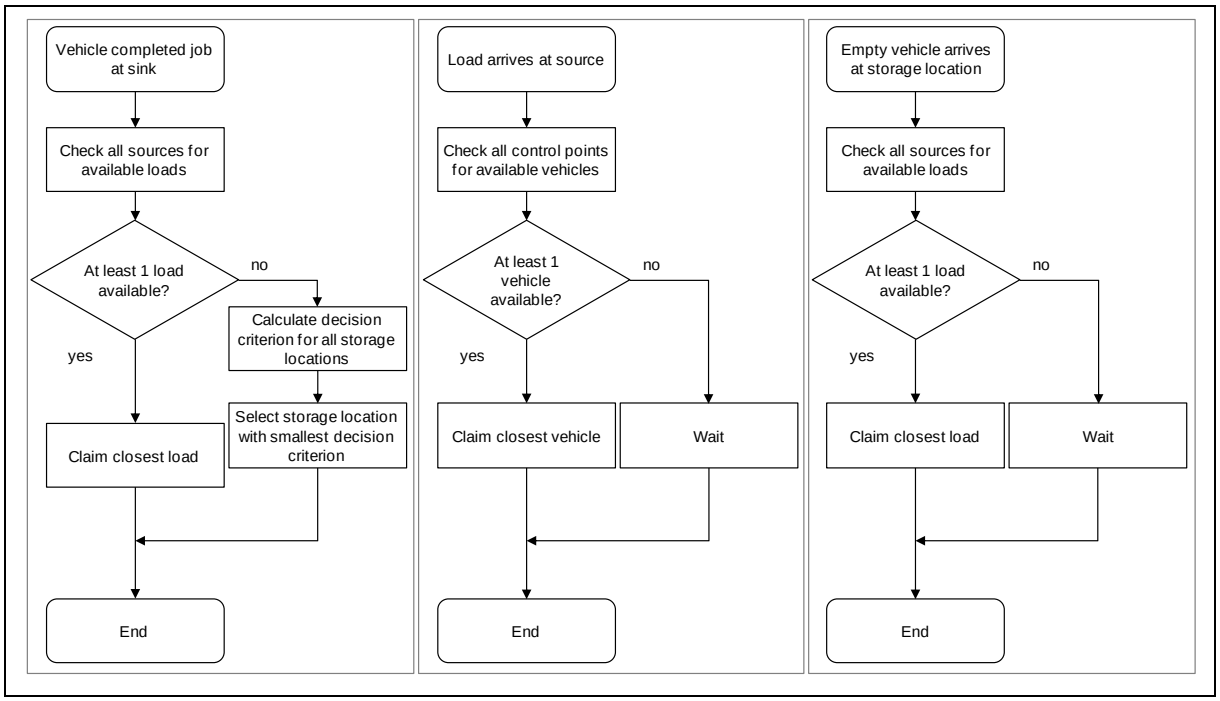



Figure 28: Dispatching decisions in a centrally controlled system

The implementation uses a "virtual" fill level for selecting one of the storage locations in the system. The calculation of the fill level not only takes vehicles into account that are waiting at a specific storage location but also those which are currently approaching that location. This means as soon as a vehicle has been sent to a storage location it is included in the calculation of the fill level, although it might not have physically arrived at the storage location yet. Vehicles which were supposed to go to park but are re-assigned to a load during their journey are subtracted from the fill level. Vehicles which are still waiting to leave a storage location but have already been assigned to a load are not considered for the calculation of the fill level any longer. The delayed departure is caused by the load-dependent source approach (see Chapter 5.4.2). In systems with more vehicles than actually required for fulfilling the throughput requirements, i.e. a low utilization of the vehicles, it may therefore happen that the physical fill level exceeds the maximum storage location capacity. This effect has been discovered during the simulation experiments. HALLENBORG (2007) does not explicitly mention how the fill level is calculated. But to our minds only the expected fill level, i.e. the virtual fill level, is a reasonable mathematical quantity. The mentioned effect only occurs in systems which could be run efficiently with fewer vehicles and it can therefore be expected that it would not occur in a real-world scenario. Vehicles are expensive and a real-world application would use as few vehicles as possible.

In the experimental setup only the maximum storage location capacity $s_i^{max}$ needs to be specified. It is set to the number of vehicles that is assigned to each storage location during the initialization process. As the vehicles are initially distributed equally to the storage locations, the latter all have the same defined maximum storage capacity.

**Major lessons learned and changes to initial implementation**
Major problems were not experienced during the implementation.

**Initialization process**
No special initialization processes were required.

The table below summarizes the different dispatching strategies with respect to the triggered actions.

| | **Load-initiated dispatching** | **Vehicle-initiated dispatching** | | |
|---|---|---|---|---|
| | Action triggered by load arrival | Action triggered when vehicle finishes delivery job at sink | Action in case empty vehicles at the sink does not find new load | Action triggered when empty vehicle arrives at storage location |
| **D5** | • Claim vehicle with the shortest travel time to the load (if vehicle is available) | • Claim closest load (if load is available) | • Select one of the available storage locations based on their fill level and the travel time | • Claim waiting load with shortest travel time from current vehicle location (if load is waiting) |
| **D2** | • Claim waiting vehicle at storage location next to the source (if vehicle is available) | • Travel to the assigned storage location (fixed destination) | • - | • Claim load at corresponding source (if load is waiting) |
| **D4** | • Claim vehicle that waits at the source or call waiting vehicle from storage location next to the source (if vehicle is available) | • Continue to next switch to get new dispatching destination assigned | • - | • Check for waiting loads at the corresponding source and proceed to pickup if load is available<br>• (Vehicles that arrive at a source check it for waiting loads) |
| **D3** | | • Travel to source with the highest remaining demand forecast for the current period. If no source has demand, select storage location with lowest estimated fill level | | |
| **D1** | | • Continue travelling randomly in the system until a storage location is reached | | |

Table 4: Summary of triggers and actions for dispatching strategies

Looking at the table above, a major difference between decentral and central dispatching strategies is that the former do not work properly with fixed load-vehicle assignment. This process has to be postponed until the vehicles physically arrive at a source. In contrast to central dispatching, empty vehicle positioning is not limited to assigning storage locations in decental systems. It rather has to support the whole journey of a vehicle from the departure at a sink to the next arrival at a source. In addition, special procedures like the release process are required.

Concluding, all the implemented dispatching rules have to be assessed regarding their potential to fulfill the requirement of using local information only. It is obvious that strategy D1 only uses local information. D2 requires the initial assignment of vehicles to storage locations (or vice versa). With this knowledge the dispatching processes can afterwards be limited to local decision making. D3 collects the information which is spread by the delivering vehicles to get an understanding of the overall network. But there are no processes in place to actively collect this information by inter-node interaction in the system. Therefore the local information requirement is still fulfilled. It is to a certain extent violated by the feedback procedure of D4. The feedback process needs communication between

the vehicle and the nodes which have been visited. However, none of the decision points really knows the network. They acquire knowledge from the vehicles which spread the network information and get temporary updates. But as in case of D3, there is no active procedure for information collection or global information exchange. The communication effort seems controllable as not all nodes communicate with each other. Strategy 5 has global knowledge of the whole system.

Before the next section takes a look at the implemented routing strategies, the figure below summarizes this evaluation of the dispatching strategies with respect to the required information quality.

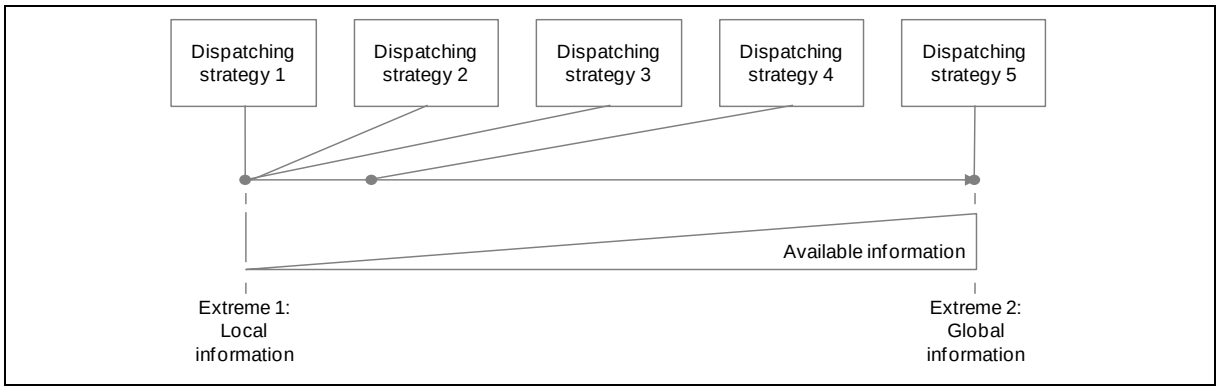



Figure 29: Evaluation of required information for dispatching strategies

## 5.3 Routing

Following the dispatching strategies, the implemented routing procedures are now explained. As we have argued above, the best route from source to sink can only be found reliably when global topology information is used. Using only local knowledge, routing cannot achieve more than simply forwarding vehicles randomly. The respective approaches for a very simple but inefficient routing and a more sophisticated algorithm are described below.

### 5.3.1 Random routing (R1)

The idea of this strategy is to reduce the used information for making routing decisions to a minimum. At each switch it is simply chosen randomly which of the available directions to take. There is no predefined path from source to sink and no routing table.

Two versions of this random routing rule are available. The first one uses random routing for laden and unladen vehicles. This would result in purely stochastic transport processes, a worst case scenario. Having a major focus on dispatching, this thesis uses the second version of the random rule. It only applies a random route choice for empty vehicles. This strategy is a prerequisite for the application of dispatching rule D1. In addition, random routing is also required for all vehicle release processes that are implemented for D3 and D4.

### 5.3.2 Central static routing (R2)

The central routing rule is based on complete network information. Using the Dijkstra algorithm, shortest paths are calculated for each source-sink combination. As explained in Chapter 4.2, the shortest path information is stored in a routing table which states for each switch which direction to take when approaching a certain destination.

The algorithm is implemented in a static version. The routing tables are only computed at the beginning of the simulation run and not updated afterwards. This approach serves the purpose of isolating the impact of the different dispatching rules from routing effects.

For this study it is irrelevant if the routing algorithm is truly central. It could also be implemented as a distributed or a decentral control strategy. It is simply assumed that there is some kind of algorithm in place which can provide the routing tables. The evaluation of the efficiency impact of the different implementation strategies is not in the scope of this thesis and would need to be examined in a separate study. We are interested in the characteristics of the dispatching strategies. Although we know that there are interdependencies between both strategy types, we treat the routing algorithm as a prerequisite for dispatching.

The static routing tables are used for laden and unladen vehicles which are dispatched according to D2, D3, D4 and D5. One exception has already been mentioned. Released empty vehicles of the strategies D3 and D4 require random routing.

## 5.4 Intersection control

### 5.4.1 Merge control (I1)

Merge control is required to enable an efficient processing of the vehicles which arrive at a merge at the same time. In addition, it helps to prevent deadlocks. As already explained, there are rules available which work locally. For simplicity, we have selected a FCFS logic.

If several vehicles arrive at a merge at the same time, they are allowed to pass it according to the time of their arrival. We use counters and the logic of three control points at each merge to implement the AutoMod functionality. The process could be thought of as maintaining a list of arriving vehicles at a merge. Each arriving vehicle is appended to this list. Once a vehicle drives through the merge zone, it is removed from the list of waiting vehicles. The first vehicle on the updated list is now allowed to pass the merge. This control strategy is applied to all analyzed scenarios.

### 5.4.2 Switch control (I2/3)

During the development process of the dispatching rules and while the different system configurations were being tested, it was recognized that besides the routing decisions additional divert control functionalities are required for two reasons:

- Tailbacks at sources and sinks – Whenever too many vehicles try to approach a station that is located in a loop siding, the resulting queue might block the major driveway. This problem could be experienced for all dispatching strategies except D2. The reason might either be demand peaks or oversteering of control strategies. In some cases randomly travelling vehicles can have the same impact. Although the major problems were experienced at sources, the effects could also be observed at sinks.
- Deadlocks – Due to the loop-based structure of the overall layout, it might happen that one of these loops gets completely filled with vehicles. In the worst case the vehicles block each other. No vehicle is able to leave or enter the loop. The system performance deteriorates.

For both problems suitable switch control strategies have been developed. These strategies use the underlying routing table but make decisions based on local information only.

**Switch control strategy 1: Load-dependent source and sink approach (I2)**
This rule was developed to mitigate the first of the two problems. The next sections explain its functionality. Sources are used as an exemplary case. The control strategy is similarly applied for all the sinks of the system.

It assumes that there are a number of buffer positions on the loop siding in front of each source. A vehicle which tries to approach a source can be denied access to the source if all buffer positions are occupied. In this case the switch at the entrance of the loop siding does not allow the vehicle to choose the direction which the routing table states. The decision is made based on local knowledge of the utilization of the buffer positions. According to the assumptions for the development of the control strategies, the utilization of the successor links is locally available information.

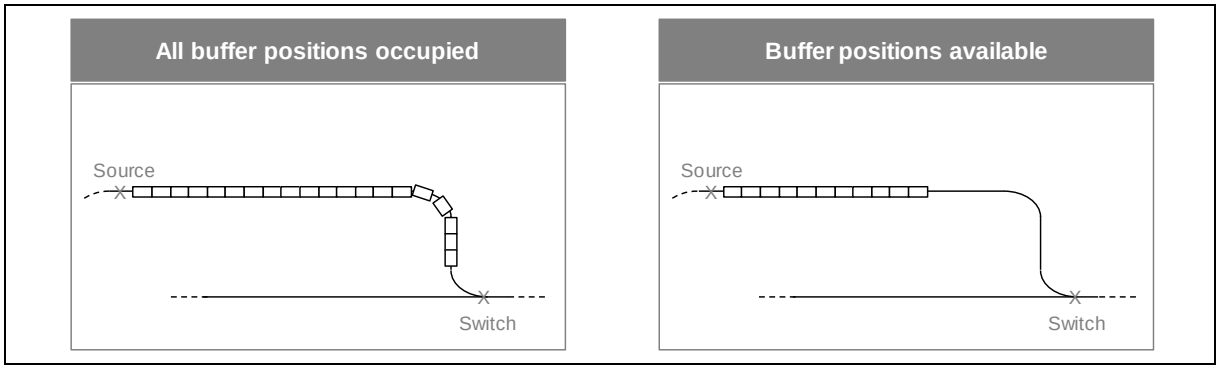



Figure 30: Accessibility of sources based on occupied buffer positions

Rejected vehicles keep their current source destination and continue travelling in the network. Based on the routing table at the next divert they are routed towards the same source again and will arrive there once they have completed a "waiting loop". The same process starts over until at least one buffer position is available and the vehicle is allowed to approach the source. Based on the test system layout, about 20 buffer positions are available at the sources. This figure is used as a threshold. Randomly travelling vehicles simply continue their journey when they are rejected. At the next divert they are routed randomly again, i.e. they will only approach the same source again by chance.

For sources, this load-dependent approach procedure is highly linked to the call-off process for D1, D3 and D4. These dispatching strategies can call vehicles from the corresponding source whenever they have a need. It might happen that vehicles are called from the storage location while there are no buffer positions available. A direct departure of the called vehicles would lead to their undesirable circulation. It is better to wait at the storage location. Therefore a second threshold value is introduced. Only if a maximum occupation of buffer positions is not exceeded, is a called vehicle allowed to leave from the storage location. The second threshold value needs to take into account that additional vehicles may arrive at the source between the time of the call and the arrival of the called vehicle at the source. A value of 14 occupied buffer positions has been found appropriate to call vehicles from the storage locations quickly enough while on the other hand preventing that called vehicles do not get access to the source which called them. Using this procedure it may happen that too many vehicles are called from the storage location because vehicles might arrive randomly at the source and serve loads which initiated a vehicle call. But as the arriving loads check the buffer positions in front of the source for vehicles before making a call, the effect only has a small impact. This has been verified by experiments. With the random influences at hand, an alternative approach is hard to develop anyway. The implementation of this procedure improved the results of D1, D3, D4 and D5 significantly. D5 does not use the call process and does not have to consider randomly travelling vehicles, but making

vehicles only leave a storage location when enough buffer positions at the source are available nevertheless has a positive performance impact. The influence of the load-dependent approach strategy for D2 is rather small as no vehicles approach the source directly. It is only during peak times that the limitation of vehicles leaving from the storage locations to approach the source can have advantages and mitigates the congestion risk.

**Switch control strategy 2: Load-dependent re-routing (I3)**
The second switch control strategy aims at mitigating the risk of deadlocks in the system. It was found that especially when many vehicles are in the system, they may block each other due to the loop-based layout of the system. Therefore a strategy for load-dependent re-routing was developed. It spreads the system load and thereby tries to prevent deadlocks. The control strategy is applied at all switches on the main driveways. The switches which are direct predecessors of sources, sinks and storage locations are exceptions. This prevents that the loop sidings are unnecessarily bothered by additional vehicles. They are bottlenecks anyway.

The basis for this rule is the routing table. Upon a vehicle arrival at a switch, the traffic load on the successor link that the vehicle is supposed to take according to the routing table is checked. The traffic load or utilization $u_p$ for path $p$ is calculated based on the patch length $l_p$ from switch $j$ to its successor, the vehicle length $l_v$ and the number of vehicles $v_p^{cur}$ which is currently travelling on this path.

$$u_p = \frac{v_p^{cur}}{\frac{l_p}{l_v}} \tag{5.12}$$

If the traffic load $u_p$ exceeds the threshold $u_{thr}$ which is defined on the interval [0, 1], the vehicle chooses the alternative direction. The re-routing is used for laden and unladen vehicles in combination with D1, D3 and D4. During the development process those strategies showed the highest deadlock risk as they require the most vehicle movement. They were the reason for developing this strategy. For D2 and D5 deadlocks are less likely, but might occur. In the test layouts (see Chapter 6.1.1) it is always possible to deviate from the shortest path as the system consists of loops and a vehicle always keeps the opportunity to reach its destination when choosing an alternative direction. This condition is a prerequisite for applying the control strategy.

This chapter introduced the implemented control strategies. The different approaches to dispatching, routing and intersection control have been illustrated. The next chapter discusses additional input parameters for the simulation experiments. It is stated how these parameters are combined with the control strategies to evaluate the latter with the help of simulation experiments.

# 6 Simulation methodology

This chapter describes the methodology for the simulation experiments which are used for performance measurement. Besides the input parameters, the relevant decisions for generating and analyzing the simulation output data are explained.

## 6.1 Input parameters

The developed test environment offers a wide variety of opportunities for testing and analyzing the impact of applying the different control strategies. There are several input parameters which need to be considered for assessing and comparing the performance of different system configurations:

- System size (scenarios of 1, 4, and 9 test loops)
- Input data sets (data set 1, data set 2)
- Input intensity (standard, elongation)
- Input complexity (number of source – sink combinations)
- Combination of control strategies (routing, dispatching, intersection control)
- Number of vehicles in the system
- Storage location capacity

The ranges for those input parameters are given in brackets in the list above (if applicable). As not all of them are self-explanatory, some additional remarks are required in the next sections.

### 6.1.1 Layout scenarios

The layouts scenarios are generated based on the simple test system which was introduced in Chapter 4.1. Mirrored duplicates of the single-loop test system are used to create more complex layouts. The first scenario consists of only one test system. Afterwards the system size is increased by adding 3 and 8 additional test system loops. It was initially planned to increase the system size further but it turned out that the simulation software runs into memory bottlenecks when the corresponding number of vehicles is brought into the system, e.g. for a scenario with 16 loops. Therefore the analysis is limited

to 9 test loops in the most complex system. It should be noted that a system of this size is smaller than many real-world intralogistics systems, e.g. baggage handling systems (see HAMMEL ET AL. 2008).

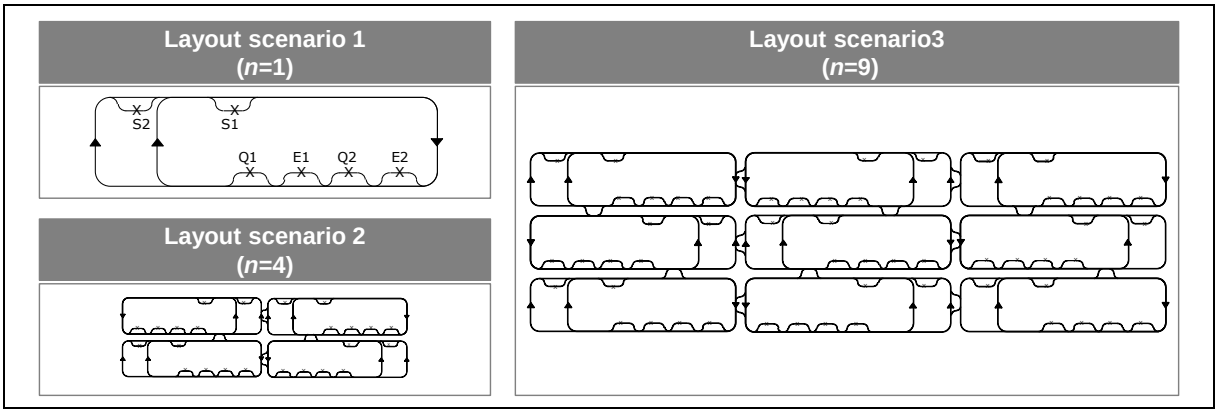



Figure 31: Layout scenarios

Based on the layout scenarios, the following numbers $m$ and $o$ of relevant system stations can be derived:

| Layout scenario | # sources ($m$) | # sinks ($m$) | # storage locations ($m$) | # switches ($o$) | # merges ($o$) |
|---|---|---|---|---|---|
| 1 | 2 | 2 | 2 | 7 | 7 |
| 2 | 8 | 8 | 8 | 36 | 36 |
| 3 | 18 | 18 | 18 | 87 | 87 |

Table 5: Number of relevant stations per layout scenario

### 6.1.2 Input data

For the simulation experiments two different real-world input data sets are used in this thesis:

- Data set 1: Transport orders recorded in a buffer replenishment (BR) application
- Data set 2: Transport orders recorded in a baggage handling (BH) system

Each line of each data set contains one transport request, i.e. a load transport that can be accomplished by one vehicle. Besides the source for load pickup, the inter-arrival time compared to the previous load is shown. Both data sets cover an eight day interval. No sink information could be recorded in the real-world scenarios. Therefore all handling units are randomly and equally distributed to the sinks in the test system. This is the major source of stochasticity. If the duration of a simulation run is longer than the eight days of data, the input data set is repeated. But while the same inter-arrival times are used again, the sinks are again generated randomly and therefore will show a different sequence.

In order to fit the available real-world data to the test system characteristics modifications were required. The raw data for data set 1 was recorded in a system of two high-bay warehouses which serve a buffer area. The data represents the transport requests for pickup at the warehouses and delivery to the buffer area. In a first step, the recorded transport requests which contained more than one handling unit were split into several lines. Each of them is now appropriate for pickup by one vehicle. The inter-arrival time between the split requests is 0. In addition, the raw data contains only two sources as there are only two warehouses in the real-world system. In the test system up to 18 sources are required for layout scenario 3. Therefore the raw data which was initially recorded for

more than 120 days is cut into eight day intervals. Each of the intervals gets assigned two sources. The overall data set is then created by combining nine of these eight day intervals to include up to 18 sources in the system. This approach has the disadvantage of averaging the data. Compared to duplicating one eight day interval nine times, it eliminates the peaks. But on the other hand, duplication would result in using only two different load profiles (those of source 1 and 2 in the raw data) for all 18 sources. In order to allow a high variability of load profiles, the duplication approach was rejected. Imbalances at different sources during operation are favored as they promise to be more demanding for the control strategies. The load profile for all 18 sources is shown in the figure below.

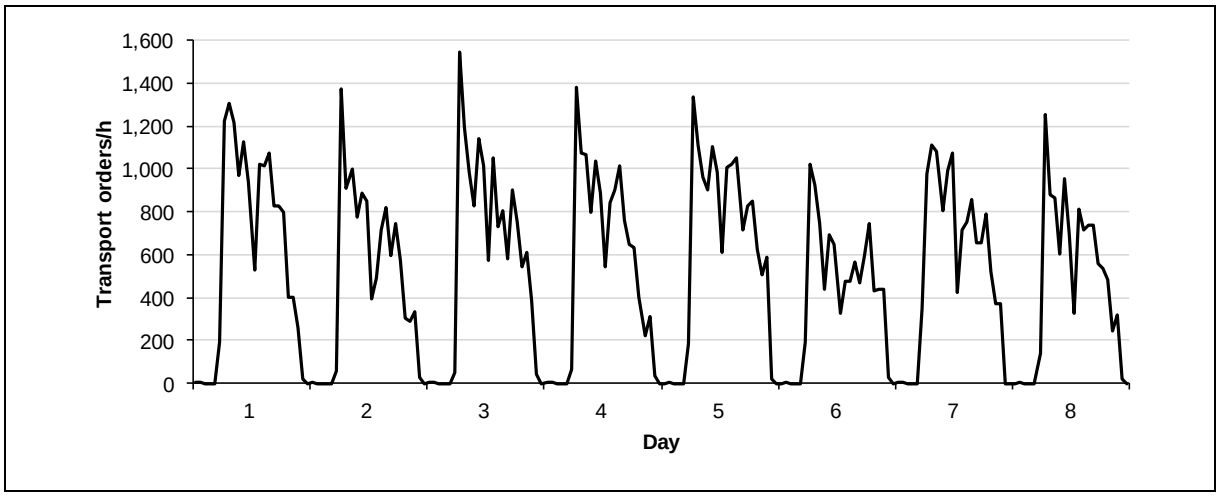



Figure 32: Load profile for input data set 1 (all 18 sources cumulatively)

The second data set also required modifications. The baggage handling data was recorded for an eight day interval. Besides the regular check-in data, transfer luggage and oversized luggage check-ins have been considered. The raw data therefore contained more than 30 sources. In order to fit the data to the test systems, 18 sources were selected randomly. Their overall load profile with the number of transport orders per hour is shown in the figure below. All of the mentioned source types are included in the data set and create high demand variability. In the airport environment each transport request automatically represents one load and therefore no splitting was required.

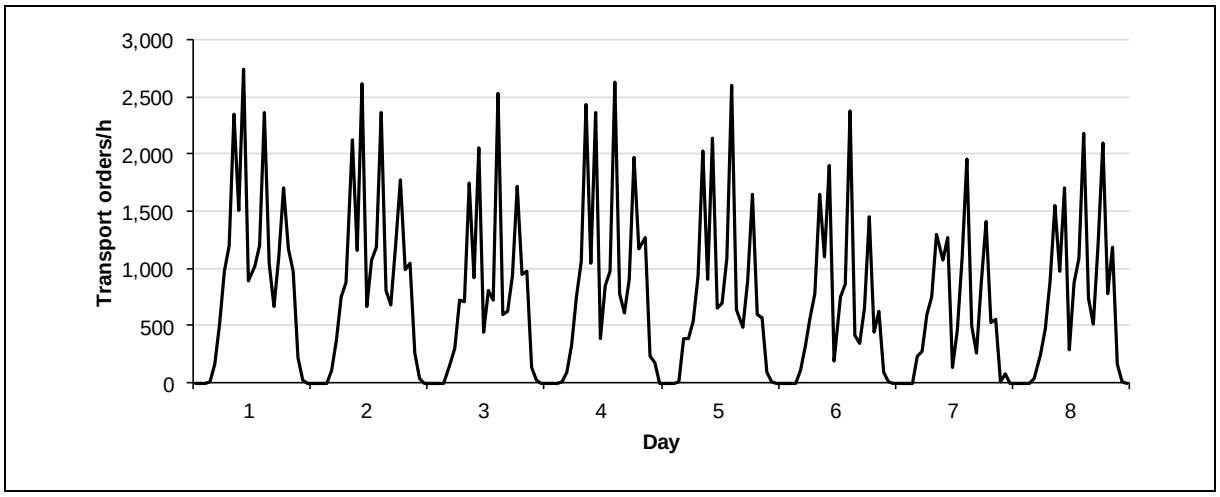



Figure 33: Load profile for input data set 2 (all 18 sources cumulatively)

Looking at the two load profiles it can be seen that the overall throughput of data set 2 is higher than for data set 1. In addition, data set 1 usually shows one extreme peak at the beginning of the day while there are usually several peaks during the day in the airport scenario.

Further analyses reveal that data set 2 shows a higher variability than data set 1 (see table below). The average of the mean throughput per source and hour shows a standard deviation of only 11.6 for data set 1 while it amounts to 39.1 for data set 2. The throughput requirements at the sources show bigger differences for data set 2 than for data set 1. The demand imbalances are higher.

Data set 1 – Buffer replenishment [transport orders / hour]

| Source | Ø | σ |
|---|---|---|
| 1 | 27.6 | 37.7 |
| 2 | 39.0 | 40,5 |
| 3 | 21.3 | 33.6 |
| 4 | 29.7 | 38.7 |
| 5 | 16.1 | 30.4 |
| 6 | 25.8 | 38.3 |
| 7 | 3.6 | 11.3 |
| 8 | 5.6 | 15.3 |
| 9 | 28.3 | 35.2 |
| 10 | 36.5 | 36.5 |
| 11 | 22.6 | 32.4 |
| 12 | 37.2 | 39.3 |
| 13 | 29.4 | 35.3 |
| 14 | 39.4 | 36.8 |
| 15 | 37.5 | 39.2 |
| 16 | 45.5 | 42.4 |
| 17 | 30.2 | 36.2 |
| 18 | 40.8 | 42.7 |
| All (1–18) | 516.2 | 420.6 |

Data set 2 – Baggage handling system [transport orders / hour]

| Source | Ø | σ |
|---|---|---|
| 1 | 71.8 | 121.1 |
| 2 | 57.4 | 100.4 |
| 3 | 118.7 | 133.3 |
| 4 | 43.5 | 48.5 |
| 5 | 3.6 | 4.4 |
| 6 | 2.7 | 4.9 |
| 7 | 1.3 | 2.2 |
| 8 | 16.7 | 32.9 |
| 9 | 124.9 | 134.6 |
| 10 | 19.8 | 18.4 |
| 11 | 14.4 | 15.9 |
| 12 | 36.5 | 35.5 |
| 13 | 0.3 | 0.9 |
| 14 | 60.8 | 100.8 |
| 15 | 53.6 | 55.1 |
| 16 | 81.2 | 63.6 |
| 17 | 2.5 | 3.1 |
| 18 | 26.7 | 54.4 |
| All (1–18) | 736.5 | 715.3 |

Table 6: Average throughput per source & hour

Many sources of data set 1 show a similar shape of load profile. The peak during the beginning of the day is followed by a decreasing demand during the rest of the day. Smaller peaks might appear, but they are not as high as in the beginning of the day. In contrast there are also a few sources that show a different behavior. They can be characterized either by load peaks during the second half of each day or a number of rather similar peaks during the whole day (see figure below). These demand imbalances and the shifting peaks make dispatching difficult.

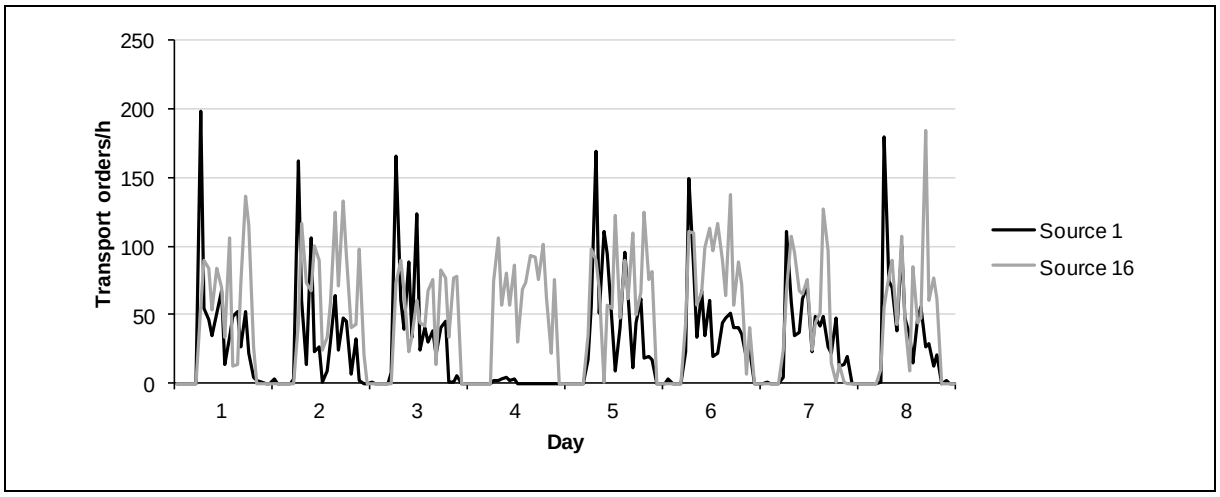



Figure 34: Load profile for data set 1 (source 1 and source 16)

The major challenges for data set 2 are the huge differences between the demands at the different sources. In addition, similar to data set 1, the control system has to deal with source-specific load profiles. The figure below compares source 2 and source 15. They both have a similar average throughput. But as can be guessed from the standard deviation of the hourly throughput, the load profiles are completely different. The demand at source 2 shows many peaks while the demand at source 15 is comparably stable. At certain times source 15 has a vehicle demand while no loads need to be picked up at source 2. These characteristics are challenges for the dispatching process.

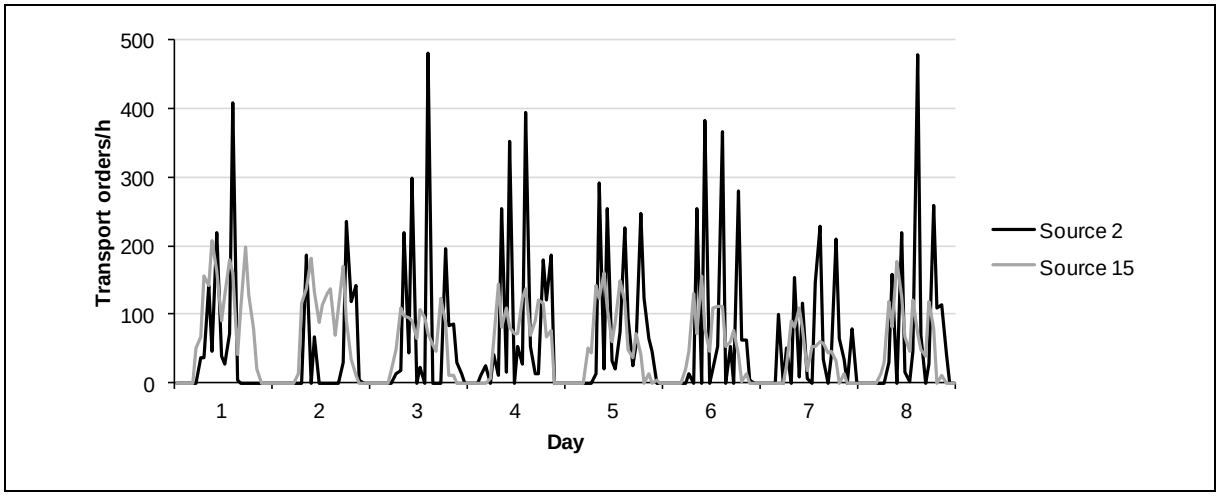



Figure 35: Load profile for data set 2 (source 2 and source 15)

Several test runs have shown that using the original input data sets does not yield useful results for all of the simulation configurations which will be analyzed. The load data and the theoretically developed test system layouts do not fit to each other. The system is not able to achieve the throughput that the load data induces. Therefore an elongation factor is introduced. All inter-arrival times of the input data sets are multiplied with the elongation factor. Thereby the load profile can be adjusted to the characteristics of the analyzed layouts. The starting point for this elongation is always the original data set, i.e. an elongation factor of 1.0. In case the system does not achieve the required throughput, the elongation factor is increased stepwise by 0.1. The elongation factors which are used for the experiments are shown in Figure 37 below. They have been derived by finding at least one operational system configuration per dispatching strategy.

### 6.1.3 Combination and parameterization of the control strategies

For testing the performance of different dispatching strategies, they have to be combined with other appropriate control strategies. The required combinations were already mentioned above, but should be summarized in this section. Random dispatching requires a random routing algorithm as no destinations are assigned to the empty vehicles. They have to continue travelling randomly. All other routing strategies are combined with static central routing (see also Figure 37). For D3 and D4 random routing is required for the vehicle release process. All merges are controlled by a FCFS logic. The control of switches needs to be looked at from two perspectives. Load-dependent sink and source approach are used in combination with all dispatching strategies. But load-dependent re-routing is only applied for preventing deadlocks in combination with D1, D3 and D4 as it was initially developed for these strategies.

During a number of experimental pre-studies several parameter settings for the dispatching strategies have been tested and evaluated. The goal was not to optimize the control strategy performance. Otherwise we would have followed a different approach and use, for example, "design of experiments" for a detailed assessment. But with the numerous system configurations even a design of experiments approach would have led to an extreme experimentation effort. We decided that the effort could not be justified for the pioneering comparison of this thesis and knowingly skipped the chance to identify all interdependencies between the different parameters. In the first place, we tried to find parameter settings that gave us operational system configurations (see Chapter 6.3.1). In addition, we strived for identifying the most important parameters which drive the behavior of each decentral algorithm. The following tables show the parameter settings which are used for the simulation experiments. Based on the pre-studies the parameters are judged for their impact on the system performance.

For forecast dispatching (D3) the following parameter settings are used during the experiments:

| Parameter | Tested settings | Chosen settings (per layout scenario) | | | Comment |
|---|---|---|---|---|---|
| | | 1 | 2 | 3 | |
| $t^N$ | 180; 360; 1,440; 2,880 | 360 | 360 | 360 | Very powerful parameter. It should be set to a value of several minutes, but 1,000 seconds should not be exceeded. |
| $\varepsilon$ | 0.15; 0.3; 0.5; 0.6; 0.8 | 0.5 | 0.5 | 0.5 | Less sensitive parameter, values between 0.2 and 0.6 provide goods results. |
| $\rho$ | 0.1; 0.25; 0.4; 0.6 | 0.25 | 0.60 | 0.25 | Less sensitive parameter, values between 0.2 and 0.6 provide goods results. |
| $t_{sou}^{rel}$ | 30; 60; 240 | 30 | 60 | 30 | Shorter release times are usually better: Vehicles which can currently not claim a load at a source should proceed quickly to find work elsewhere, values between 30 and 60 seconds deliver the best results. |
| $t_{sto}^{rel}$ | 100; 400; 600; 1,200 | 100 | 100 | 600 | Less sensitive than the release time at the sources, values between 100 and 600 seconds deliver good results. |
| $u_{thr}$ | 0.2; 0.4; 0.6; 0.8 | 0.6 | 0.6 | 0.7 | Usually values between 0.6 and 0.8 lead to fairly good results. |

Table 7: Parameter settings for forecast dispatching (D3)

For forecast dispatching (D4) the following parameters are used during the experiments:

| Parameter | Tested settings | Chosen settings (per layout scenario) | | | Comment |
|---|---|---|---|---|---|
| | | 1 | 2 | 3 | |
| $\tau$ | 0.005; 0.05; 0.2 | 0.005 | 0.005 | 0.05 | In more complex systems a more reactive calculation is required. Therefore the parameter value needs to be increased. |
| $b$ | 5; 10; 20; 30; 100; 1,000 | 1,000 | 1,000 | 20 | Especially in more complex systems it makes sense to limit the number of changes in order to increase the stability of the results. |
| $c_{max}$ | 0.1; 0.2; 0.3; 0.4 | 0.1 | 0.1 | 0.2 | This parameter should be kept at a rather low value. Otherwise the calculation of the dispatching probabilities starts to oscillate between 0 and 1 (especially for values > 0.7). |
| $t_{thr}$ | 400; 500; 600; 700; 800 | 500 | 500 | 500 | Powerful parameter that should be kept slightly below the technical threshold of about 600 seconds (20 buffer positions with 30 seconds load setdown and pickup time each). |
| $t_{sou}^{rel}$ | 30; 60; 120; 240 | 120 | 120 | 60 | Powerful parameter which is required to balance the system. Often it is useful to accept a slightly worse system performance but in return achieving a wider range of operational system configurations (especially for more complex systems). |
| $t_{sto}^{rel}$ | 100; 400; 1,200 | 1,200 | 1,200 | 720 | In smaller systems the variation of this factor has a big impact. The importance decreases when the system gets more complex. |
| $u_{thr}$ | 0.5; 0.6; 0.7; 0.8 | 0.6 | 0.6 | 0.7 | Usually values between 0.6 and 0.8 lead to fairly good results. |

Table 8: Parameter settings for feedback-based dispatching (D4)

For all strategies the update interval for the vehicle release process has been set to the minimum of $t_{sou}^{rel}$ and $t_{sto}^{rel}$. As the chapter on the results will show (see Chapter 7.1), random dispatching does not achieve the required throughput performance for any input parameter combination. This is also true for very short release times. We therefore did not spend too much time on pre-studies and kept the effort low by using release times of 60 seconds at the source and 300 seconds at the storage location for all layout scenarios. In a similar fashion the threshold value $u_{thr}$ for load-dependent re-routing is set to the same values as for D3 and D4.

### 6.1.4 Number of vehicles and default settings

The number of vehicles is one of the three major design decisions in a transport system and highly influences the system performance. It can only be estimated when a certain desired performance level is to be achieved (LE-ANH & DE KOSTER 2006) and interdependencies with layout and control strategies have to be taken into account (see Chapter 2.4). We do not have a predefined performance that needs to be achieved and the true characteristics of the developed control strategies are still unknown. We therefore follow a different approach and treat the number of vehicles in the system as one of our variable input parameters. We want to examine how it influences the system performance and how it interacts with the used control strategies. The number of vehicles is to be varied stepwise between a minimum and a maximum value. The minimum value is set to the number of storage locations in the system, i.e. two in layout scenario 1. As the vehicles initially start at the storage

locations, this ensures that there is at least one vehicle available at each storage location and source. The maximum number of vehicles in a transport system can be calculated by dividing the overall length of all paths by the length of one vehicle. But the result is only a theoretical figure. For several reasons, the maximum number of vehicles in the system is limited to 25% of the theoretical maximum in this study:

- The theoretical maximum would never be applied in a real-world scenario. Independently travelling vehicles or carriers would be replaced by a continuous conveyor if the throughput requirements lead to more than 25% vehicle coverage.
- Pre-studies have shown that even vehicle numbers of less than 25% might lead to significant mutual interferences (congestion, deadlocks etc.) of the vehicles. Using more than 25% would therefore only result in deadlocks and inefficient systems. The elongation factors can be used to adjust the throughout requirements to vehicle figures which are operational with less than 25% vehicle coverage and allow detailed analysis.
- The number of vehicles in the system has a severe impact on the duration of each simulation run. By not testing every theoretical case, the effort for the simulation experiments can be reduced considerably.

In the test system a vehicle is one meter long. The single-loop test layout has a path length of almost 1,000 meters (without storage locations). The path length per loop is slightly increased in layout scenario 2 and 3 as additional transfers between the loops are required. The maximum number of vehicles is set to 278 for each basic test loop. Initially, each layout scenario is to be tested with 10 different vehicle configurations according to the table below:

| Vehicle configuration | Share of vehicle maximum | # of vehicles | | |
|---|---|---|---|---|
| | | Layout scenario 1 | Layout scenario 2 | Layout scenario 3 |
| 1 | Minimum | 2 | 8 | 18 |
| 2 | ~ 5% | 14 | 56 | 126 |
| 3 | ~ 10% | 28 | 112 | 252 |
| 4 | ~15% | 42 | 168 | 378 |
| 5 | ~25% | 70 | 280 | 630 |
| 6 | ~35% | 98 | 392 | 882 |
| 7 | ~50% | 140 | 560 | 1,260 |
| 8 | ~65% | 180 | 720 | 1,620 |
| 9 | ~80% | 222 | 888 | 1,998 |
| 10 | ~100% | 278 | 1,112 | 2,502 |

Table 9: Vehicle configurations for simulation experiments

One of the core assumptions of this thesis is the distribution of empty vehicles to storage locations which have sufficient capacity. The impact of jams etc. which are caused by vehicles which do not fit into the existing storage locations should be minimized in order to make the true characteristics of the control strategies observable. Due to the modelling effort the same basic layout should be used for each layout scenario and each vehicle configuration. For D2 and D5 the storage location capacity $s_i^{max}$ is defined based on the number of vehicles and the number of storage locations in the system. But as must be expected that the defined capacity is exceeded for D5 and the requirements of D3 and D4 are unknown, storage locations with an increased capacity are integrated into the basic test layout. Each of the storage locations is able to physically accommodate 350 vehicles. Based on the maximum

numbers of vehicles in each layout scenario up to 125% (scenario 1), 31% (scenario 2) and 14% (scenario 3) of the system vehicles can be parked at each of the storage locations.

But the size of the buffers should not influence the measured performance indicators. The simulation model configuration does therefore contain that the vehicles travel 1,000 times faster in the storage locations than in the rest of the system. Thereby the influence of the storage location capacity on the average service time is minimized. Additionally, the distance which the vehicles travel in the storage locations is subtracted from the overall travelled distance. The figure below shows the adjusted version of the layout with increased storage location capacity for layout scenario 2. Due to this approach only one layout is required per scenario and can be used to test all vehicle configurations.

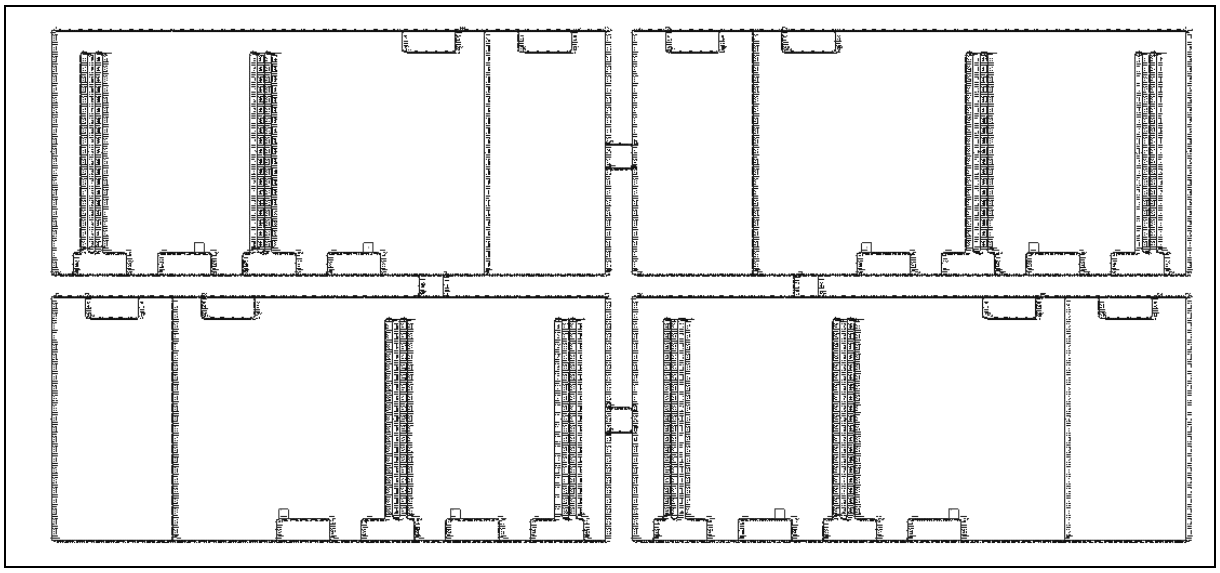


Figure 36: AutoMod layout of scenario 2 with increased storage location capacity

At the beginning of each simulation run the vehicles are distributed equally to all storage locations in the system. They enter the system at the entrance of the storage location and afterwards wait at the exit until they are called by a load.

The default AutoMod settings are used for the vehicles in the system:

| Parameter | Unit | Settings |
|---|---|---|
| Acceleration | $m/sec^2$ | 0.3 |
| Deceleration | $m/sec^2$ | 1.0 |
| Forward velocity | m/sec | 1.0 |
| Curve velocity | m/sec | 1.0 |
| Load pick up time | sec | 15.0 |
| Load set down time | sec | 15.0 |

Table 10: Default vehicle settings

## 6.2 Execution of experiments

Once the input parameters have been defined, the simulation experiments can be carried out. Several decisions regarding their parameterization have to be made.

### 6.2.1 Considered system configurations

The introduced input parameters are used in different combinations to generate the results of this study. Each dispatching strategy is combined with each layout scenario and each input data set. For each of the dispatching strategies a suitable routing algorithm is chosen (see Chapter 5.3). In total, 30 different experimental setups are generated (see figure below). For all experiments the switch and merge control strategies which were introduced above are in use. Each experimental setup is simulated with the defined 10 vehicle configurations, i.e. 300 simulation experiments are carried out in total.

| | | Layout scenario | 1 | | 2 | | 3 | |
|---|---|---|---|---|---|---|---|---|
| | | Input data set | 1 | 2 | 1 | 2 | 1 | 2 |
| | | Load enlongation factor | 1.0 | 1.2 | 1.0 | 1.7 | 1.8 | 3.0 |
| D1 | R2/R1* | I1+I2+I3 | 1.1.1 | 1.2.1 | 2.1.1 | 2.2.1 | 3.1.1 | 3.2.1 |
| D2 | R2/R2 | I1+I2 | 1.1.2 | 1.2.2 | 2.1.2 | 2.2.2 | 3.1.2 | 3.2.2 |
| D3 | R2/R1*+R2 | I1+I2+I3 | 1.1.3 | 1.2.3 | 2.1.3 | 2.2.3 | 3.1.3 | 3.2.3 |
| D4 | R2/R1*+R2 | I1+I2+I3 | 1.1.4 | 1.2.4 | 2.1.4 | 2.2.4 | 3.1.4 | 3.2.4 |
| D5 | R2/R2 | I1+I2 | 1.1.5 | 1.2.5 | 2.1.5 | 2.2.5 | 3.1.5 | 3.2.5 |
| Dispatching | Routing (laden/unladen) | Intersection control | Simulation setups | | | | | |

Naming of simulation setups: X.Y.Z: X – Layout scenario, Y – Input data set, Z – Dispatching strategy
*Application of R1 for release processes

Figure 37: Simulation setups

### 6.2.2 Warmup phase

In order to obtain meaningful results, the execution of simulation experiments needs to follow statistical requirements. The startup problem is the first topic to discuss in this context. As mentioned, each simulation experiment estimates the true model characteristics. All output data mean estimations have to be combined with confidence intervals that contain the true mean with a certain probability (see Chapter 6.3.2). The simulation model must have reached a steady-state before the collection of output data starts. Otherwise the statistical estimations can be biased. Especially the mean estimations are affected in this context. Therefore, it is necessary to define a warmup period for each simulation run. The collection of output data starts after the warmup period. At this time the simulation model has reached its steady-state and the problem of initial transient can be avoided (LAW 2006).

The simplest and most popular method to define the length of the warmup period is the graphical analysis of Welsh (LAW 2006). But to apply this method for each of the defined 300 simulation experiments would be extremely complex. A simplification is necessary and also possible due to the nature of our simulation experiments and the used input data sets.

Both input data sets follow a repetitive pattern. Although the exact hourly transport demands change from day to day, the data has a cyclic structure. Low or no demand during the night time is followed

by several demand peaks during the days. Afterwards follows a less busy night and so on. The simulation data of both data sets starts at 00.00 (midnight) of the first day.

According to the initialization procedure the simulation starts with all vehicles being equally distributed to the storage locations in the system. This initial state equals the status which the system reaches during periods without demand when the dispatching strategies D2 and D5 are applied. The strategies distribute the vehicles equally to all storage locations when the system is idle. As our observation period is longer than one day, we would therefore not need to include a warmup period for those strategies. The system is at the beginning of the simulation run in a state that it could potentially reach during the run. For D2 and D5 it definitely reaches this state during idle periods.

An equal argumentation could be applied for D3 and D4. The system could also reach the same state as at the beginning of the simulation at any point in time during the simulation run. It is less likely than for D2 and D5 that this state is exactly reached. But, it cannot be excluded.

However, we have decided to use a warmup period of one day. This means we record data when 24 hours of input data have been used in the simulation. The simulation length which equals one day of input data depends on the input modification factor and needs to be calculated for each combination of layout scenario, input data set and load modification factor. If the load modifications factor is 1.0, the warmup period equals 24 hours of simulation time. If the load modification factor is 1.7, the warmup period equals 41 hours of simulation time (1.7 x 24 hours).

The reason for using the warmup period was the fact that D3 and D4 both use a certain logic to spread network knowledge in the system. The vehicles inform sinks (D3) and switches (D4) about the dispatching locations in the system. As pre-studies have shown, the knowledge spread process might take several hours of simulation time. Although there was no evidence that this process significantly affects the average performance estimators of the system, we wanted to exclude any side-effects and therefore chose to use the one day warmup phase.

The figure below shows an example. The service times for each transported load were recorded during two simulation runs. One of the models uses D3, the other one D4. Both applied input data set 2 (experimental setups 2.2.3 and 2.2.4) and contained 168 vehicles. Although using only one run is not statistically sufficient, we show the data to illustrate the basic reasoning. The load data of the first day ends in both figures after the first peak which can be observed roughly at load 8,700. The required throughput on the first day is higher than on all the following of the five recorded days. Therefore it is hard to judge if there is really a difference in the daily patterns. At most a slightly higher variability of the service times might be present on day 1. Excluding the first day from the data collection process can therefore only improve the result quality.

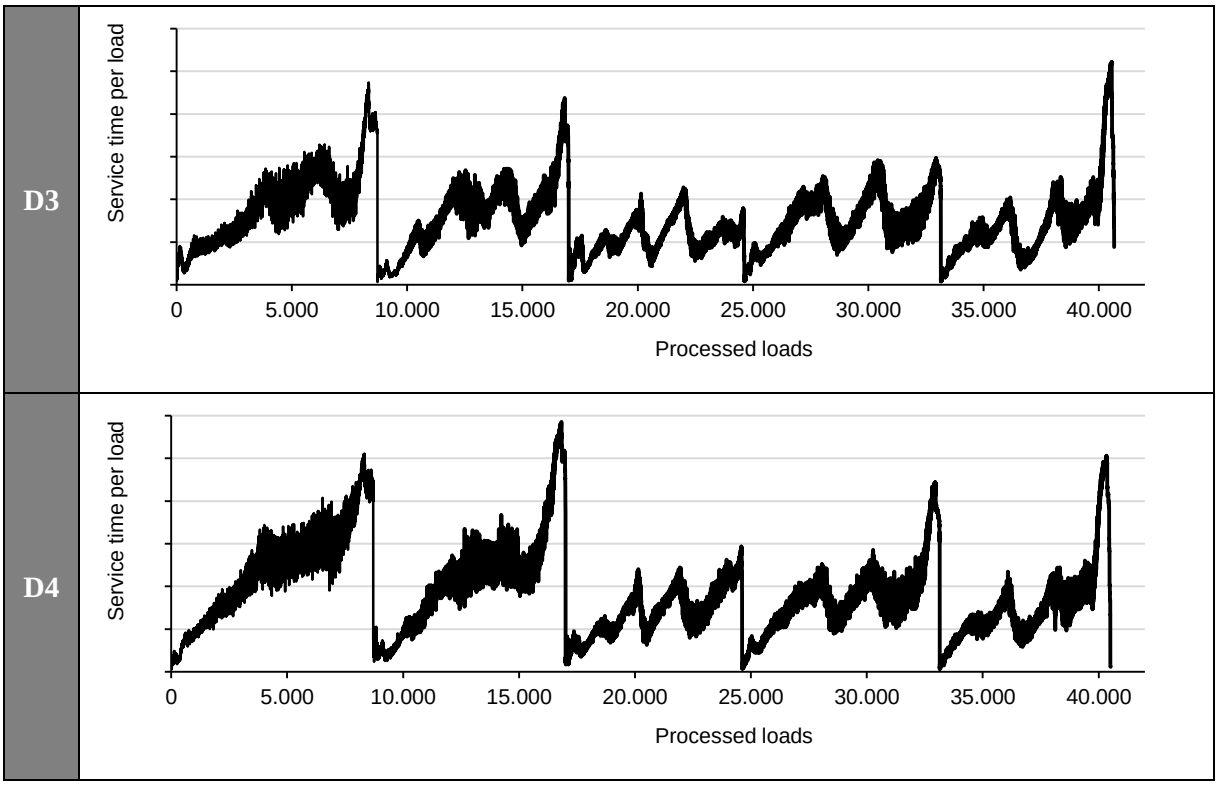



Figure 38: Service time per load (layout scenario 2, data set 2, 168 vehicles)

In a similar fashion the vehicle-based KPI can be examined. The figure below shows the hourly averages of the share that vehicles spend parking at storage locations. The shape of the curves is influenced by the initialization procedure at the very beginning. While the release process starts at the beginning of the simulation when D3 is applied, it is postponed to the first load arrival in scenarios with D4. Therefore the figure for D4 shows an initial value of 1 which is never reached again during rest of the simulation.

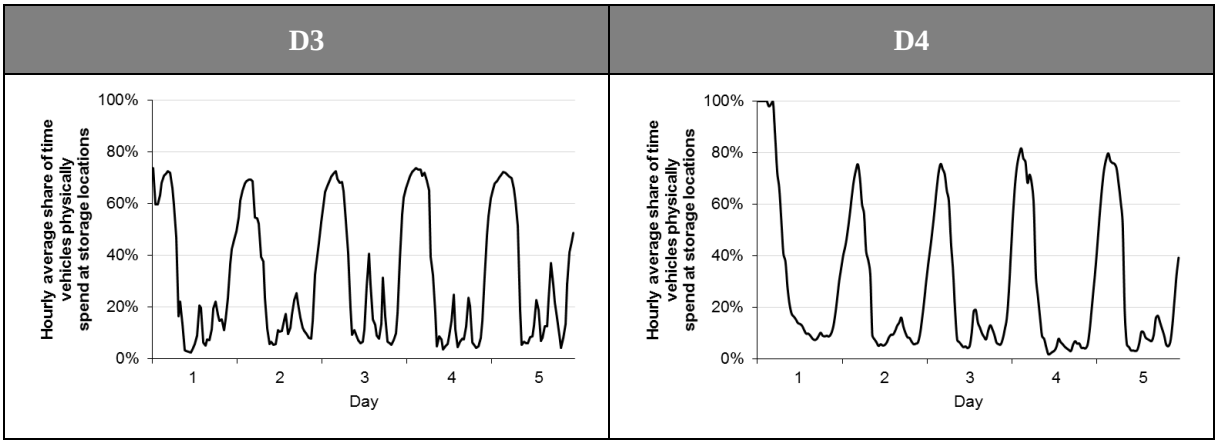



Figure 39: Share of time vehicles spend at storage locations
(Layout scenario 2, data set 2, 168 vehicles)

The two examples show that the impact of the initial transient is very small in our simulation setups. For the service times it might be questioned if there is any influence at all. However, deleting the output data of the first simulated day makes sure that all initialization effects are excluded from the KPI estimations. Looking at the overall simulation length which will be defined in the next

subchapter, one day of simulation data will not account for more than 4% of the overall simulation time in any of the simulation setups. Therefore even initialization effects that affect the second day and are extremely unlikely will only have a negligible impact on the overall KPI estimations. To avoid the graphical determination of the warmup phase is therefore a reasonable approach. In addition to the deletion of the warmup phase, a test for stationarity will be introduced below. It supplements these thoughts about the startup problem and makes sure that only operational system configurations are considered. As already mentioned, D1 does not achieve the throughput requirements of this thesis. Its warmup requirements were therefore not explicitly analyzed here.

### 6.2.3 Length of simulation runs

Besides defining the length of the warmup period, the definition of the length of the simulation run is the next important aspect. A first relevant decision to be made in this context is whether a replication/deletion approach or one single long simulation run is used (LAW 2006). We chose the latter approach as the simulation only has to go through the warmup phase once. In addition, the effort for analyzing the output data can be expected to be lower for a single run compared to merging the data of several runs. Both reasons refer to the simulation and analysis effort that the 300 simulation setups of this thesis require. While the most serious bias problem for the replication/deletion approach concerns the estimation of the mean, the usage of one long simulation run is more likely to affect the variance of the mean estimator and thereby the width of its confidence interval (LAW 2006).

The required length of the simulation run can now be defined by a sequential procedure (LAW 2006). The starting point is to define a desired precision of the KPI estimators. The precision depends on the variance and the number of samples which are included in the KPI estimation (see Chapter 6.3.2 below). While the former can be assumed to be constant in a steady-state simulation, it is the number of observations which defines the precision. As the number of observations grows with the length of the simulation, the latter has to be increased until the desired precision is reached.

Our objective was to find a simulation length that can be applied to every simulation setup in order to avoid individual parameterization of up to 300 models. Pre-studies showed that the performance of the decentral strategies can be unstable and unpredictable. Therefore the benchmark strategy D5 was used for defining the simulation length. The desired relative error of all KPI for the first operational system configuration ought to be less than 11% when central dispatching is applied. This rule was tested in all combinations of layout scenarios and input data sets. Finally, the average service time in layout scenario 3 with input data set 2 was the KPI that drove the simulation length. 1,800 hours of simulation time are required to obtain the required precision in all simulation setups.

### 6.2.4 Termination of simulation runs

Each of the 30 simulation setups is run with 10 different vehicle configurations. It needs to be expected that not all of the tested vehicle configurations will make the system operational. Depending on the control strategy, layout and input data set, some vehicle configurations might simply not achieve the required throughput. The results from these runs are not meaningful for our analysis. On the one hand, only system configurations that achieve the required throughput are to be compared in order to exclude this parameter from our detailed analysis. On the other hand, systems which do not achieve the required throughput will generally not achieve a steady-state. The statistical output analysis is therefore not applicable. We therefore try to limit the simulation effort by recognizing and terminating simulations which will not deliver useful results. A termination strategy ensures that a simulation run is only completed when the system is able to achieve the throughput requirement which is induced by the input dataset. Endlessly growing queues of waiting loads indicate that more vehicles are needed in the system. A simulation is terminated when the number of loads waiting at a source is

500% higher than the amount of loads which can be expected to be picked up during the remaining simulation time based on the experience from the transports which have already been completed. Besides reducing the simulation effort, this procedure also helps to prevent an AutoMod-specific problem. Loads which have entered the system but are not picked up and wait in a queue require a huge amount of memory. If the termination criterion was not in place, many simulation runs would otherwise by terminated due to low memory.

## 6.3 Output data analysis

Once the simulation experiments have been carried out, their output data needs to be analyzed. This subchapter explains the procedures which are used for this process.

### 6.3.1 Determination of operational system configurations

Using the termination criterion during the simulation runs already recognizes system configurations which do not achieve the required throughput performance. But for all completed runs, it has to be defined exactly which number of vehicles makes a system operational. In order to apply a selective criterion for this decision a statistical procedure is applied. Only systems with stationary output data are considered operational. Generally, we test for weak stationarity. This means that mean and variance of the tested time series are finite and constant over time. The autocovariances only depend on the lag length (see e.g. HACKL (2004) for a discussion on stationarity). We use the development of all queues at the sources in the system as decision criterion. The queue length is tracked on an hourly basis during the simulation.

Stationarity refers to properties that have implicitly already been included in the termination criterion. But it can hardly be checked during a simulation run. Having already eliminated obviously inoperable system configurations with the termination criterion, the following test can be considered as a detailed analysis. Besides identifying operational systems, they also make sure that the output data allows proper application of statistical analysis. Weak stationarity comprises the main requirements which are fulfilled by a simulation that has reached a steady-state.

The DF test and the KPSS test are used for the stationarity analysis. The reader is referred to the statistics literature for the theoretical background of the tests. We only state our test configuration here and do not describe the tests in detail.

The objective of the DF test is to test the null hypothesis of non-stationarity, i.e. a stochastic trend. The existence of a unit root is used as criterion to confirm the latter. In total there are three different versions of the basic test (see DICKEY & FULLER (1979) or HACKL (2004) for an overview). Each of them uses another assumption about the underlying time series model for calculating the test statistics. We use the first two models and state the autoregressive models for calculating the random variable $x_t$ at time $t$:

$$x_t = \varphi x_{t-1} + h_t \tag{6.1}$$

$$x_t = a + \varphi x_{t-1} + h_t \tag{6.2}$$

In these models $\varphi$ is the autoregressive parameter, $a$ is a constant and $h_t$ is the output of a white noise process. While the first model uses the null hypothesis of a random walk without drift (i.e. alternative hypothesis: stationary AR(1) process), the second uses the null hypothesis of a random walk with drift (i.e. alternative hypothesis: stationary AR(1) process with mean $\neq 0$). We do not consider the third

model. It would imply trend stationarity as alternative hypothesis. But this is exactly the characteristic that we want to exclude with the test.

In addition to the standard DF test, we also consider its extended version, i.e. the Augmented Dickey-Fuller test (see e.g. HACKL 2004). This version of the tests explicitly takes possible autocorrelation effects within the time series into account. We consider the same time series models as for the standard DF test. Choosing the correct number of lags is always a critical factor when applying the Augmented DF test. As we need a standardized approach for applying the test to many different systems, we use a very simple version which is stated by ZIVOT & WANG (2003). Firstly, the maximum number of lags $k_{max}$ is calculated based on the rule of thumb which was introduced by SCHWERT (1989) and where $M$ is the number of observations:

$$k_{max} = \left[ 12 \left( \frac{M}{100} \right)^{\frac{1}{4}} \right] \tag{6.3}$$

Afterwards the number of lags is reduced by 1 until the t-statistic for the significance test of the last lagged difference is greater than the absolute value of 1.6.

The KPSS test (KWIATKOWSKI ET AL 1992) interchanges the null hypothesis and the alternative hypothesis. It tests the null hypothesis of stationarity against the alternative hypothesis of non-stationarity. We only use the first version of the KPSS test and do not consider the null hypothesis of trend stationarity. Similar to the third version of the DF test, the second KPSS model would test for characteristics that we want to exclude.

All tests are carried out with the R statistics software (R DEVELOPMENT CORE TEAM 2012). The "ur.df" function from the "urca" library is applied for the (Augmented) Dickey-Fuller test and the "kpss.test" function from the "tseries" package for the KPSS test. For the KPSS test the default truncation lag parameter is used. The required statics for comparison are taken from RINNE (2008).

Using the different versions of the two tests, in total five test statistics have to be calculated and compared for each queue (model 1 and 2 of the DF test and the Augmented DF test, model 1 of the KPSS test). The Dickey-Fuller test has several disadvantages (see RINNE 2008). Knowing about those characteristics we compared the test results with graphical representations of the queues length. Based on the results, we chose to use a confidence level of 90% for the (Augmented) DF test and a 95% confidence level for the KPSS test.

It is commonly accepted that the Dickey-Fuller test and the KPSS test will in some cases lead to contradictive results. These are on the one hand caused by the limited discriminatory power of the DF test (RINNE 2008) and on the other hand by the fact that the hypotheses are interchanged. Due to the theory of hypothesis testing, this makes the results hard to compare as the confidence levels cannot easily be transferred. Therefore alternative approaches for the joint application have been developed (CHAREZMA & SYCZEWSKA 1998), but are not generally accepted as they deviate from the theory of hypothesis testing.

The possibly contradictive results of the tests make it hard to judge about stationarity. Therefore we developed a classification scheme for assessing stationarity. For each of the 30 simulation setups we only consider runs that have not been terminated according to the criterion introduced above. The remaining system configurations should all achieve the same throughput. Starting with the smallest number of vehicles, the system configurations are analyzed according to the following rules:

- The first operational setup (i.e. the operational system with the minimum number of vehicles) is defined as the system configuration for which all five test statistics favor stationarity with the smallest amount of vehicles for all queues. This means the (Augmented) DF test rejects the null hypothesis and the KPSS test accepts the null hypothesis. An additional requirement is a relative error of at most 15% for the mean service time estimator.
- For all other system configurations (i.e. systems configuration with numbers of vehicles which are bigger than the minimum number):
    - System configurations for which all five test statistics favor stationarity are considered to be operational.
    - System configurations for which only one of the two tests favors stationarity are considered to be operational, but shown in a different color in all figures.
    - System configurations for which both tests favor non-stationarity are considered to be none-operational.

The next section explains how the KPI are calculated based on the output data.

### 6.3.2 Estimation of the performance indicators

Due to the simulation approach all systems that are judged to be operational achieve the throughput that the input data sets require. The average service time is therefore the most important performance criterion. It is the distinctive factor for comparing the control strategies.

The service time contains the waiting time and the transport time. The waiting time is recorded from the point in time when the load enters the system until the point in time when it is actually picked up by a vehicle at one of the sources in the system. Once the order has been picked up, the transport time measurement starts. The transport time covers the time period from pickup until delivery at the source.

At first sight, the waiting time of the loads seems to be more important when assessing dispatching strategies. A direct link can be seen. But the simulation studies have shown that the transport time becomes equally important when there are many vehicles in the system. As soon as the vehicles start to interfere with each other (because there are so many vehicles) the decision processes become more complex. Simply minimizing waiting times without looking at the transport times will not work for systems with many vehicles. The flow of empty vehicles does also influence the flow of the full vehicles and therefore needs to be considered. It was already mentioned that additional load balancing strategies are required. Therefore the service time which also takes the transport time into account needs to be analyzed. Also from the perspective of each load the overall time it spends in the system is relevant, not only the waiting time.

In addition to the average service time which is based on the transport orders, performance indicators are required to measure the efficiency of the used control rules. We use two criteria to assess the efficiency:

- The number of vehicles in the system
- The average distance travelled per hour by all vehicles in the system

The number of vehicles in the systems can simply be read from the simulation experiment configuration. The distance travelled is calculated based on the AutoMod vehicle statistics. We consider the distance travelled without the portion which the vehicles spend between entrance and exit of the storage locations. As explained above, the storage locations have a comparably high capacity and shall not bias the results. We use a simplification for excluding the distance spent in the storage location from the overall travelled distance. As soon as a vehicle has passed the entrance of a storage location, the whole distance between storage location entrance and exit is subtracted from the overall distance travelled. This is necessary as the current implementation logic does not allow an exact determination of the vehicle position between the two control points. But this simplification is feasible and only has very limited influence on the overall results as the vehicles are travelling extremely fast while they are in the storage location. In addition, the share of the distance travelled in the storage location siding is rather small compared to the overall distance which the vehicles travel.

To get a better understanding of the system behavior a fourth KPI is used. It measures the time that the vehicles spend on average between storage location entrance and exit.

Summing up, the number of required vehicles (one of the two efficiency KPI) can be read from the simulation model configuration. Out of the other three KPI, the average service time is calculated based on the orders which were transported during the simulation run. The average time spent at storage locations and the average travelled distances are calculated based on hourly averages. The vehicle-based figures can only be measured with respect to a certain time scale. As hourly throughput figures are commonly used, it was decided to adopt this perspective.

The last three KPI have to be estimated based on the simulation results. Simulation studies are experiments with stochastic processes. The calculated average figures are therefore not necessarily the average figures which can be expected in reality. It is rather necessary to build a confidence interval for each of the averages. With a defined probability the estimated mean can be expected to be within this confidence interval. The confidence interval is calculated based on the variance of the mean estimator. Besides being covariance-stationary the output data has to show additional characteristics to enable the proper application of statistical calculations. In this context, uncorrelated data is the most important requirement. Only for independent and identically distributed random variables classical statistical techniques are applicable. Otherwise the variance of the mean estimator is not an unbiased estimation. The calculated confidence interval might be too small. However, simulation data is almost always correlated (LAW 2006). In the case of performing only one single simulation run to generate the output data, the most common approach for reducing the correlation is the batch means approach. We tried to use batch means but were not successful in eliminating autocorrelation. Especially in small systems the relatively strong and constant input patterns led to highly correlated output data. Therefore a procedure from spectral analysis is used for calculating the confidence intervals. Before the procedure is introduced another problem in the context of autocorrelation shall be highlighted.

Most publications on simulation mention that the output data must not be autocorrelated to allow a proper statistical analysis. But none of the authors defines exactly how autocorrelation should be measured and how the non-existence of autocorrelation should be assessed. The major question is the

number of lags that should be considered when judging the existence of autocorrelation. Only LIEBL (1995) states clearly that the non-existence of serial autocorrelation is sufficient. But neither KHEIR (1995) nor LAW (2006) answers this question explicitly. However, from their statements can be concluded that they do not consider the non-existence of serial autocorrelation a sufficient condition. As precise statements are missing, a look at the standard statistics literature should be helpful to understand how the non-existence of autocorrelation can be confirmed. But the dilemma continues. It is hard to find a precise statement about how to test for autocorrelation. Generally, the Durbin-Watson test can be used. But this test only considers serial correlation (SCHLITTGEN & STREITBERG 2001) and would therefore only be able to check the weak condition of LIEBL (1995).To test for autocorrelation of a higher order, the Box-Pierce statistic or the Ljung-Box test could be used. Both approaches are able to test for autocorrelation with respect to several lags. But for both tests a precise statement of the number of lags which should be included in the test is missing. Therefore the requirement of non-existence of autocorrelation is hard to prove. We used the Durbin-Watson test and several rules of thumb which can be found in the literature for testing lags of higher order. All results showed that the batch means procedure cannot reduce the autocorrelation sufficiently.

Therefore a method from spectral analysis is used for determining the confidence interval for the mean estimation $\bar{y}(M)$. The mean is simply estimated across all $M$ observations of the output data $y_1 \ldots y_M$. The autocorrelation of the underlying data is not relevant for the calculation of the confidence interval. The spectral analysis rather tries to explicitly include the correlation in its calculations as the covariances $C$ are summed up. The index $j$ specifies the separation of the covariances, i.e. the lag. $W_k(j)$ represents a weighting function. The factor $k$ needs to be specified.

$$Var\,[\bar{y}(M)] = \frac{C_0 + 2\sum_{j=1}^{k-1} W_k(j) C_j}{M} \tag{6.4}$$

Based on the variance estimation a 100(1-α) percent confidence interval can be calculated as:

$$\bar{y}(M) \pm\ t_{df,1-\alpha/2}\sqrt{Var[\bar{y}(M)]} \tag{6.5}$$

In this formula $df$ represents the degrees of freedom and depends on $M$, $k$ and $W_k$. Following LAW & KELTON (1984) we apply a Tukey window as weighting function. According to their references we assume an underlying t distribution with $AM/k$ degrees of freedom and set $k = M/10$. As $A$ is 1.33 for the Tukey window, this leads to 13 degrees of freedom for the t distribution.

The main disadvantage of the spectral analysis is its computational effort. Additionally, there are only rules of thumb for choosing $k$ (LAW 2006). For a discussion of all assumptions regarding the spectral analysis see LAW & KELTON (1984).

For the estimation of the KPI we use as confidence level of 95%. The half-length of the confidence interval of an estimator can be used to define the estimation precision or relative error $\theta$. It is calculated as follows (LAW 2006):

$$\theta = \frac{t_{df,1-\alpha/2}\sqrt{Var[\bar{y}(M)]}}{\bar{y}(M)} \tag{6.6}$$

The precision allows a statement about the quality of the estimation and can for example be used to define the required length of a simulation run (see Chapter 6.2.3). The simultaneous usage of multiple

KPI has implications for the interpretation of the results. If $j$ KPI are considered for an analysis with 100(1-$\alpha$) percent confidence intervals, the "probability $p_{sim}$ that all $j$ confidence intervals simultaneously contain their respective measures satisfies" (LAW 2006, page 537):

$$p_{sim} \geq 1 - \sum_{i=1}^{j} \alpha_i \tag{6.7}$$

This is another reason for using as few KPI as possible for the performance evaluation. With the three considered KPI, $p_{sim}$ can only be claimed to be greater than or equal to 85%.

### 6.3.3 Comparison of central and decentral strategies

The main objective of this thesis is to compare the performance of decentral control strategies with the performance of central strategies. The central dispatching strategy is considered the benchmark. This section explains the comparison procedure that is applied for evaluating the simulation setups. Six different analyses are created to compare central and decentral control strategies. Each of them evaluates central and decentral strategies with respect to one combination of layout scenario and input data set (e.g. Analysis 1 contains the experiments 1.1.1–1.1.5). Each of the six analyses uses the following methodology.

Having carried out the simulation experiments with the 10 different vehicle configurations, the stationarity analysis approach (see Chapter 6.3.1) is applied to define the first operational system configuration for the central dispatching strategy. Afterwards the results are refined by six additional runs. An example shall help to illustrate the approach (see figure below). A possible outcome for central dispatching in layout scenario 1 with input data set 1 could be that 28 vehicles are required for the first operational configuration. Based on the step size of the vehicle configurations this result shows that 14 vehicles (vehicle configuration 2) are not sufficient, but 28 (vehicle configuration 3) are. However, there is no statement about vehicle configurations in between. Therefore six additional simulation experiments are carried out with 16, 18, 20, 22, 24 and 26 vehicles respectively. The results are also tested for stationarity and the minimal number of required vehicles, i.e. the first operational system configuration, might be reduced to 22 vehicles. The results of the experiments simply get more precise when additional runs are carried out. This approach is necessary for getting a good understanding of the service time range which can be achieved with the benchmark strategy. We have found that minimal service times which are achievable with central dispatching usually converge to a certain boundary. Therefore additional experiments regarding the maximum number of vehicles do not provide further insights. The minimum number is far more interesting anyway, as it promises the lowest investment.

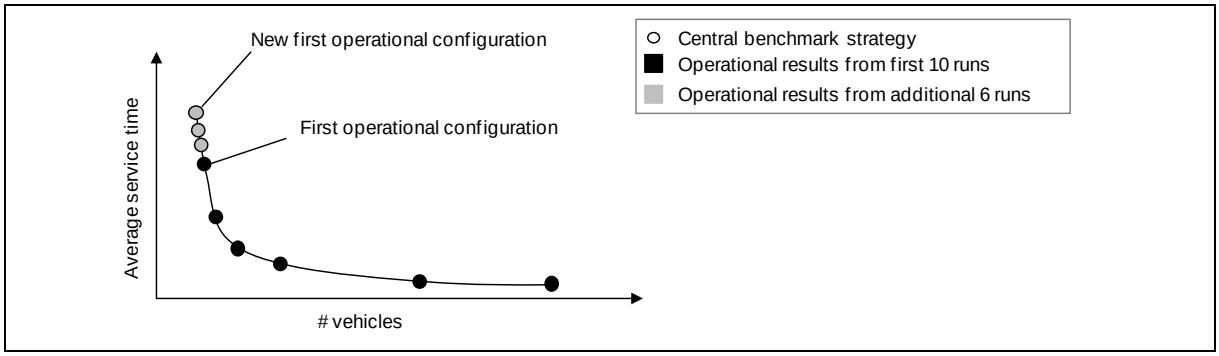



Figure 40: Exemplary results from initial and additional runs of benchmark strategy

The additional runs shape the performance curve of the benchmark strategy with the minimum and maximum average service times which are achievable (see figure above). In a second step, the results of the decentral strategies are added to the analysis and compared with those of the central strategy. Depending on the performance of the decentral strategies additional simulation runs are initiated to get a complete picture of their performance with respect to the benchmark performance corridor (see figure below):

- If the minimum vehicle requirement based on the first 10 runs shows an average service time which is smaller than the highest average service time of the benchmark strategy, 6 additional runs are carried out to check if an operational system configuration with fewer vehicles can be found.
- If the last operational system configuration (maximum number of vehicles) leads to an average service time which is higher than the lower boundary of the corridor which is covered by the benchmark strategy, 6 additional runs are carried out to check if lower service times are achievable. The runs are only carried out when the shape of the performance curve suggests that adding more vehicles will lead to better results.

The additional runs explore the average service time performance of the decentral strategies above the first operational and below the last operational system configuration. The results might be within or outside the central performance corridor.

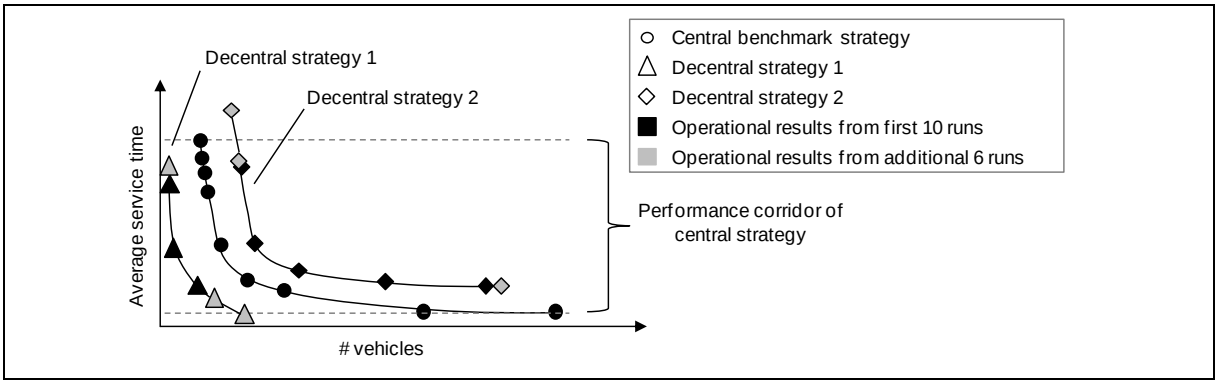



Figure 41: Additional runs for decentral strategies to explore benchmark corridor

The following evaluation of the decentral strategies judges their efficiency compared to the benchmark strategy. It calculates the additional vehicle requirement and the additional vehicle movement that the decentral strategies need to achieve the same performance level as the central benchmark strategy. The comparison curves are created based on the operational vehicle configurations of the central dispatching strategy. The figure below illustrates the calculation of the additional vehicle requirement. For each operational central configuration it is checked how many vehicles a decentral strategy would need to achieve the same average service time. A linear shape of the performance curves between two operational decentral vehicle configurations is assumed and interpolated accordingly. All comparisons are based on average service times. The comparison results in an average figure for the additional vehicle requirement. It could also be based on the lower and/or upper limits of the respective average service time confidence intervals to create best case and worst case scenarios. The results from the calculation of the additional vehicle requirement are the starting point for the evaluation of the additional vehicle movement. The simulation results are used to estimate for each decentral strategy how much vehicle movement would be required if the calculated additional vehicle requirements were used. Linear interpolation is applied if required.

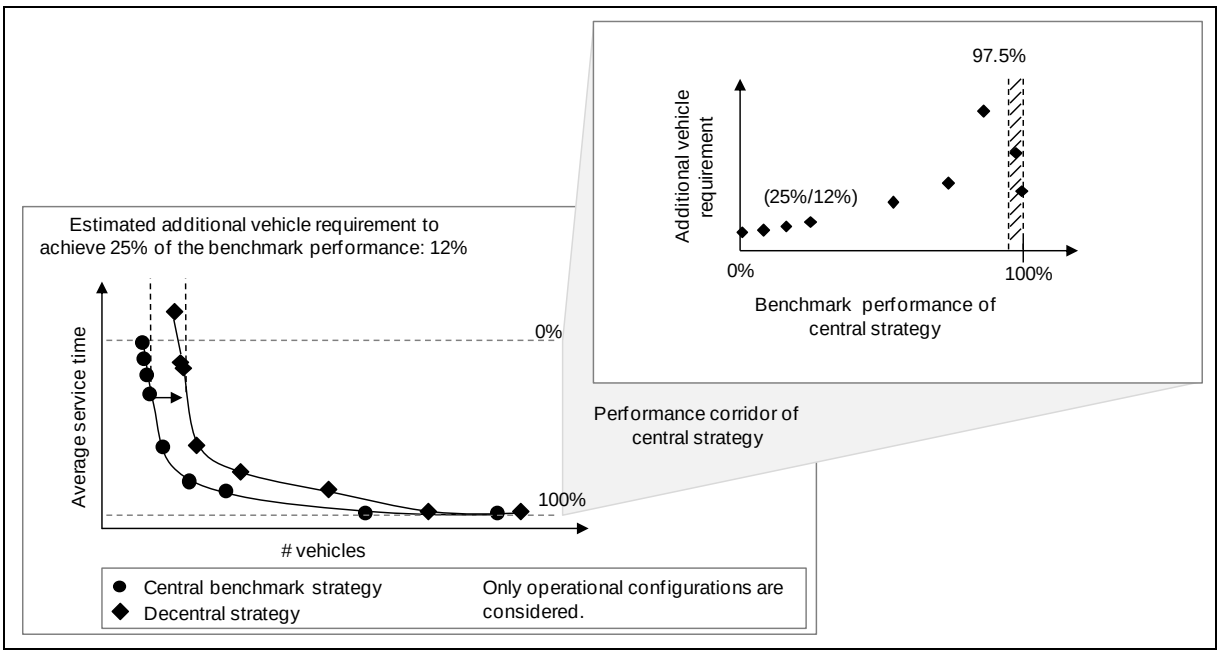



Figure 42: Methodology for calculation of additional vehicle requirement

We only consider the upper 97.5% of the performance range of the central strategy. The reason for this approach is that all the performance curves get quite flat with high vehicle figures. Even small differences between strategies can then lead to high differences regarding vehicle requirements and the computation of differences becomes fuzzy. A real-world system would not be operated in this area of the performance curve anyway. Generally, it should be noted that the analysis is only an estimation. All values which are considered are themselves only simulation-based performance estimates of the true system characteristics. The analysis is based on average service times. It does not explicitly take their confidence intervals and thereby the variance into account. The simulation methodology is supposed to limit the width of the confidence intervals in order to increase the reliability of the comparison results.

The description of the comparison procedure completes this chapter about the simulation methodology. The different input parameters and their combinations were described. All parameter settings for the simulation experiments were stated. Based on the introduced output data analysis techniques, the next chapter shows the results of the simulation experiments.

# 7 Results

The following sections present the output data analyses of the different simulation runs. In a first step the throughput is investigated. The evaluation helps to exclude one of the dispatching strategies from further consideration. Afterwards the performance of the remaining decentral dispatching strategies is compared to the central benchmark. Their efficiency with respect to the number of vehicles and required vehicle movement is derived. The chapter ends with several in-depth analyses. In this context disturbances and how they affect the performance of the transport system are one of the focal points. Except the throughput, all other performance indicators are normalized on the interval [0, 1]. 1 equals the maximum. As already mentioned above, the real-world input data sets and the generic layouts do not really fit each other. The elongation factor had to be introduced. However, the absolute performance values do not in most cases have a range that reflects real-world scenarios. In order to avoid confusion and as we are only interested in the relative comparison of central and decentral strategies, the normalization approach is chosen.

## 7.1 Throughput

The throughput is one of the core KPI for a transport system. This thesis tries to exclude it from the evaluation scope by only considering system configurations that achieve the same throughput. The termination criterion and the stationarity analysis are applied to derive the relevant operational system configurations. The figure below shows the achieved average throughput per hour for the different simulation setups and their vehicle configurations. They partly contain data from simulation runs which were terminated and not all the system configurations lead to operational systems. Therefore the statistical output analysis cannot be applied in every case. However, the figures provide first insights to the system characteristics. They are mainly used to make a simplifying decision for the following analyses. Due to the load elongation factor the shown throughputs might differ from those which are shown in Chapter 6.1.2.

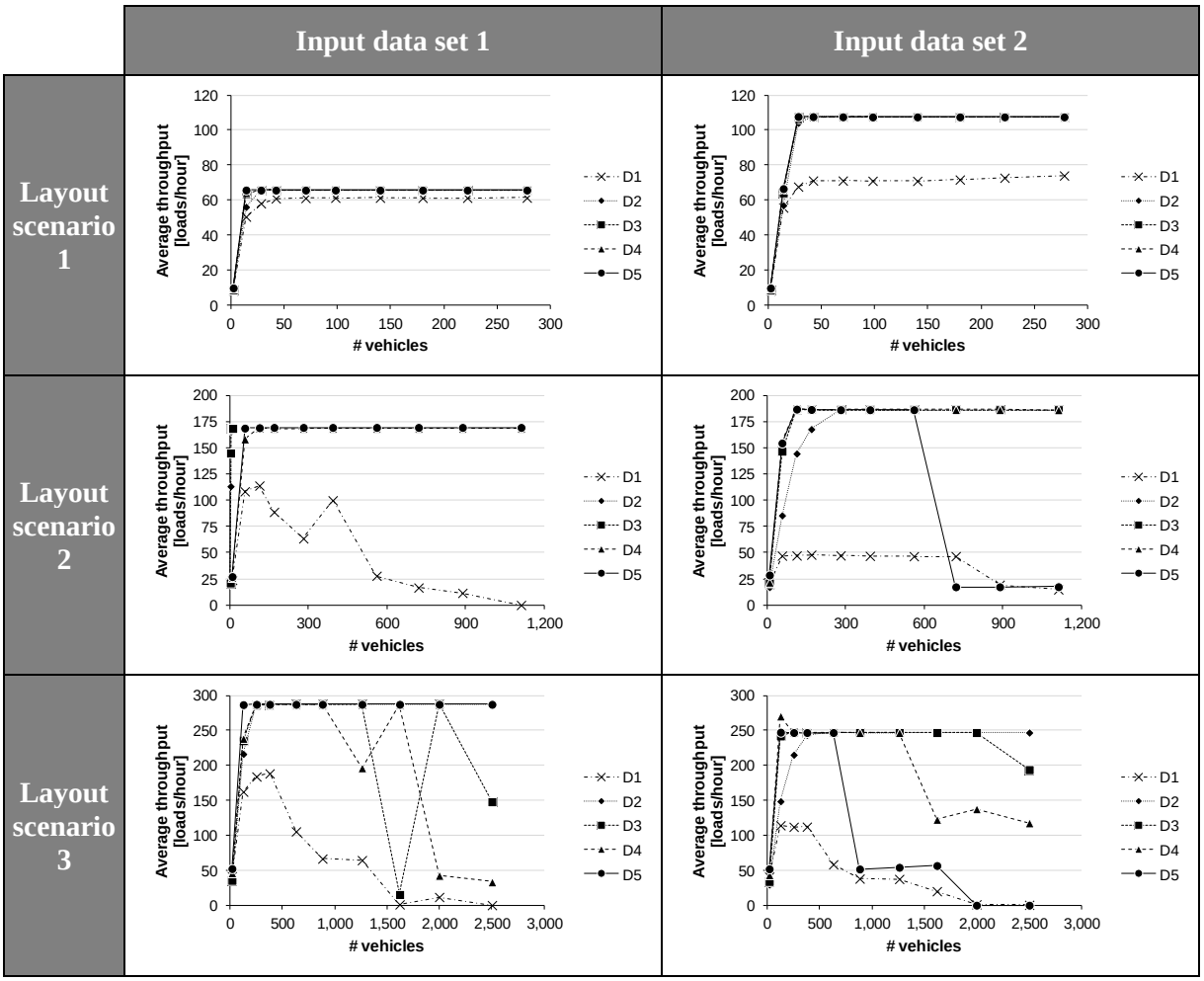



Figure 43: Average throughput per hour and simulation setup

Looking at the six diagrams, it becomes evident that there exists a certain upper boundary for the average throughput per hour in every simulation setup. The performance curves of the different control strategies approach this upper boundary when the number of vehicles in the system is increased. The upper throughput boundary is induced by the input data set and represents the throughput performance that a control strategy in combination with a vehicle configuration needs at least to achieve to be judged operational (additional requirements might apply). In contrast to layout scenario 1, the throughput performance in more complex layouts (scenario 2 and 3) is not stable across the whole range of vehicle configurations for some of the control strategies. Mutual interferences of the vehicles decrease the performance when too many are added to the system. Deadlocks can occur. These effects have already been mentioned during the description of the implemented control strategies and they are also relevant for the following analyses. The decrease of the throughput performance will usually lead to a termination of the simulation run.

Besides these effects, a look at the random dispatching strategy provides additional insights. Random dispatching is the only strategy that does not achieve the average throughput level which leads to an operational system configuration for any of the six scenarios. All simulations are terminated due to constantly increasing queue lengths. In the simplest layout (scenario 1) random dispatching still performs considerably well. But after a relatively steep initial increase, adding additional vehicles only results in rather small throughput growth. Due to the simple single-loop layout, deadlocks cannot occur (only 25% of vehicle coverage on the path layout). The vehicles circulate in the system. The network structure also leads to a very high probability of empty vehicles finding a sink or a storage

location. However, the single-loop layout also has a disadvantage. Leaving from one of the two sinks, it is much more likely that a vehicle travels to storage location 2 (E2) than to one of the sources or to the other storage location. As the direction is chosen randomly at each switch, the probability for directly arriving at E2 is 0.5 after the departure from a sink. The probability for arriving at source 2 is $0.5^2$, for storage location 1 it is $0.5^3$ and for source 1 it is $0.5^4$. The consequence is that loads at source 2 are processed far faster than at source 1. The queues at source 1 therefore trigger the termination of the simulation runs. The performance of random dispatching deteriorates for layout scenario 2 and 3. The main reason is that the layouts are far more complex. There are more switches and it less likely that a vehicle directly finds a storage location or a source when it has left a sink.

Summarizing, it can be said that random dispatching does not achieve the required throughput performance in the simple layout and even performs worse in more complex layouts. As the required throughput is not achieved, a comparison to the central dispatching strategies does not provide reasonable results. Therefore the random dispatching strategy will not be considered in the remainder of this thesis.

Although all other performance comparisons shall be left to the next sections, the absolute average throughput figures are considered as a last aspect. Due to the approach for load elongation, none of the simulation setups have the same throughput requirement. For layout scenario 1 the average throughput for the baggage handling data is about 64% above the level of the buffer replenishment data. There is only a small difference of 11% for layout scenario 2. For layout scenario 3 the throughput for the second input data set had to be decreased below the level of input data set 1. It is about 14% lower. This gives rise to the thought that the baggage handling input data is more demanding for the dispatching strategies.

After this first analysis which helped to eliminate the random dispatching strategy from our analysis, we will now specifically evaluate the performance of the dispatching strategies in the different layout scenarios. We only consider stationary, i.e. operational systems configurations. This means that they all achieve the same throughput rates. We can therefore limit our following thoughts to the comparison of the service times and related performance indicators.

## 7.2 Layout scenario 1

The comparison of central and decentral control strategies starts with the simple layout scenario 1. According to the procedure which was explained above (see Chapter 6.3.1), the first step is to find the first operational system configuration for each control strategy, i.e. the system configuration which achieves stationarity with the smallest number of vehicles. For input data set 1 the following figures have been derived:

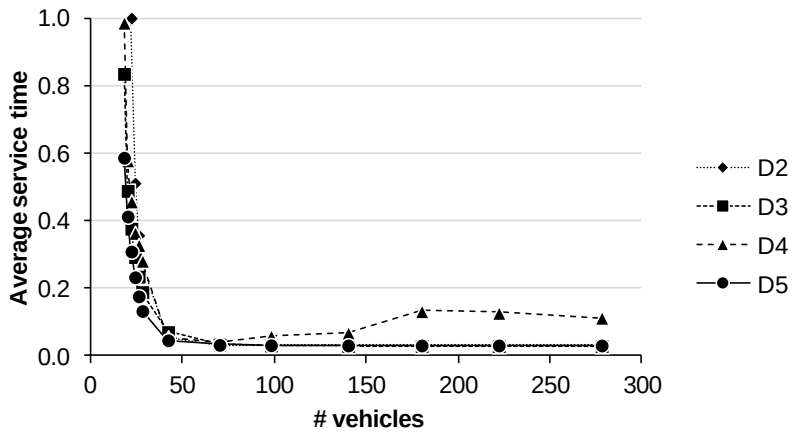
| Dispatching strategy | Simulation setup | Minimal vehicle requirement [# vehicles] |
|---|---|---|
| D2 | 1.1.2 | 22 |
| D3 | 1.1.3 | 18 |
| D4 | 1.1.4 | 18 |
| D5 | 1.1.5 | 18 |

Table 11: Minimal vehicle requirements for experiments 1.1.2–1.1.5

Strategies D3, D4 and D5 all require at least 18 vehicles, while D2 requires 22 for the first operational system. The corresponding average service times can be read out of the figure below which

summarizes the performance for all vehicle configurations and control strategies. D2 results in the highest average service time for the first operational system configuration. D3 needs on average about 16%, D4 about 1% and D5 about 41% less time for processing a load when the minimal vehicle configuration is used.

The performance curves for all strategies show a very quick decrease of the average service times below about 40 vehicles. For systems with more vehicles the average service times for D2, D3 and D5 only decrease slightly or remain constant. An increase of the service times can be observed for D4. The service time comprises load waiting time and transport time (see Chapter 6.3.2). Both performance indicators can therefore be used for an in-depth analysis. In contrast to all other strategies, the waiting times start to grow when D4 is applied in combination with more than 80 vehicles. As the transport times stay fairly stable, the increase of the average service times can only be attributed to the dispatching decisions and not to the increased delivery traffic.



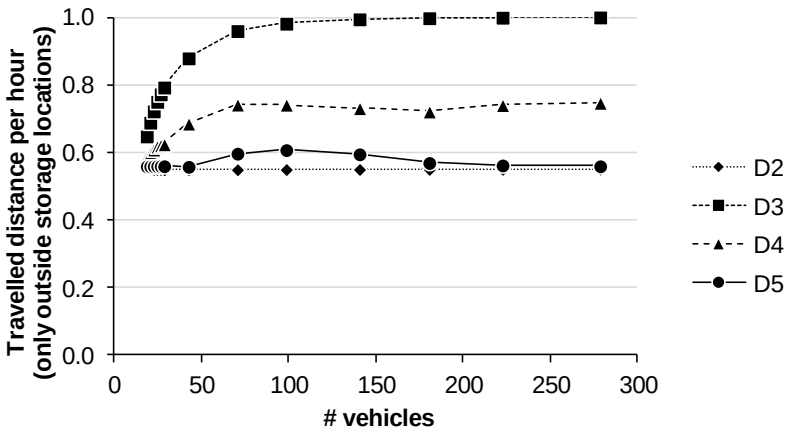

| Layout scenario 1 | Input data set 1 | Load elongation factor 1.0 |
|---|---|---|

Figure 44: Average service times for simulation experiments 1.1.2–1.1.5

Looking at the average service times in the figure above, D5 outperforms the other strategies. Its curve is the lower performance boundary across all strategies up to about 70 vehicles in the system. While D2 and D3 are able to reach the same average service time threshold value as D5 above 70 vehicles, no vehicle configuration achieves this benchmark value with D4. The performance of the latter control strategy always stays at least slightly above the benchmark figures.

Such comparisons have to be interpreted with caution as they are only based on the average values from the simulation experiments. Especially for D4, it can be observed that the size of the confidence intervals increases considerably with higher numbers of vehicles (see Chapter 7.6.1). Generally, pair-wise hypothesis testing would be necessary to confirm or reject significant differences statistically. As we are mainly interested in the efficiency of the different strategies and their comparison, we will not elaborate too much on evaluating the strategies based on the achieved average service times. The following comparisons will also show that the efficiency considerations concentrate on performance levels of D4 which result in quite small confidence intervals.

Besides the number of vehicles, the travelled distance is used as a second indicator for the efficiency of a control strategy. The figure below clearly indicates that D3 and D4 require far more vehicle movement than D2 and D5. This is true for all vehicle configurations. While D2 shows a completely stable movement requirement, the latter increases up to a certain value for D3 and D4. The small peak which can be observed for D5 will be explained in relation to the second input data set below.




| Layout scenario 1 | Input data set 1 | Load elongation factor 1.0 |
|---|---|---|

Figure 45: Average travelled distances per hour for simulation experiments 1.1.2–1.1.5

Having looked at the buffer replenishment data, we will now turn our attention to the second input data set. In this simulation setup the average throughput is about 64% higher.

| Dispatching strategy | Simulation setup | Minimal vehicle requirement [# vehicles] |
|---|---|---|
| D2 | 1.2.2 | 38 |
| D3 | 1.2.3 | 28 |
| D4 | 1.2.4 | 36 |
| D5 | 1.2.5 | 66 |

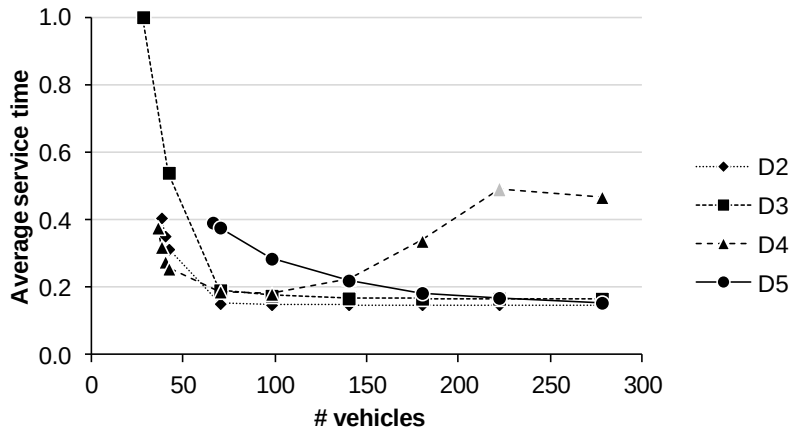
Table 12: Minimal vehicle requirements for experiments 1.2.2–1.2.5

The minimal number of required vehicles already indicates how well the dispatching strategies are able to deal with the changed demand patterns. The additional requirements of D2 (+72%) and D3 (55%) are in the range of the throughput increase. D4 needs twice as many vehicles as for input data set 1. D5 performs worse and requires an increase of 266% for the first operational system configuration. The corresponding first average service times of D2, D4 and D5 are all more than 60% below the initial performance of D3.

The figure below illustrates that all strategies individually show roughly the same behavior as in the case of the buffer replenishment data. However, the curves are initially not as steep as in the analysis above. For system configurations with many vehicles, D2, D3 and D5 approach a certain minimal performance threshold value. Similar to the behavior in the buffer replenishment case, the average service times for D4 start to increase from a certain vehicle amount and do not reach the lower performance boundary. In contrast to input data set 1, the growth is not only caused by the waiting time. An increase of the transport time due to mutual interferences of the vehicles can also be recognized.

Generally, it could be expected that the transport time increases for all strategies when more and more vehicles are added to the system. In the given test environment there are two reasons why this does not becomes observable in every case. On the one hand, the transport time represents only a very small share of the average service time. Therefore a slight increase of the transport time does not necessarily

become visible. On the other hand, storage locations with high capacity are used. Therefore even if many vehicles are used, there is enough room for storing them and preventing mutual interferences for most of the vehicle configurations.




| Layout scenario 1 | Input data set 2 | Load elongation factor 1.2 |
|---|---|---|
| Configurations that do not fulfill all stationarity requirements are shown in grey. | | |

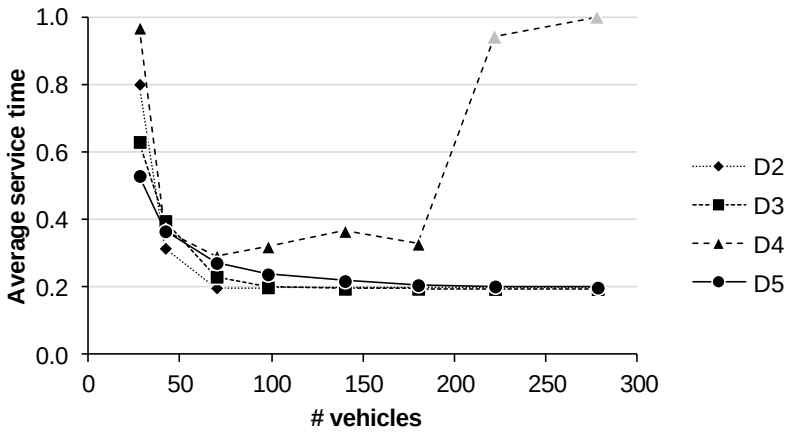
Figure 46: Average service times for simulation experiments 1.2.2–1.2.5

The major difference compared to input data set 1 is that the central strategy D5 does not generally achieve the best average service times and defines the lower boundary for the achievable performance. Instead, it requires more than 200 vehicles to reach the lower performance threshold value. The performance of the strategy depends on the layout of the system and the throughput pattern. The dispatching decisions are made based on the distances between empty vehicles, available loads and storage locations. In the single-loop layout of scenario 1, it is therefore more likely that a vehicle which requires dispatching at a sink is sent to source 2 instead of source 1 when loads are waiting at both locations. Similarly, storage location 2 will always be replenished before storage location 1 when both have the same fill level. Consequently, the service times for loads which were picked up at source 2 are rather short while the loads at source 1 have to wait longer. The system is not balanced and the overall average service time increases. The described effect is boosted by the input characteristics of data set 2. It shows a higher demand at source 1 than at source 2. But due to the distance-based decision making this source is served less frequently. The system performance decreases. For input data set 1 the demand characteristics are exactly the other way around.

In addition, the overall throughput has an impact on the performance. The vehicles are utilized less in the buffer replenishment scenario and are sent to the storage locations for a certain share of their time. As the empty vehicle positioning does not only consider the distance, but also the fill level for its decisions, it balances the vehicles in the system during idle periods. In the case of the baggage handling data the vehicles are sent to the storage locations less frequently because the required throughput is higher. The storage locations can therefore not balance the system. Vehicles often leave from a sink and directly approach a source. In this case priority is given to source 2 and leads to waiting loads at source 1.

To check the plausibility of the results and to verify the second of the mentioned reasons for the poor performance of the central algorithm, an additional series of experiments was carried out. The load modification factor for the baggage handling data was increased until the average throughput rate of loads per hour equals the level of the buffer replenishment data. The results show that the performance

of D5 is improved considerably compared to the other strategies when the throughput is reduced (see figure below). But also after the throughput adjustment the average service times of D5 do not represent the lower performance threshold values for most system configurations. The remaining differences are induced by the first reason which has been explained above, i.e. by the load profiles at the two sources in the system. Although the overall throughput has been modified, the demand at source 1 is still higher than at source 2.




| Layout scenario 1 | Input data set 2 | Load elongation factor 2.0 |
|---|---|---|
| Configurations that do not fulfill all stationarity requirements are shown in grey. | | |

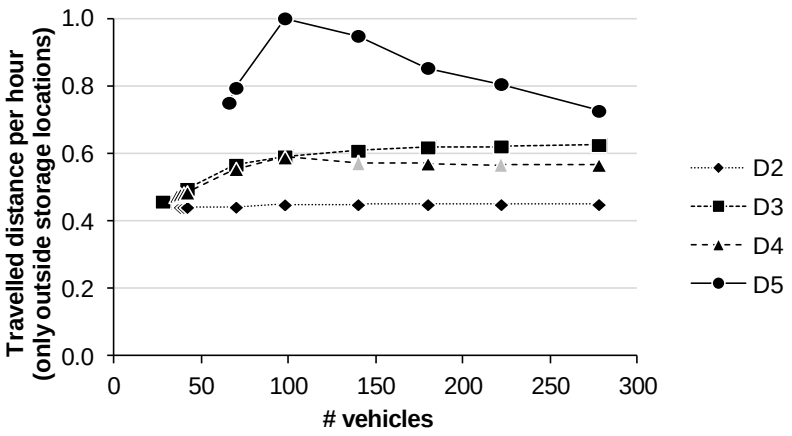
Figure 47: Average service times for simulation experiments 1.2.2 – 1.2.5 (elongation factor: 2.0)

Except for D5, Figure 48 for the travelled distances looks very similar to the findings for input data set 1. D2 shows again a stable level of vehicle movement across the different vehicle configurations. The slightly increasing shape of the curves for D3 and D4 is similar compared to the buffer replenishment data. However, the differences between D3 and D4 are far smaller and only become visible above 100 used vehicles. Both curves run closer to D2 than in the buffer replenishment case. The reason is that D3 and D4 do not perform very well in sending vehicles to storage locations. In many cases oversteering or understeering can be observed. In the case of the baggage handling data the overall throughput requirement is higher and therefore the time that vehicles spend at the storage locations in total is smaller. The importance of perfectly controlling the storage locations decreases. This leads to a reduction of the differences in vehicle movement.

But among all strategies, the biggest deviations compared to the first data scenario can be reported for D5. They highlight another interesting aspect which can be observed when D5 is used in simple systems with high throughput requirements. The travelled distance increases tremendously due to the early load-vehicle assignment in combination with the load-dependent source approach.

Vehicles are only allowed to travel to a source for pickup when enough buffer positions are available. In contrast to D3 and D4 the central dispatching algorithm directly assigns vehicles to loads. This process is not postponed until a vehicle physically arrives at a source for D5. Therefore, many vehicles are assigned to the waiting loads in times of high throughput rates. But the vehicles might be denied access to the sources. Therefore they start circulating in the system until enough buffer positions are available. This leads to an extreme increase in vehicle movement. The effect decreases when more vehicles are in the system because this automatically leads to an increased usage of the storage locations and more vehicles are called from the storage locations instead of being assigned after

having left from the sinks. Vehicles which are called from the storage locations wait until buffer positions are available before they start their journey and thereby cause less vehicle movement.



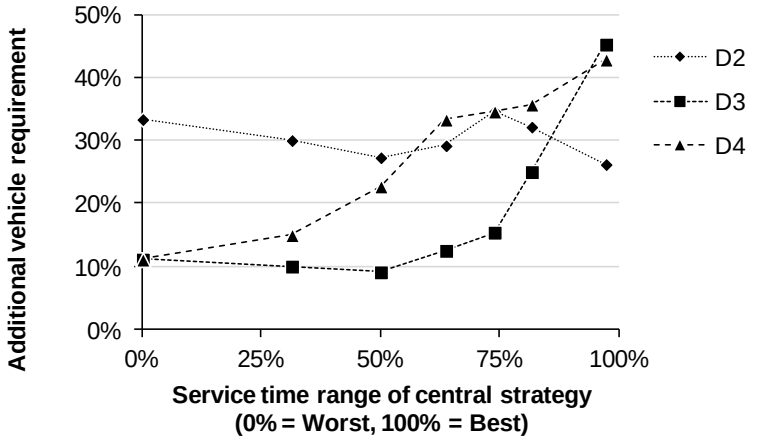

| Layout scenario 1 | Input data set 2 | Load elongation factor 1.2 |
|---|---|---|
| Configurations that do not fulfill all stationarity requirements are shown in grey. | | |

Figure 48: Average travelled distances per hour for simulation experiments 1.2.2–1.2.5

After having reviewed the results from both load scenarios, the first analysis section is completed by a comparison of the efficiency of central and decentral strategies. The additional vehicle requirement and additional vehicle movement are compared according to the methodology which was introduced above (see Chapter 6.3.3). Based on the computations the following ranges for additional vehicle requirement can be derived if input data set 1 is used with layout scenario 1:

- D2: +26% to +35% vehicles
- D3: +9% to +45% vehicles
- D4: +11% to +43% vehicles

The specific value always depends on the level of central performance that is to be achieved. The additional requirement for D2 is quite stable along the whole performance range of D5. For D4 it increases constantly. D3 requires comparably few additional vehicles for the lower 60% of the performance range. Above this level, a rather steep additional requirement can be observed.

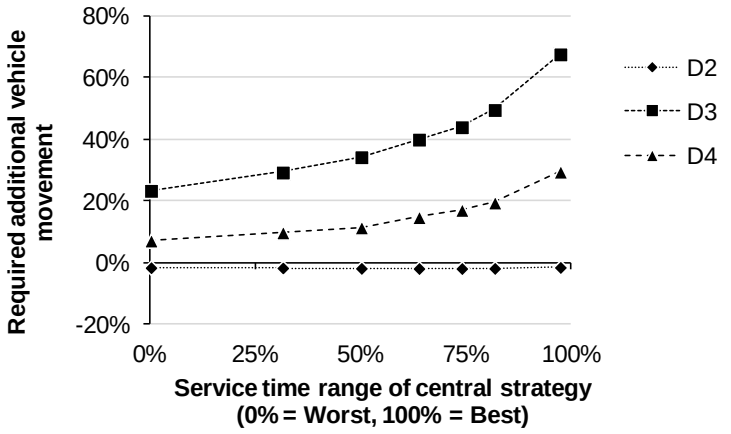



| Layout scenario 1 | Input data set 1 | Load elongation factor 1.0 |
|---|---|---|

Figure 49: Additional vehicle requirements for decentral strategies (experiments 1.1.x)

Adding more vehicles to the system also leads to additional vehicle movement. D2 is an exception in this context. As shown in Figure 45 the required vehicle movement does not depend on the number of vehicles for D2. All vehicles always return to their assigned home location when this strategy is used. Therefore the distance that laden and unladen vehicles cover is always the same. Adding more vehicles to the system only reduces the waiting times as there are always vehicles available at the corresponding storage locations when new loads arrive in the system. It leads to shorter service times but does not have an impact on the travelled distance.

The vehicle movement for D3 and D4 grows with the number of vehicles that are added to the system. For D4, vehicle movement and additional vehicle requirement increase uniformly. Below 60% of the central performance level D3 seems to achieve a better performance by more vehicle movement. The additional vehicle requirement is stable or even decreases slightly while the vehicle movement increases constantly in this corridor.



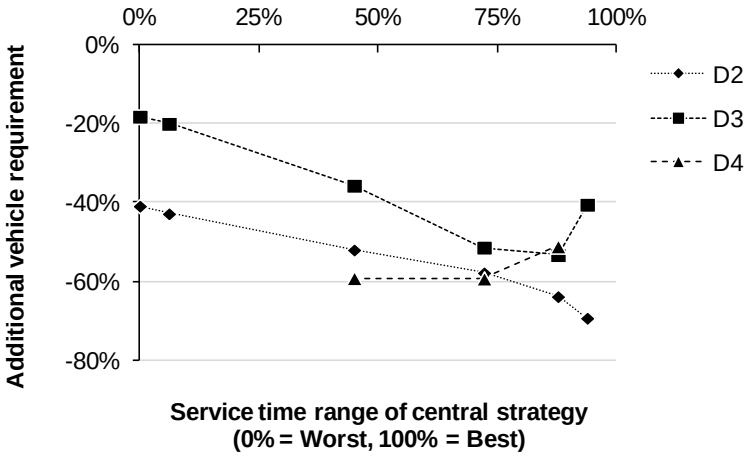

| Layout scenario 1 | Input data set 1 | Load elongation factor 1.0 |
|---|---|---|

Figure 50: Additional vehicle movement for decentral strategies (experiments 1.1.x)

The findings for the baggage handling data are completely contrary. The decentral strategies require fewer vehicles than the central benchmark:

- D2: -69% to -41% vehicles
- D3: -53% to -18% vehicles
- D4: -59% to -51% vehicles

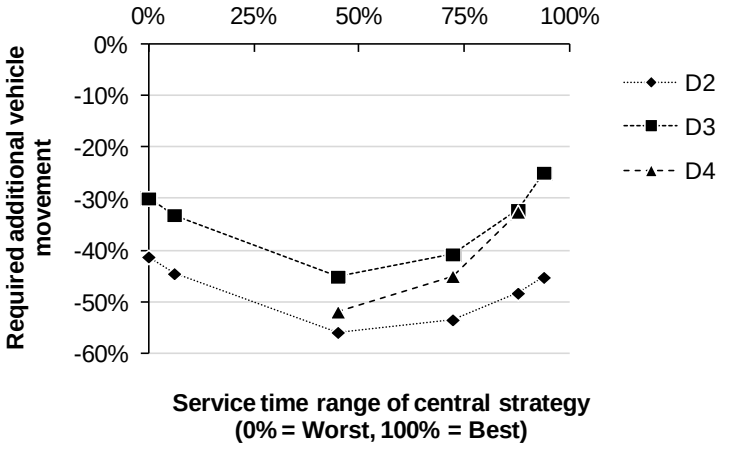
The interesting aspect is that for D2 and D3 the additional vehicle requirements decrease when a better performance ought to be achieved. This contradiction shows clearly that the central strategy does not perform very well in the given load scenario. It needs a high amount of vehicles before achieving a decent performance. D4 does not cover the whole performance range of the central strategy and is therefore only contained partly in the figure below. Its additional vehicle requirement only has a small range.




| Layout scenario 1 | Input data set 2 | Load elongation fator 1.2 |
|---|---|---|

Figure 51: Additional vehicle requirements for decentral strategies (experiments 1.2.x)

Using fewer vehicles for reaching the central performance level should result in less vehicle movement. This can be confirmed by the figure below. Besides showing the decrease of vehicle movement, it clearly reflects the peak of the travelled distance that D5 creates when more vehicles are added to the system (see Figure 48 and discussion above). As the performance of D5 is the benchmark, its characteristics influence the shape of the comparison curves. In contrast to the vehicle requirement the movement saving constantly declines for the upper 50% of the central performance.




| Layout scenario 1 | Input data set 2 | Load elongation factor 1.2 |
|---|---|---|

Figure 52: Additional vehicle movement for decentral strategies (experiments 1.2.x)

The key findings from this analysis of layout scenario 1 are as follows:

- The performance of the central benchmark dispatching strategy depends on the layout and the input data characteristics. Due to distance-based decision making the performance decreases when central dispatching is applied in combination with high throughput requirements and unfavorable load profiles at the different sources in the system. Early load-vehicle assignment and load-dependent source approach are additional factors which contribute to this effect.

- Depending on the input data characteristics (throughput, load profile at the sources) the decentral dispatching strategies require between 69% less and 45% more vehicles than the central strategy. In this simple layout scenario the decentral strategies are less vulnerable to the load profile differences than central dispatching. Therefore they require fewer vehicles than the benchmark strategy when input data set 2 is used. For input data set 1 the decentral strategies always require more vehicles to reach the central performance level.
- Similarly, the efficiency regarding vehicle movement is influenced by the load pattern. The only difference is a negative additional requirement for both input data sets when D2 is applied.
- The additional vehicle and movement requirement both depend on the central performance level that is to be achieved.

This layout scenario only allows a first rudimentary comparison of central and decentral control strategies. It is very simple and thereby influences the performance of the benchmark strategy. Consequently, the next section examines a bigger layout.

## 7.3 Layout scenario 2

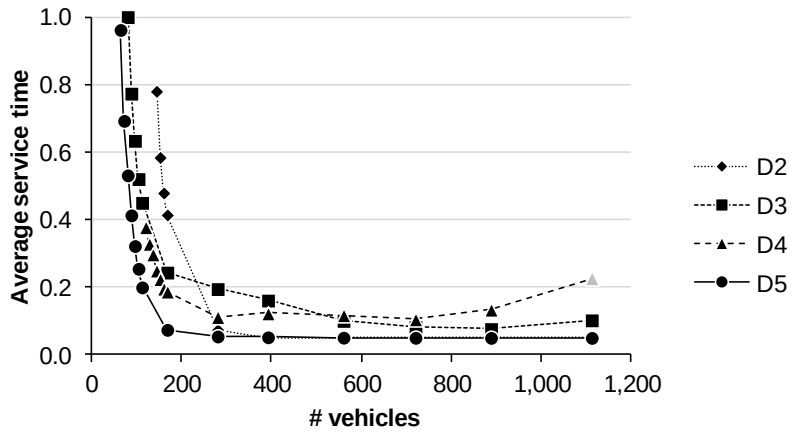
After the analysis of layout scenario 1, we will now turn our attention to layout scenario 2. The layout is far more complex than the single-loop system. The analysis starts with the determination of the minimal vehicles requirements and the resulting performance curves for input data set 1.

| Dispatching strategy | Simulation setup | Minimal vehicle requirement [# vehicles] |
|---|---|---|
| D2 | 2.1.2 | 144 |
| D3 | 2.1.3 | 80 |
| D4 | 2.1.4 | 120 |
| D5 | 2.1.5 | 64 |

Table 13: Minimal vehicle requirements for experiments 2.1.2–2.1.5

Dispatching strategy D5 requires the smallest amount of vehicles for a stationary system configuration. D3 needs 25% and D4 needs 88% more vehicles. D2 performs worse and requires more than twice as many vehicles as D5. While the first average service time for D3 is slightly above the central performance, the two other decentral strategies achieve 81% (D2) and 39% (D4) of it.

The figure below shows that the average service time performance of D5 is generally the lower boundary for all strategies. The shape of the curves is very similar to the results which were reported for layout scenario 1 and input data set 1 (see Chapter 7.2). All curves are very steep below about 200 vehicles. Above this value the performance improvements by adding more vehicles become smaller. D2, D3 and D5 all approach the same performance threshold value when the number of vehicles is increased. D3 shows a slight increase when more than 1,000 vehicles are in the system. Similar to the results in the layout scenario above, the average service time of D4 starts to increase once a turning point is passed (about 700 vehicles). For both strategies D3 and D4 the growing service times are caused by longer transport and waiting times.

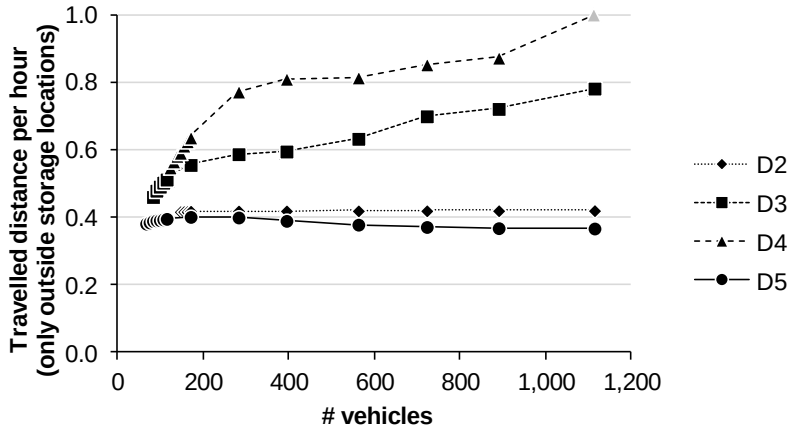



| Layout scenario 2 | Input data set 1 | Load elongation factor 1.0 |
|---|---|---|
| Configurations that do not fulfill all stationarity requirements are shown in grey. | | |

Figure 53: Average service times for simulation experiments 2.1.2–2.1.5

The increased transport times can be explained by Figure 54. The vehicle movement of D3 and D4 is far above the level of D2 and D5. Therefore mutual interferences among the vehicles are more likely to occur. D2 and D5 make better use of the storage locations and park idle vehicles.

D5 does not only achieve the best average service times, it also requires the least vehicle movement (when looking at similar vehicle configurations). While there are initially only very small differences between D5 and D2, the latter requires almost 15% more movement when more than 1,000 vehicles are used. The differences between D5 and the other two decentral strategies are far more severe. Compared to layout scenario 1, D4 now has the biggest movement requirements. It changes positions with D3 in this ranking. The same applies for D2 and D5.




| Layout scenario 2 | Input data set 1 | Load elongation factor 1.0 |
|---|---|---|
| Configurations that do not fulfill all stationarity requirements are shown in grey. | | |

Figure 54: Average travelled distances per hour for simulation experiments 2.1.2–2.1.5

After the analysis of the buffer replenishment data, we will now take a look at the same layout, but use the baggage handling data. In contrast to layout scenario 1, the average throughput difference between both input data sets is comparably small (see Chapter 7.2). The average throughput for input data set 2 is only about 11% above the level for data set 1. As in the case of layout scenario 1, the minimal

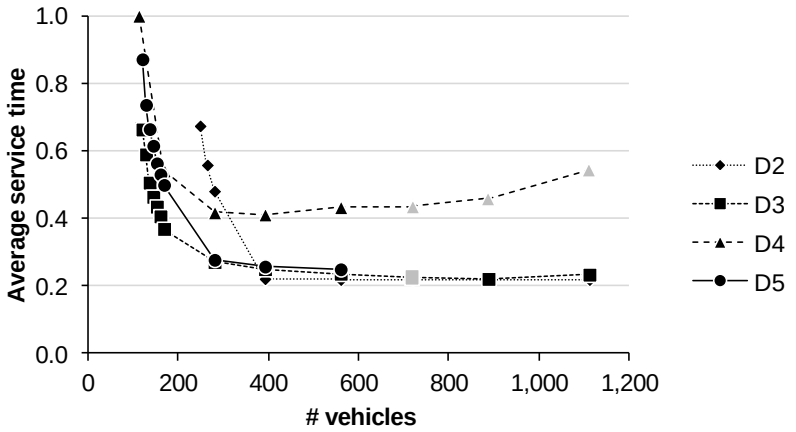
vehicle requirement shows that the throughput is not its only influencing factor. All strategies except D4 require far more than 10% additional vehicles. D2 needs 72% more, D3 50% more and D5 88% more vehicles. The first operational configuration of D4 requires 7% less vehicles than for input data set 1.

| Dispatching strategy | Simulation setup | Minimal vehicle requirement [# vehicles] |
|---|---|---|
| D2 | 2.2.2 | 248 |
| D3 | 2.2.3 | 120 |
| D4 | 2.2.4 | 112 |
| D5 | 2.2.5 | 120 |

Table 14: Minimal vehicle requirements for experiments 2.2.2–2.2.5

While the first resulting average service time for D4 is 15% above the central performance, D2 and D3 are about 23% below this level. The performance curves individually show similar characteristics as in layout scenario 1. A rather steep initial decrease is followed by a flat course which starts between 300 and 400 vehicles for D2, D3 and D5. But the figure below also allows the derivation of some new findings.

Firstly, D5 performs much better in layout scenario 2. It still does not represent the lower performance boundary as it does for the buffer replenishment data. But its performance runs very close to the curve of D3 which can be considered the lower boundary, at least up to about 300 vehicles. It is hard to judge whether this change is caused because D5 performs better or the decentral strategies perform worse. The decentral strategy D2 can be used to evaluate this question. It makes quite simple dispatching decisions and can be expected to show similar behavior in all layouts. While D5 scores worse than D2 in layout scenario 1 (with input data set 2), it outperforms D2 in this scenario below 300 vehicles. These findings support the hypothesis that D5 performs better in layout 2 than in layout 1. The layout-dependent performance limitations are not as severe as before when a more complex layout is used.




| Layout scenario 2 | Input data set 2 | Load elongation factor 1.7 |
|---|---|---|
| Configurations that do not fulfill all stationarity requirements are shown in grey. | | |

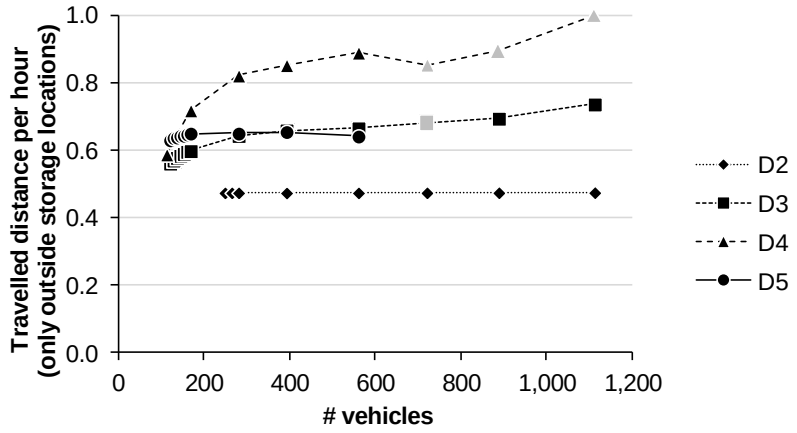
Figure 55: Average service times for simulation experiments 2.2.2–2.2.5

However, in contrast to input data set 1 central dispatching is not the lower performance boundary for input data set 2. This can still be explained by the load profiles at the different sources. For input data set 1 the sources with an even index (sources 2, 4, 6, 8) always show a higher throughput than the sources with an odd index. For input data set 2 it is the other way around. Especially for the sources 1–4 which have the highest demand, the throughput at the sources with an odd index is higher than at the corresponding sources with even index (source 1 vs. source 2 and source 3 vs. source 4). Similar to what has been said about layout scenario 1 these load profiles limit the performance of D5 when input data set 2 is used.

Another interesting aspect is the termination of the simulation runs for D5 when more than 560 vehicles are used. Mutual interferences of the vehicles become evident in this scenario. Deadlocks occur and the simulation runs cannot be completed. In contrast to D3 and D4 the load-dependent re-routing is not applied for central dispatching and therefore the central strategy can only deal with a limited amount of vehicles. As D5 has already reached a decent performance in the flat part of the curve before the deadlocks occur, this effect does not limit the further analysis.

Several vehicles configurations of D3 and D4 do not fulfill the stationarity requirements of both applied statistical tests. While D4 achieves a performance level that is only slightly above the lower boundary of the remaining strategies in the experiments 1.2.2–1.2.5, it does not reach this performance benchmark in the current scenario.

Regarding the travelled distance, D2 achieves the lowest requirement across all vehicle configurations. This is a similarity to scenario 1 when baggage handling data is considered. The remaining strategies require more movement than D2. But for configurations with few vehicles they also show a better performance regarding average service times. The biggest difference compared to scenario 1 can be observed for D5. It requires far less vehicle movement in scenario 2 and does not show a peak anymore. The performance of D4 decreases compared to the first scenario.



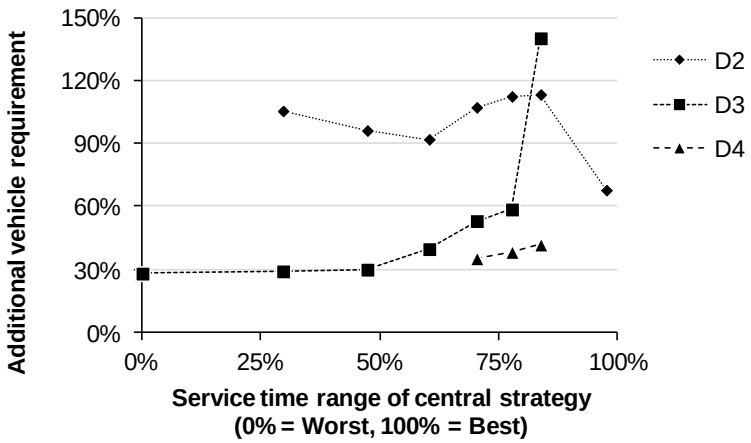

| Layout scenario 2 | Input data set 2 | Load elongation factor 1.7 |
|---|---|---|
| Configurations that do not fulfill all stationarity requirements are shown in grey. | | |

Figure 56: Average travelled distances per hour for simulation experiments 1.2.2–1.2.5

The remaining lines of this section take a look at the additional vehicle requirement for both input data sets. Similar to the evaluation for scenario 1, the additional vehicle requirements are stated in comparison to the range of the throughput times which the central benchmark strategy achieves. For the buffer replenishment data they amount to:

- D2: +68 to +113% vehicles
- D3: +28 to +140% vehicles
- D4: +35 to +42% vehicles

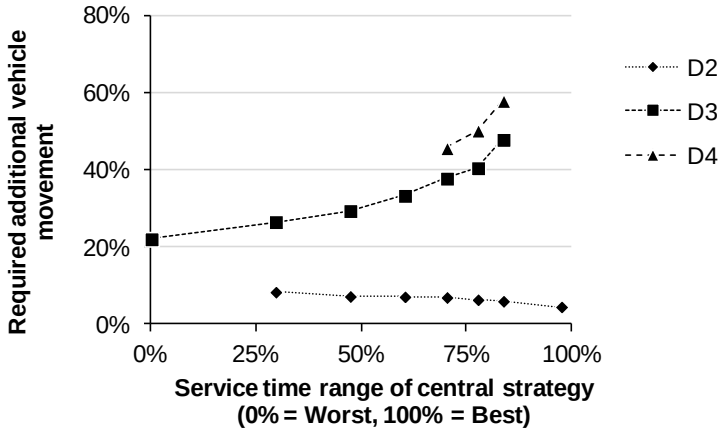
Except for the last value of almost 97.5% of the central performance range, D2 shows a rather stable additional requirement of around 100%. This stability is similar to what was found for layout scenario 1 in combination with the same data set. The shape of the curve for D3 also equals the results of scenario 1. A comparison of the results for D4 is difficult because this strategy only has a very small overlapping range with the central performance.




| Layout scenario 2 | Input data set 1 | Load elongation factor 1.0 |
|---|---|---|

Figure 57: Additional vehicle requirements for decentral strategies (experiments 2.1.x)

As in layout scenario 1 (with input data set 1), increasing the number of vehicles in the system for D2 does not have a major impact on the distance travelled. Generally, less than 10% additional vehicle movement is required. D3 and D4 require a comparably high amount of additional vehicle movement. While the additional movement requirement for D3 grows slower than the number of additional vehicles across the performance range, the contrary applies for D4. It should be noted that due to the steep initial decrease of the performance curves for the average service time, only a small share of the vehicle configurations is included in these analyses.



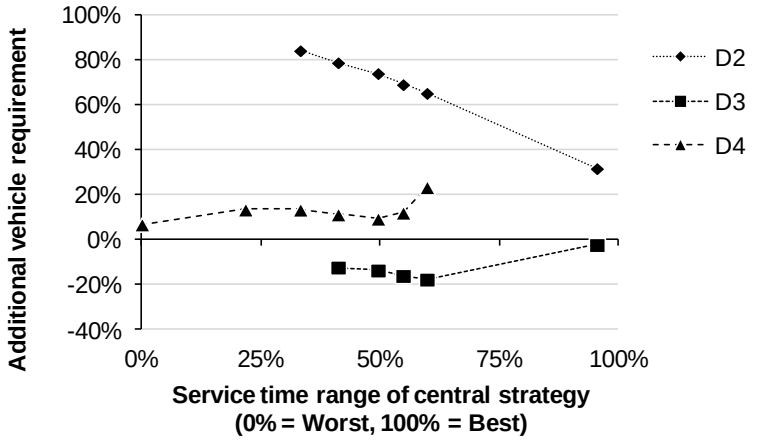

| Layout scenario 2 | Input data set 1 | Load elongation factor 1.0 |
|---|---|---|

Figure 58: Additional vehicle movement for decentral strategies (experiments 2.1.x)

The overall picture deviates when applying the baggage handling input data. In this case the following additional vehicle requirements can be derived:

- D2: +31% to +84% vehicles
- D3: -18% to -3% vehicles
- D4: +7% to +23% vehicles

In contrast to layout scenario 1, D2 and D4 require now more vehicles for the baggage handling scenario. However, both curves show roughly the same shape as in layout scenario 1. While the additional demand for D2 has a declining trend, the demand for D4 is comparably stable in the beginning and increases for higher performance levels. D3 still performs better than the central dispatching strategies and requires fewer vehicles to achieve the same performance. The differences tend to disappear for the upper 10% of the central performance range.

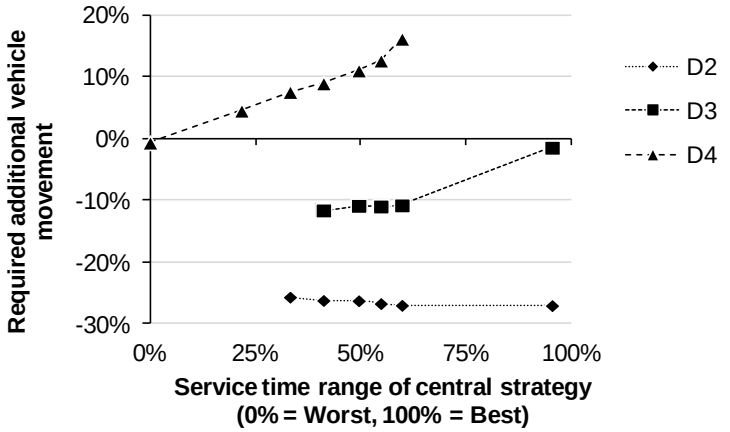



| Layout scenario 2 | Input data set 2 | Load elongation factor 1.7 |
|---|---|---|

Figure 59: Additional vehicle requirements for decentral strategies (experiments 2.2.x)

The additional vehicle movement is not biased by a peak in the travelled distances for D5 (see explanations for scenario 1 above). Therefore the requirements for the decentral strategies show the same shape as for input data set 1 (in layout scenario 2). The difference is that all strategies require

additional vehicle movement for input data set 1. In the case of data set 2, D2 and D3 require less movement than the central strategy. D4 is the only strategy with positive additional demand.




| Layout scenario 2 | Input data set 2 | Load elongation factor 1.7 |
|---|---|---|

Figure 60: Additional vehicle movement for decentral strategies (experiments 2.2.x)

Similar to layout scenario 1, the comparisons always depend on the performance that is to be achieved with the decentral strategies. The main findings based on layout scenario 2 are:

- As for layout scenario 1, the additional vehicle and vehicle movement requirement are influenced by the input load profile.
- D5 seems to perform better in the more complex layout, especially for input data set 2. The undesired effects which are caused by distance-based decision making and load-dependent source approach are less significant.
- The decentral strategies require between -18% and +140% additional vehicles. The overall range of the additional requirements is thereby wider than in layout scenario 1 and has been shifted to the positive direction of the y-axis.
- D2 and D4 have a positive additional vehicle requirement across the whole central performance corridor for both input data sets.
- D3 has a positive additional vehicle requirement for input data set 1 but needs fewer vehicles than D5 when input data set 2 is applied. Although the other decentral strategies already suffer from the increased layout complexity, D3 and its simple-minded decision rules are still able to cope better with the load profile characteristics of the baggage handling data than the benchmark strategy.
- The efficiency regarding vehicle movement generally depends on the used load scenario. The overall shape of the strategy-specific curves is very similar for both input data sets. Differences are mainly induced by the different performance of D5 in the two load scenarios. For input data set 2 increasing the number of vehicles in the system can lead to less vehicle movement when D2 or D3 are applied.
- D5 starts to suffer from vehicles interferences and deadlocks in this layout scenario. For D3 and D4 the load-dependent re-routing mitigates this risk.

Based on those results the next section reviews the most complex layout scenario which is considered in this thesis.

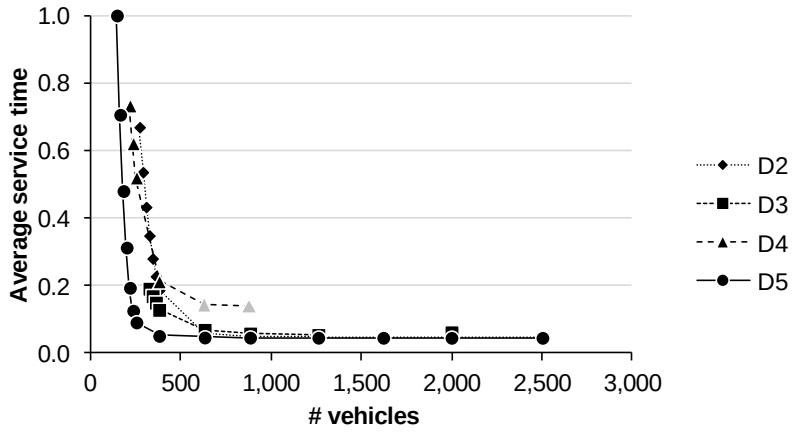
## 7.4 Layout scenario 3

The evaluation of layout scenario 3 completes our analysis of the simulation runs. Input data set 1 will be assessed first. The initial vehicle requirements for finding the first operational system configurations increase compared to layout scenario 2:

| Dispatching strategy | Simulation setup | Minimal vehicle requirement [# vehicles] |
|---|---|---|
| D2 | 3.1.2 | 270 |
| D3 | 3.1.3 | 324 |
| D4 | 3.1.4 | 216 |
| D5 | 3.1.5 | 144 |

Table 15: Minimal vehicle requirements for experiments 3.1.2–3.1.5

D3 needs the highest number of vehicles for the first operational system configuration. In return, the average service time of this configuration is more than 80% below the initial performance of the central strategy. D2 and D4 both require fewer vehicles. Their performance level is 33% (D2) and 26% (D4) below the initial benchmark service time. The general shape of the performance curves is similar to those for layout scenario 1 and 2. After a very steep decrease up to about 400 vehicles, a rather flat course can be observed. D5 again shows the best performance across all vehicle configurations and thereby defines the lower performance boundary. D2 and D3 reach the same boundary value when enough vehicles are added to the system. For system configurations with more than 1,000 vehicles D3 and D4 both struggle with the layout complexity. The simulation runs with 1,620 and 2,502 vehicles are canceled for D3. D4 performs worse. For configurations with 630 and 882 vehicles only one of the statistical tests favor stationarity. For configurations with more vehicles either the simulation is terminated or none of the tests confirms stationarity. These problems limit the performance level that D4 can achieve. This strategy would need more vehicles to reduce the average service time further.

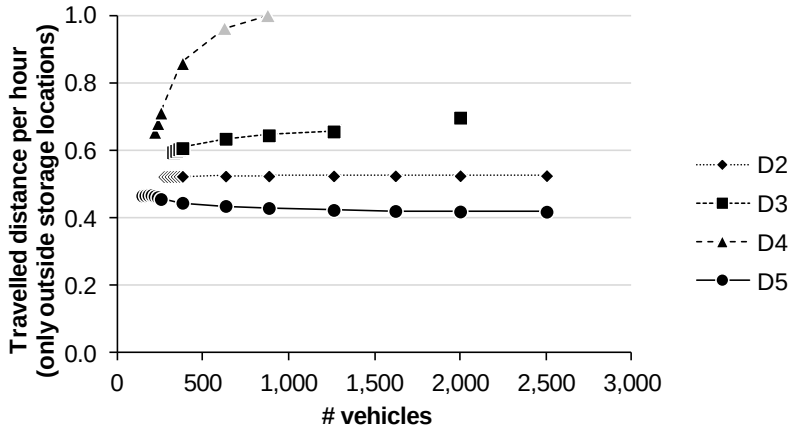



| Layout scenario 3 | Input data set 1 | Load elongation factor 1.8 |
|---|---|---|
| Configurations that do not fulfill all stationarity requirements are shown in grey. | | |

Figure 61: Average service times for simulation experiments 3.1.2–3.1.5

The travelled distance per hour behaves similarly to layout scenario 2 but reflects the increased layout size. The additional movement which D2 requires compared to D5 increases as the empty vehicles

now have to travel longer to return to their assigned home location. The differences between D3 and D5 also show a slight increase but are not as extreme as for D2. Due to the cancelation of several simulation runs only a few data points for D4 are available. Those which could be obtained show a far faster increase in the travelled distance than in layout scenario 2.




| Layout scenario 3 | Input data set 1 | Load elongation factor 1.8 |
|---|---|---|
| Configurations that do not fulfill all stationarity requirements are shown in grey. | | |

Figure 62: Average travelled distances per hour for simulation experiments 3.1.2–3.1.5

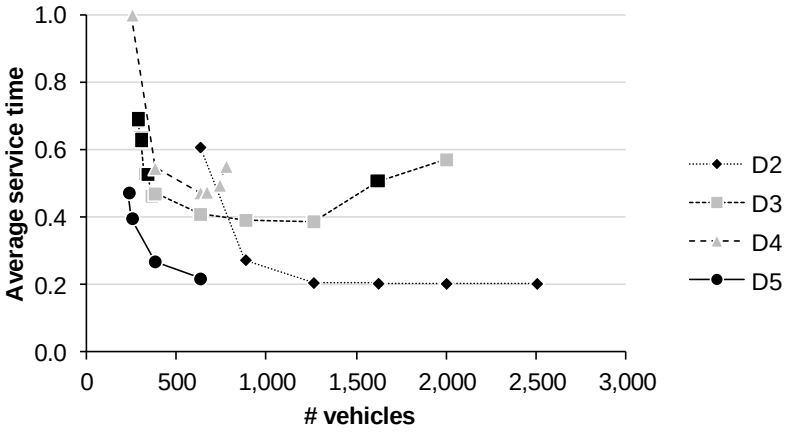
When layout scenario 3 is analyzed in combination with the baggage handling data, the impact on the minimal vehicle configuration is inconsistent. Compared to the buffer replenishment data the throughput is now about 14% lower. This is only reflected by D3 which needs 11% less vehicles (288 vs. 324). D2 and D5 now require 130% (630 vs. 270) and 63% (234 vs. 144) more vehicles for the first operational system. For D4 a slight increase of only 17% (252 vs. 216) can be observed. Compared to the other strategies D5 achieves a good performance with the first operational system configuration. The achieved average decentral service times are 28% (D2), 46% (D3) and 111% (D4) above the performance level of the benchmark strategy.

| Dispatching strategy | Simulation setup | Minimal vehicle requirement [# vehicles] |
|---|---|---|
| D2 | 3.2.2 | 630 |
| D3 | 3.2.3 | 288 |
| D4 | 3.2.4 | 252 |
| D5 | 3.2.5 | 234 |

Table 16: Minimal vehicle requirements for experiments 3.2.2–3.2.5

One exception had to be made with respect to the methodology for selecting the first operational vehicle configurations. The KPSS test does not favor stationarity for any of the vehicle configurations of D4. But as the Dickey-Fuller tests supports stationarity and the confidence intervals are rather small (< 19% relative error) we decided to include the results anyway in the figures below. But the stationarity limitations have to be kept in mind. No additional simulation runs were carried out to check if systems with fewer vehicles would lead to operational configurations.

The performance curves for the average service times show generally similar shapes as in the other two layout scenarios with the same input data sets. But for the first time D5 is the lower performance boundary. D2 and D5 both approach the same lower performance threshold value. Neither D3 nor D4 achieve this lower boundary. After a quick decrease of the average service time in systems with few vehicles, they both suffer from increased transport and waiting times when more vehicles are added to the system.



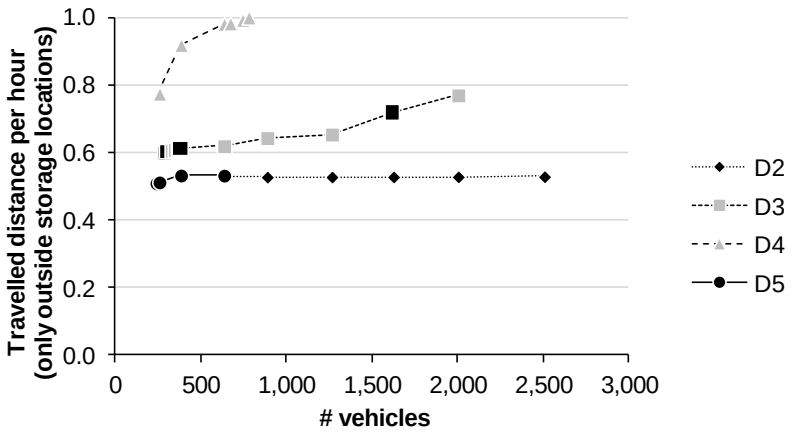

| Layout scenario 3 | Input data set 2 | Load elongation factor 3.0 |
|---|---|---|
| Configurations that do not fulfill all stationarity requirements are shown in grey. | | |

Figure 63: Average service times for simulation experiments 3.2.2–3.2.5

Although D5 achieves the lowest average service times with the least vehicles, it should be noted that out of the 10 initial runs only 3 deliver reliable results. Runs with less than 252 vehicles do not achieve the required throughput or stationarity. Runs with more than 630 are terminated due to deadlocks.

Similar findings can be reported for D3 and D4. D3 suffers from deadlocks for vehicle figures above 2,000 vehicles. For D4 the problems are even more severe. Deadlocks occur and lead to termination of the simulation runs for system configurations with more than 1,000 vehicles. In addition, D3 and D4 can only partly fulfill the stationarity requirements. D2 is the only strategy which does not run into the risk of deadlocks as the biggest portion of vehicles is parked at the storage locations when they are not used.

This effect of a high share of parked vehicles can also be recognized when looking at the average distance which the vehicles need to travel. The shape of the corresponding curve (see below) for D2 is very flat once the first operational system configuration has been found. In contrast to what could be observed for scenario 1 and 2, the required vehicle movement for D2 and D5 does not show a significant difference. D3 and D4 show the same behavior that has already been reported for the first of the two input data sets in combination with layout scenario 3.



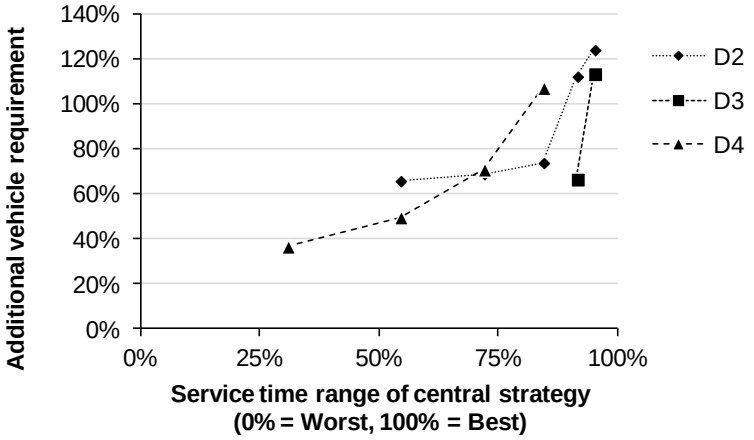

| Layout scenario 3 | Input data set 2 | Load elongation factor 3.0 |
|---|---|---|
| Configurations that do not fulfill all stationarity requirements are shown in grey. | | |

Figure 64: Average travelled distances per hour for simulation experiments 3.2.2–3.2.5

The evaluation of the additional vehicle requirements completes this analysis section. In contrast to layout scenarios 1 and 2 none of the decentral control strategies covers the whole bandwidth of the average service times which D5 achieves when input data set 1 is applied. For their covered performance range, the strategies require additional vehicles:

- D2: +66% to +124% vehicles
- D3: +66% to +113% vehicles
- D4: +36% to +107% vehicles

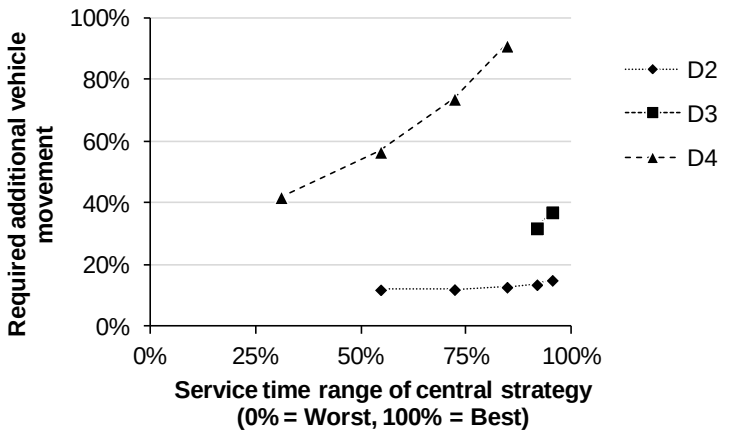
Compared to the other two layout scenarios it can be observed that the vehicle requirements of D2 are less stable. The roles of D3 and D4 seem to be mixed up in comparison with layout scenario 2. Here, D3 only covers a comparably small performance range. The same is true for D4 in the preceding layout scenario. D3 and D4 both show the growth trend that could already be observed in less complex systems.




| Layout scenario 3 | Input data set 1 | Load elongation factor 1.8 |
|---|---|---|

Figure 65: Additional vehicle requirements for decentral strategies (experiments 3.1.x)

The figure below indicates that for D4 the need for vehicle movement increases roughly proportional with the number of vehicles in the system. For D2 and D3 on the other hand, the growth of the additional vehicle movement is far smaller than the increase in vehicles. D2 shows the stable additional requirement across the whole central performance range which is already known from the other layout scenarios. It can be concluded that adding vehicles will reduce the response time, but many vehicles will have a rather poor utilization and mainly be waiting at a storage location or are used inefficiently.



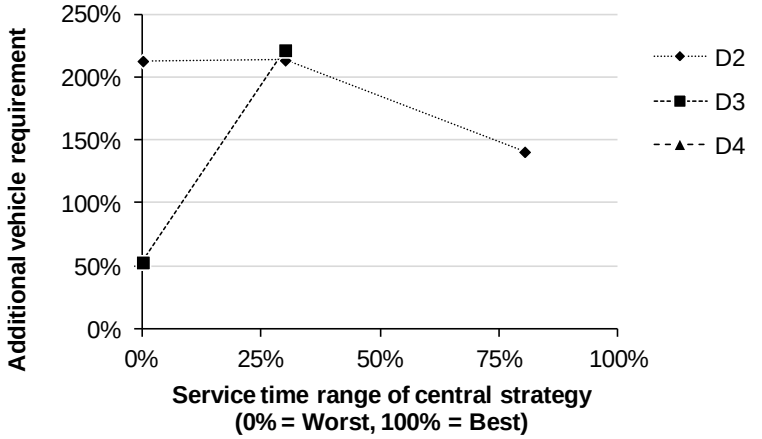

| Layout scenario 3 | Input data set 1 | Load elongation factor 3.0 |
|---|---|---|

Figure 66: Additional vehicle movement for decentral strategies (experiments 3.1.x)

The analysis of the baggage handling data for the first time does not reveal any strategy which requires fewer vehicles than the central strategy. The additional vehicle requirements are:

- D2: +141% to +213% vehicles
- D3: +53% to +221% vehicles

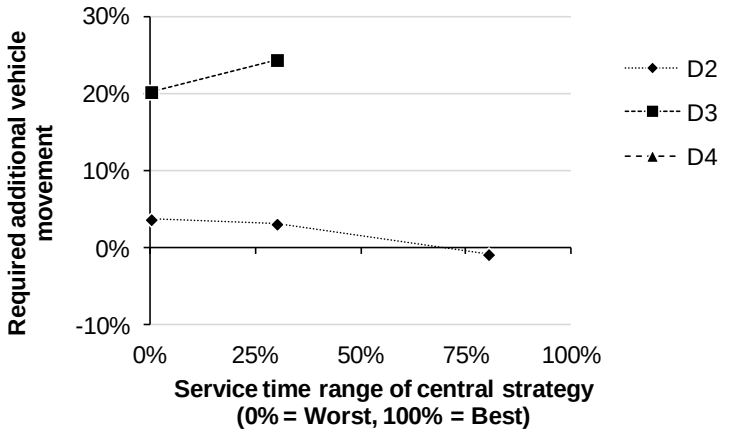
D4 could not be included in the analysis as its performance does not achieve the range of the benchmark strategy. For D2 the additional vehicle requirement slightly decreases when better performance levels are to be achieved. The results for D3 only cover the lower 30% of the performance of the central dispatching strategy.




| Layout scenario 3 | Input data set 2 | Load elongation factor 3.0 |
|---|---|---|

Figure 67: Additional vehicle requirements for decentral strategies (experiments 3.2.x)

The conclusions regarding the additional vehicle movement are very similar to what has been said about the first input data set above. The demand of D2 is stable and can nearly be neglected as all values are between -1% and +4%. The additional requirement of D3 is also stable and far lower than the additional vehicle requirement. Adding vehicles to the system does not have a major impact on the overall travel distance.




| Layout scenario 3 | Input data set 2 | Load elongation factor 3.0 |
|---|---|---|

Figure 68: Additional vehicle movement for decentral strategies (experiments 3.2.x)

As for the previous two layout scenarios the comparison of the performance is influenced by the input data set which is applied. Some insights should be highlighted:

- The benchmark strategy D5 outperforms the other dispatching algorithms for both input data sets.
- The range of additional vehicle requirements grows further (now between 36% and 221%) compared to layout scenario 2. It is shifted further to the positive direction of the y-axis. None of the decentral strategies needs fewer vehicles than D5.
- The strategies D3 and D4 struggle with the layout complexity. Deadlocks occur and the covered central performance corridor is quite small. D4 does not reach the central performance corridor for input data set 2.
- D5 also suffers from deadlocks but can already achieve the benchmark performance with relatively few vehicles.
- The efficiency regarding vehicle movement generally depends on the used load scenario. The application of decentral strategies involves far less additional movement for input data set 2 compared to input data set 1.

The analysis of the different layout scenarios has shown that a universal statement about the comparison of central and decentral performance is difficult. The results are case-dependent. Further analyses in the following subchapters shall provide some additional insights before drawing final conclusions.

## 7.5 Assessment of the dispatching strategies

Having reviewed and analyzed each of the layout scenarios, this section summarizes the main characteristics of each control strategy.

**D2 – static destination dispatching**

For four out of the six possible combinations of layout scenario and input data set the static destination dispatching rule has the highest vehicle requirement for reaching the first operational system configuration. At least one strategy needs more vehicles than D2 for the first operational configuration when layout scenario 1 is considered in combination with input data set 2 and layout scenario 3 is considered in combination with input data set 1. In the steep section of the performance curve diagrams the curve of D2 is usually the upper envelope for the curves of all other strategies. Only in the case of layout scenario 1 in combination with input data set 2 D3 and D5 perform worse than D2. The additional vehicle requirement shows a rather stable course for small layouts (scenario 1, 2) and input data set 1. For layout scenario 3 a slight increase can be observed. When D2 is applied to layouts with data set 2, the additional vehicle requirement shows a declining trend. The additional movement requirement is stable across the whole central performance corridor and usually between 0 and 10%. This is only different for layout scenario 1 and 2 in combination with the baggage handling data. Here, the additional movement requirement reflects the poor performance of the central strategy. Less movement would be required when using the decentral algorithm instead of the central strategy.

The behavior of D2 can mainly be explained by the fixed assignment of vehicles to storage locations. The control strategy is only operational when there are enough vehicles at each storage location to serve the demand at the corresponding source. As the same amount of vehicles is assigned to each source, this might result in an extreme imbalance of vehicle utilization in case the demand does not show similar patterns at each system source. This is also the reason why the initial vehicle requirements are much higher for the baggage handling data than for the buffer replenishment data. The transport demands of the former data set simply show a much higher variability. The source with the highest demand defines the number of required vehicles.

Once there are enough vehicles in the system to serve each source appropriately, adding more vehicles to the system simply reduces the waiting times. In the best case there would always be a vehicle available at the corresponding storage location, whenever a new load enters the system. Compared to the other decentral strategies, D2 requires only little vehicle movement. Except for layout scenarios 2 and 3 in combination with input data set 1, it performs even better than the central rule. Therefore the mutual interferences of the vehicles are kept to a minimum. An increase of the transport times can hardly be reported for configurations with many vehicles. Therefore the average service times stay at the lower boundary value once it has been reached. The risk of deadlocks is low. D2 is the only strategy that delivers stationary results for all simulation experiments with vehicle amounts above the minimal figure of the first operational system configuration.

**D3– forecast dispatching**

Besides D2, D3 is the decentral strategy which performs best. Except for layout scenario 3 in combination with input data set 2 it is able to achieve the lower performance boundary which is defined by D2 and D5. In fairly simple layouts (scenario 1 and 2) with variable load profiles (baggage handling) D3 performs better than the central strategy. The additional vehicle requirement shows the same increasing trend for all scenarios with input data set 1. For input data set 2 the course of the curve changes from declining (layout 1) to stable (layout 2) and finally to an increasing trend (layout 3). This means once D5 has reached a decent performance in layout scenario 3, the additional vehicle requirement curve for D3 shows the same shape as for all layout scenarios with input data set 1. The

additional movement requirement behaves similarly, but the relative changes are smaller compared to the vehicle requirement. In absolute figures, the distance travelled for D3 increases with the number of vehicles in the system and is between the curves of D2 and D4. Layout scenario 1 is an exception. Here, the vehicles have to travel the longest distances when D3 is applied.

D3 usually requires fewer vehicles than D2 to achieve the same benchmark performance level (layout scenario 1 in combination with input data set 2 is an exception). The usage of fewer vehicles is "bought" by more vehicle movement. Investment costs could be changed to operating costs. Due to the forecast and estimation procedure the dispatching process is less goal-oriented than for D2 and D5. It may happen that vehicles are sent to sources and storage locations without demand or would have better been sent to other locations which currently have a higher demand. Therefore the waiting times of the loads are longer than for the central strategy. During the implementation process it was found that release processes are required in order to ensure an operational system status. In addition, more vehicles are necessary to compensate the less efficient dispatching process. Less precise dispatching, release processes and additional vehicles lead to more vehicle movement. In return, the transport times grow as the vehicles start to mutually interfere. The deadlock risk increases and can only be mitigated by load-dependent routing that leads to a further increase of waiting and transport time. All these factors limit the performance of the decentral strategy. It is less predictable than D2 or D5. Besides deadlocks the lack of stationarity limits the expressiveness of the results. D3 is not able to cover the whole performance corridor of the central strategy for both data sets applied to layout scenario 3 and layout scenario 2 in combination with data set 2.

**D4 – feedback-based dispatching**

Among the decentral dispatching strategies, feedback-based dispatching performs worst. For the simple scenarios 1 and 2 the achieved average service times are still competitive. But for input data set 2 the strategy already fails to achieve the lower performance boundary in the second layout scenario. The same is true for both input data sets in combination with layout scenario 3. In this complex environment D4 does not reach the central performance corridor at all for the second input data set. The additional vehicle requirement shows an increasing shape for input data set 1. For input data set 2 a more stable or slightly increasing additional demand can be reported. The same statements are true for the additional vehicle movement. Looking at the absolute figures, the vehicles need to travel the longest distances when D4 is used for layout scenarios 2 and 3.

The behavior of D4 is even less predictable than for forecast dispatching. Deadlocks occur more frequently and the variability of service times is higher. Generally, the same problems that have been discussed for D3 are present. D4 performs even worse than D3 when it comes to the control of storage locations. Constant oversteering and understeering can be observed. As the number of destinations in the probability tables increases in complex layouts, the distinctive approach of a certain destination becomes less likely as the residual probability of choosing another destination grows. This decreases the precision of the dispatching process further and limits the system performance.

In addition, some other aspects have an impact. Due to the feedback procedure self-enforcing effects can occur that are hard to control. It is difficult to find a parameterization that achieves a decent system performance. A certain system balance needs to be found and even small changes can have a huge impact on the system. This is also true for D3. In particular sources in remote locations and those with sporadic demand are not served on a stable basis. It is always a stochastic process to detect the demand. As the feedback process is based on the waiting times, it is always delayed. The feedback cannot be given when the vehicle arrives at a dispatching destination but only after its departure. There might be long times in between. The given feedback does therefore not totally reflect the current system status.

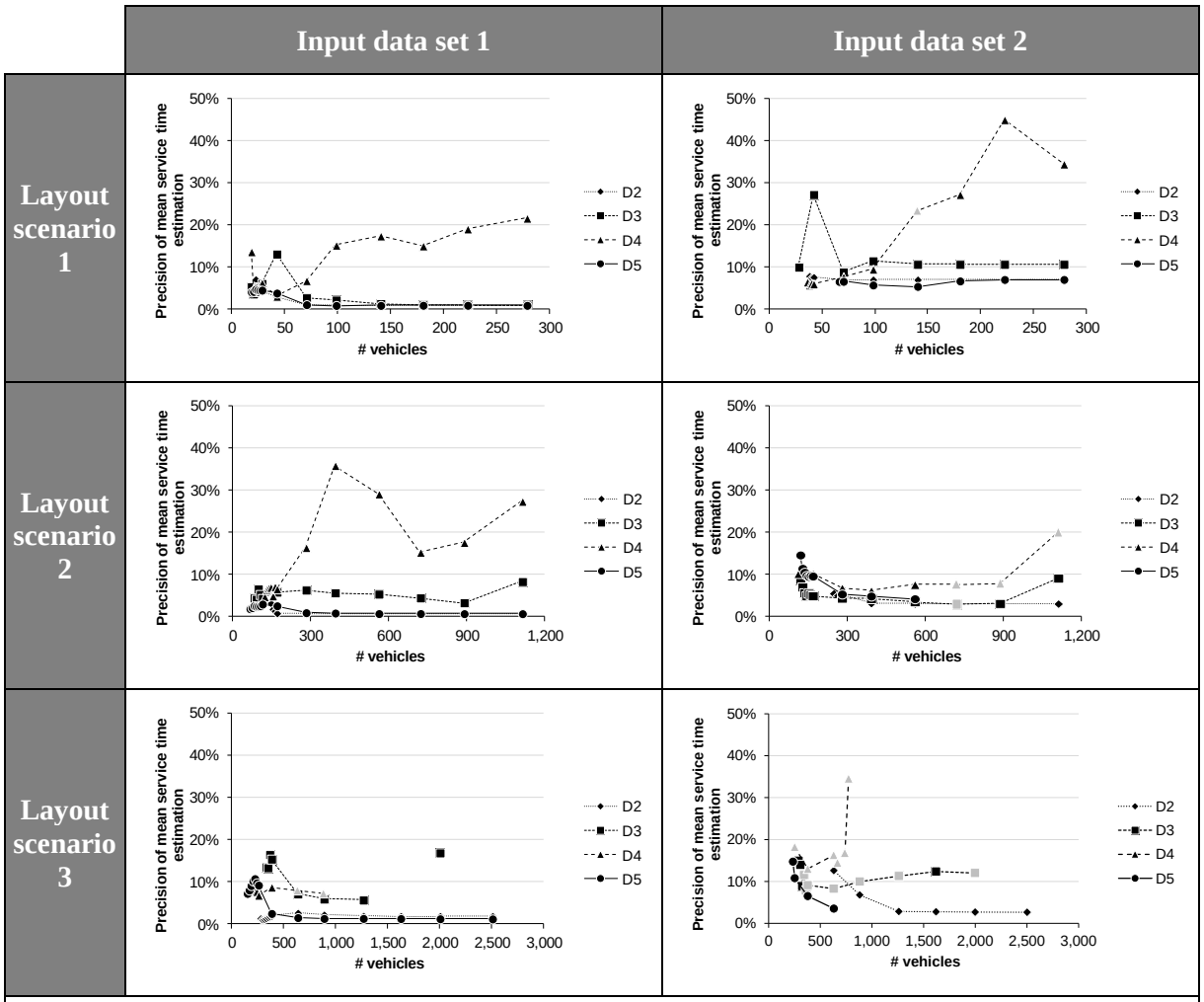
**D5– central dispatching**
As already discussed in the Chapters 7.2 and 7.3, the central dispatching strategy shows weaknesses when input data set 2 is used in layout scenario 1 and 2. In contrast to all other scenarios, the central performance is not the lower boundary for all performance curves in these cases. But the problems with distance-based decision making, early load-vehicle assignment and load-dependent source approach lose their significance in more complex layouts. However, another topic becomes relevant in this context. Deadlocks occur when central dispatching is used in layout scenarios 2 and 3 with the second input data set. One of the loops in the system gets blocked by vehicles. This does not have an impact on our analysis as the central strategy has already reached the lower performance boundary with fewer vehicles. But it needs to be kept in mind when developing control strategies. In one of the following sections we will show how load-dependent re-routing can help to reduce this problem and which consequences the application of this control strategy has.

## 7.6 Additional analyses

After the analyses of the different layout scenarios and the assessment of the control strategies, this subchapter aims at highlighting some additional aspects for the comparison of central and decentral control strategies. Firstly, the variability of the results is analyzed. Afterwards, implications of decentral strategies for the capacity of storage locations are derived. It follows an analysis of the application of load-dependent re-routing to central dispatching. Finally, different disturbances and their influence on the performance of the control strategies are evaluated. The results provide the stage for drawing final conclusions.

### 7.6.1 Confidence intervals and quantiles

We have argued above that all our analyses have to be treated with caution. They are based on average service times and do not take into account that the simulation experiments can only deliver an estimation of the mean. The width of the confidence interval for this estimation has to be considered. We have chosen the first operational system configuration with a precision of at least 15% for the average service time. The figure below shows how the relative precision develops when the amount of vehicles is increased beyond the first operational configuration. This does not only allow a judgment of the result reliability, but also shows the control strategy characteristics regarding service time variability.




Configurations that do not fulfill all stationarity requirements are shown in grey.

Figure 69: Precision of mean service time estimations

Generally, it can be concluded that for D2 and D5 the precision of the mean estimation is improved when more vehicles are added to the system. A second general observation refers to the widths of the confidence intervals. For D2 and D5 they are smaller for input data set 1 compared to input data set 2. These statements are true for all combinations of layout scenario and input data set except layout scenario 1 in combination with the baggage handling data. In this scenario a slight decrease in precision can be observed when more than 170 vehicles are in the system. The relative error of the estimations for D2 and D5 tends to be smaller than for D3 and D4.

D3 shows a predictable and stable precision across all considered scenarios. The only exception is a peak in layout scenario 1. It can be observed for both input data sets and can be explained by a higher variance in the service times which are achieved in these specific scenarios. These two exceptions are the only two cases when the relative error for D3 is bigger than for D4 (looking at similar vehicle configurations). Compared to D2 and D5 adding more vehicles to the system does not necessarily decrease the width of the confidence intervals, when D3 is applied. Slight increases are possible due to higher variability of waiting and transport times.

The precision of the results for D4 is less stable. Especially in layout scenario 1, D4 does not deliver reliable results for bigger vehicle amounts. Another interesting aspect is that D4 does not perform very well in layout scenario 2 when the buffer replenishment data is applied. The bigger width of the

confidence intervals can be explained by a higher variability of the output data. In addition, a higher correlation could contribute to this effect. In the case of the buffer replenishment data the load profiles for each day look very similar and the variance among the different sources is limited. For the baggage handling scenario there are bigger differences between the different sources during one day. The probability of finding repetitive patterns in the output data is therefore much lower. But as this effect can hardly be observed for D2 and D5, the variability of the output data has to be the major trigger. Generally, adding more vehicles to the system for D4 normally leads to an increase of the confidence interval width. This increase is bigger than for D3.

Concluding, D3 and D4 show a higher variability of service times and therefore have a lower precision of mean service time estimation. Their results are less predictable. It should be noted that these findings do not limit the expressiveness of our comparisons. Looking at the additional vehicle figures which are calculated (see Chapters 7.2–7.4), they usually refer to vehicle configurations with less than 30% of the maximum vehicle amount per scenario. In these sections of the performance curves the precision of the output data estimations for all decentral strategies is comparably good. The only exception is the peak for the second vehicle configuration when D3 is applied in layout scenario 1 in combination with input data set 2.

In addition to judging the pure variability of the data based on the confidence intervals, it would also be interesting to analyze the 0.95 quantiles of the service times. Those quantiles are often used as performance indicators in real-world intralogistics transport systems. However, analyzing this figure is not very meaningful in this study because we do not compare actual figures but standardize all comparisons on a scale between 0 and 100%. As a consequence purely showing the 0.95 quantile curves would lead to figures which are comparable to those which were created for the average service times. The general shape of the figures stays the same. We have therefore analyzed the quantiles with respect to the mean of the different strategies. However, this did not provide additional insights in the comparison of central and decentral control strategies. It simply shows again that the variability for D3 and D4 is higher than for D2 and D5. We therefore do not discuss these results here in detail. Similar conclusions can be drawn when the maximum service times are considered. D2 and D5 converge to the same lower boundary when the number of vehicles is increased (except for layout scenarios 1 and 2 in combination with input data set 2). If more than 50% of the maximum vehicle amount is used, D3 and D4 perform worse than D2 and D5 in all scenarios. Referring to what has been said about the precision of the mean estimation, the big differences regarding the quantiles and maximum service times do not have a major influence to the comparisons which were shown above. Only as small range of the decentral performance corridor is usually considered.

### 7.6.2 Usage of storage locations

The next in-depth analysis explores the quality of the dispatching decisions. We have argued above that the dispatching precision of D3 and D4 is comparably poor. The consequences of these decisions limit the achievable performance. This section tries to illustrate another facet of how the poor decision making influences the system behavior.

First, we look at the share of their time that vehicles physically spend at storage locations. It becomes evident that this analysis does not provide many additional insights. It simply supports the results from the comparison of the average service times. Therefore it is not necessary to discuss all combinations of layouts scenarios and input data. Only two exemplary figures are given below.

In the simple layout with the baggage handling data, D5 makes the vehicles travel a lot. They spend the least time at the storage locations. This changes in more complex layouts where the vehicles spend the most time in the storage locations when D5 is used. As could be expected, D2 is either itself the

upper boundary for the time which vehicles spend at the storage locations or its performance is at least close to this boundary. D3 and D4 are below the level of D2 for small layouts. In larger layouts particularly D4 does not achieve a decent time share of vehicle parking. This reflects the poor quality of the dispatching decisions regarding empty vehicle positioning. D3 does not perform as badly as D4, but still does not achieve the parking share of D2 and D5 in complex layouts. For all strategies and all layouts the time share spent at the storage locations increases when there are more vehicles in the system.

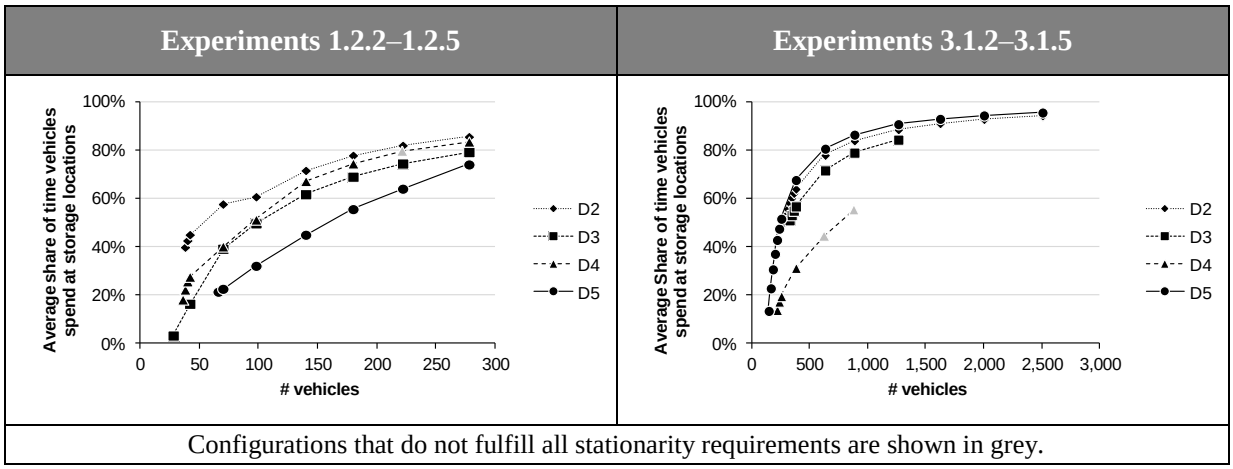



Figure 70: Two examples for time vehicles spend at storage locations

A second analysis is far more interesting. It considers the maximum number of vehicles that can be found at each storage location during the operation of the system. This analysis does not look at a certain point in time, but searches for the maximum values which were reached during the whole simulation run at any of the storage locations. One of our core assumptions is to use storage locations with a sufficient capacity. This approach does not have any implications for D2. A fixed number of vehicles is assigned to each of the storage locations. In our case the overall number of vehicles is equally distributed to the storage locations in the system. During operation, the maximum number of vehicles at certain storage locations will therefore never be above this initially assigned amount.

For D5 the situation changes slightly. As already mentioned above, due to the virtual list of vehicles at a storage location and the call-off process within the load-dependent source control strategy, it might happen that the physical fill level of a storage location exceeds its nominal maximum capacity. This effect can be observed in systems with many vehicles, i.e. in systems with poorly utilized vehicles.

The table below shows the 10 different initial system configurations and when the oversteering can be observed.

| Layout scenario 1 | | | Layout scenario 2 | | | Layout scenario 3 | | |
|---|---|---|---|---|---|---|---|---|
| # vehicles | Percentage exceedance | | # vehicles | Percentage exceedance | | # vehicles | Percentage exceedance | |
| | Data set 1 | Data set 2 | | Data set 1 | Data set 2 | | Data set 1 | Data set 2 |
| 2 | - | - | 8 | - | - | 18 | - | - |
| 14 | - | - | 56 | - | - | 126 | - | - |
| 28 | - | - | 112 | - | - | 252 | - | - |
| 42 | - | - | 168 | 10% | - | 378 | 24% | 29% |
| 70 | - | - | 280 | 51% | 106% | 630 | 57% | 134% |
| 98 | - | - | 392 | 31% | 171% | 882 | 51% | - |
| 140 | 10% | - | 560 | 34% | 146% | 1,260 | 50% | - |
| 180 | 17% | 14% | 720 | 33% | - | 1,620 | 51% | - |
| 222 | 13% | 23% | 888 | 30% | - | 1,998 | 45% | - |
| 278 | 9% | 35% | 1,112 | 26% | - | 2,502 | 33% | - |

Table 17: Maximal percentage exceedance of maximum storage location capacity for D5

The analysis shows that D5 is not able to perfectly deal with system configurations that contain many vehicles. The maximum storage location capacity is exceeded in several cases, and in one case up to 171%. Looking only at the input data it is impossible to predict at which of the sources the highest fill level can be expected. The effect seems to depend on the layout and the demand patterns. It is more likely that the defined maximum capacity $s_i^{max}$ is exceeded at storage locations where the corresponding source has a high demand. The maximum number of vehicles which can be reported for any of the storage locations is summarized in the figures below. These figures illustrate also that the oversteering effects for D3 and D4 are far more severe than for D5.

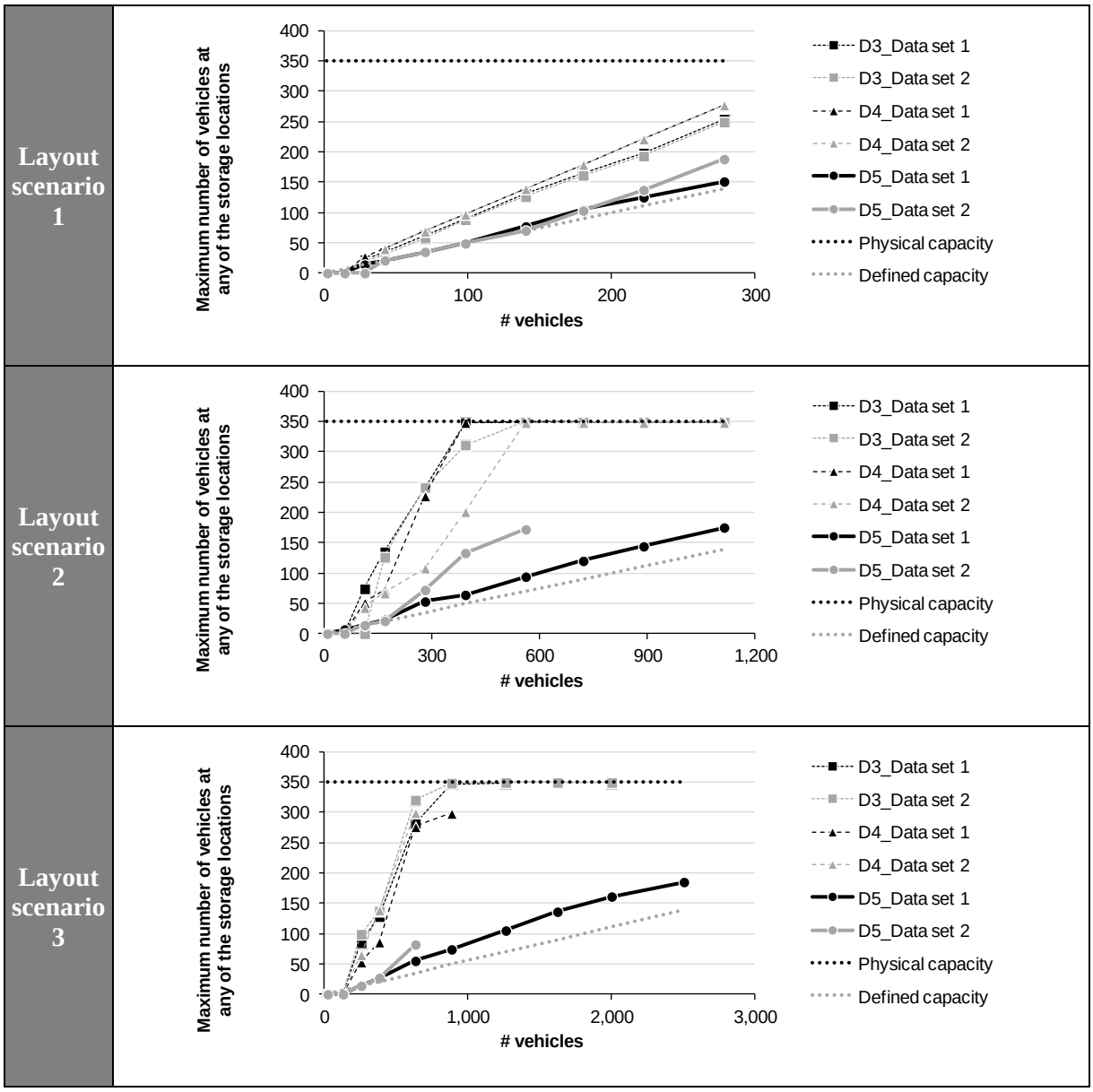



Figure 71: Maximum number of vehicles at any of the storage locations

For D3 and D4 the maximum number of vehicles at any of the storage locations usually exceeds the defined nominal storage location capacity $s_i^{max}$. But for layout scenarios 2 and 3 even the physically available storage location capacity is not sufficient when more than 50% of the maximum vehicle amount is used for experiments. Congestion might occur due to vehicles which are not able to enter the storage location they have been sent to and block the major driveway. These blockages influence the achievable service times, but their impact should not be overestimated. Firstly, there can also be reported an increase of the service times for layout scenario 1 when many vehicles are used. This increase occurs although the physical storage location capacity is not exceeded. The mutual interferences of the vehicles are much more important. In addition, the blockages mostly occur during periods with low system load when there are only a few transport requests in the system. The impact on the overall average service time is therefore limited. As soon as the system load increases, a faster vehicle circulation is required and the blockages disappear. Finally, the exceedance can only be observed for systems with many vehicles.

The number of storage locations with exceeded capacity grows with the number of vehicles in the system. Depending on the simulation setup between one and four storage locations are affected. Usually, the exceedance does not occur simultaneously but changes between the storage locations over time. In more complex systems it is less likely that all vehicles accumulate at a certain source. For all system configurations it was found that the more vehicles there are in the system, the higher the variance of the maximum figures at the different storage locations. Similar to what has been said about D5, the exceedance is hard to predict. The effect depends to a certain extent on the demand at the corresponding sources, but is not limited to this input factor. It might occur at sources with very high or very low demand.

The central strategy tries to distribute the vehicles equally during idle periods. This is the best guess as future demands are by definition unknown. The distribution of vehicles is always affected by previous demand patterns for decentral dispatching. The accumulation of vehicles at one or few sources makes the decentral systems more inert and contributes to the efficiency loss.

These analyses have shown that D3, D4 and D5 have weaknesses regarding the control quality of the storage locations fill levels. These effects are more severe for the decentral strategies than for the central algorithm. While D5 only exceeds the predefined maximum capacity, D3 and D4 send more vehicles than physically feasible to storage locations if many vehicles are used. In almost all cases D3 and D4 require a higher storage location capacity than the central algorithm. At the same time they are not able to achieve a decent share of vehicle parking in complex layouts. The results illustrate the problems that D3 and D4 have with precise empty vehicle positioning. Due to the usage of local knowledge for the dispatching process the current fill level is not known for sure and cannot be considered accordingly. For D4 the accumulation of vehicles at a certain source is sometimes the consequence of reinforcing feedback processes. The exceedance of the capacity has only a limited impact on the achievable service times. It has rather implications for the layout of the transport system. In a real-world system storage locations with high capacity would be required to successfully implement decentral strategies. As this is not economically viable, additional strategies will be required to mitigate the exceedance effects. A circulation or re-routing of vehicles could be first ideas.

### 7.6.3 Load-based re-routing

During the analysis phase, it was found that D5 does not deliver results for all vehicle configurations when layout scenario 2 and 3 are used in combination with the baggage handling data. Therefore the idea of applying the same load-based decentral re-routing strategy as for D3 and D4 came up. Layout scenario 3 is used as an example to present the consequences of combining central and decentral control strategies.

The results indicate that applying the additional intersection control strategy helps to distribute the system load and to prevent deadlocks (see figure below). Operational system configurations could be found up to the maximum vehicle amount of 2,502. The decentral re-routing does not significantly affect the average service time for the system with 252 vehicles. Slight increases become visible for systems with 378 (+2.6%) and 630 (+8.2%) vehicles. The travelled distance does not even show half of this percentage increase for the two vehicle configurations.

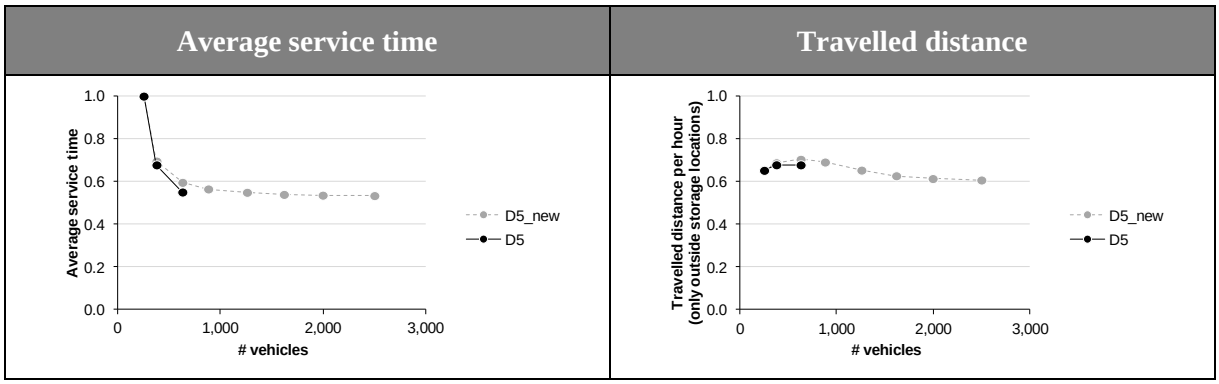



Figure 72: Impact of load-dependent re-routing for D5 in experimental setup 3.2.5

There is no reference data available for systems with more than 630 vehicles. But the fact that operational system configurations could be found demonstrates the effectiveness of the decentral intersection control strategy. Because of its simplicity it might also be an appropriate tool in other environments. Of course, it has to be considered that the impact always depends on the system layout. In the loop structure of the test system taking an alternative path does not result in an extreme increase in the transport distances and times. This might be different in real-world systems. Airport baggage handling systems sometimes connect terminal buildings via tunnels. In this case re-routing to an alternative path might have a far more severe impact and can make the load-dependent re-routing less appropriate.

### 7.6.4 Disturbance scenarios

The stable operating conditions which were assumed for the simulation experiments in this thesis so far are unlikely to exist in a real-world scenario. Interruptions due to vehicle or path downtimes are common in material handling systems (VAN DER MEER 2000). It is therefore necessary to analyze how central and decentral control strategies can cope with this kind of disturbance. Adaptivity and robustness to disturbances are seen as some of the major advantages of decentral strategies. But they are in most cases only stated as qualitative characteristics. The following experiments try to quantify them. The analysis is not limited to temporary vehicle and path breakdowns. In addition, the impact of extensions and reductions of the system size is evaluated. This relates to the requirement of easy reconfigurability and thereby another dimension of robustness. The following sections first define three different disturbance scenarios:

- Path/vehicle failure
- System extension
- System size reduction

Their implications for the AutoMod implementation are stated. Afterwards the simulation methodology and the results from the simulation runs are discussed.

All analyses are based on layout scenario 2 in combination with input data set 2. The layout has been chosen because it offers a decent degree of complexity. It is more complex than scenario 1 but on the other hand still makes sure that the effect of the local disturbances can be recognized on a global scale. For bigger layouts it might happen that a local disturbance only has a limited impact on the overall system performance. The consequences of the different disturbance scenarios would be much harder to trace. The baggage handling data set has been selected as the results from the previous experiments indicate that it is the bigger challenge for the control strategies.

Only D3, D4 and D5 are included in the following analyses. Whenever a source or storage location is added or taken off the system, D2 would require additional emergency strategies as there is a fixed assignment of the vehicles to the storage locations. The current implementation is not able to deal with all of the disturbance scenarios. The analysis could not evaluate the performance of the currently implemented algorithm but would require new strategies. Therefore D2 is not considered.

Regarding the central dispatching strategy two lines of thought are possible. One the one hand, it could be argued that a central control system is not operational after the occurrence of a disturbance. This relates to what has been mentioned in the introductory chapter of this thesis. A central control system has a single point of failure and can only make meaningful decisions when it knows about the status of the overall network. This is especially a problem when a path or vehicle failure occurs. Similarly, adding new components to the network or taking nodes off the network would not be easily feasible in a real-world environment. Both events would require a major modification of the central control system. Its rigid structure makes changes during operation difficult. Based on these aspects the following analyses would not be reasonable for the central system. But in order to compare the influence of local to global information the central system is required as a benchmark. Therefore we will follow the second line of thought and assume that the central system has global knowledge of the system status. It is able to sense the changes which are caused by the disturbance and reacts accordingly.

**Path failure / vehicle failure**

Depending on the technical features of the considered transport system path failure and vehicle failure are two different disturbance types which lead to the same consequences. In a carrier-based conveyor system a path failure is a technical breakdown which makes a path impassable. Logistics entities which are currently on the path are stopped and approaching ones cannot access the path. A path failure could also be interpreted as congestion on a certain path.

In vehicle-based systems a vehicle failure occurs when a vehicle has a technical defect and is not able to proceed to its current destination. Unless emergency strategies are in place (e.g. other vehicles pass the stopped vehicle) the vehicle failure leads to the same consequences as the path failure for the carrier-based system. Following our understanding of transport systems in this thesis, we treat both failure types as variants of the same disturbance scenario and only use the term “path failure”.

For the analysis we consider a path failure of the path which connects the lower with the upper left basic loop of layout scenario 2 (see figure below). The disturbance occurs at time $t_{dist}$. After a certain time period the path failure is fixed at time $t_{norm}$ and the system returns to standard operating conditions.

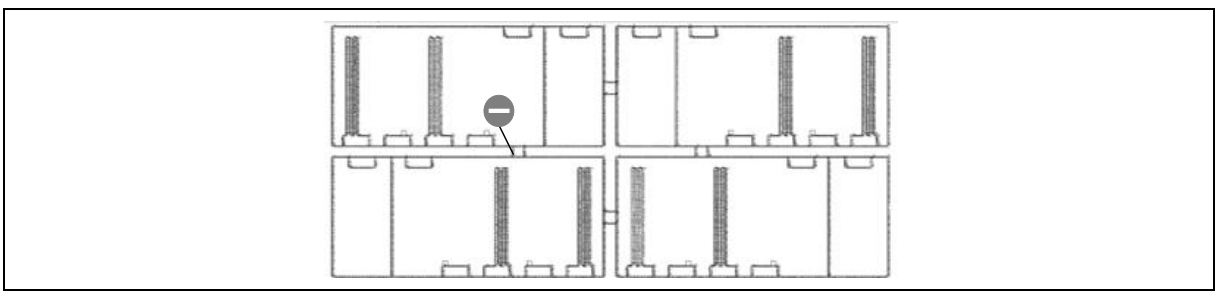


Figure 73: Location of path failure in layout scenario 2

For modeling the path failure in AutoMod an additional control point has been placed on the disturbance path. Its capacity is set to 0 for the time of the disturbance. Approaching vehicles have to stop because they cannot claim this next control point.

The path breakdown stops vehicles which want to travel from the lower to the upper loop. A control strategy is required which re-routes the vehicles. Otherwise they would simply accumulate behind the stopped vehicles and block an entire part of the system. Blocked parts of the system reduce the performance significantly. An analysis of the impact that path breakdowns have on the dispatching performance would become difficult. It can be assumed that re-routing strategies would be available in real-world systems. At first sight, the load-dependent re-routing strategy should be able to deal with the path breakdown. But a detailed look reveals that it is not. The algorithm recognizes the path breakdown based on the path utilization. The latter increases when vehicles are sent to the broken path and cannot continue due to the technical failure. Once the utilization is higher than the threshold value, vehicles are properly rerouted to the alternative direction. But the procedure shows a weakness as soon as there is also congestion on the alternative path. In this case the broken path might become the preferred solution again and the system could be blocked.

Additionally, re-routing due to a path breakdown would lead to a constant circulation of some vehicles within the lower left loop of the system. Vehicles that are re-routed due to the path breakdown proceed to the next switch within the same loop. But as the routing tables have not been updated, the next switch sends the vehicle to the switch which is located in front of the path breakdown again. At this point, the same process starts over again.

These two problems show that the load-dependent re-routing only has limited abilities to deal with path breakdowns. Implementing additional local decision rules would be an option. But as the consequences are hard to predict and the major focus is dispatching rules, we focus more on our core assumption of an efficient routing algorithm. All routing tables are updated once the path breakdown occurs. Accordingly, the vehicles will not choose the affected path anymore. Vehicles which were approaching the path already or are travelling on the path need to stop and can only proceed after the path failure is resolved.

The switch that is located in front of the failure path loses its core functionality during the breakdown. No decisions about selecting one or the other direction have to be made. Vehicles are simply forwarded and not sent to the path failure. For D4 the specific switch keeps its network knowledge during the breakdown. It receives feedback from vehicles which it dispatched before $t_{dist}$. After $t_{norm}$ the switch is reintegrated into the standard dispatching processes.

**System extension**

As a system extension we consider adding more entities to a transport network. The entities could either be vehicles, paths, sources (and storage locations) or sinks. Increasing the number of vehicles is already covered in this study. Adding new paths would affect the routing rather than the dispatching strategy. Out of the two remaining options, we choose the network extension by adding new sources to a system. It seems in connection with the required corresponding storage locations to be the more complex disturbance.

Technically, the extension can either be interpreted as physically adding a new set of source and storage location to the system or simply as a delayed start of the load arrival at a source which is already contained in the system.

The AutoMod implementation follows the first of the two options. Although it already contains the complete layout, two of the sources and storage locations only become accessible after $t_{dist}$ (see

Figure 74). Accordingly, the demand patterns at the two sources start at this point in time. Loads which arrive at those sources before $t_{dist}$ based on the input data set, are simply discarded. The accessibility of sources and storage locations is controlled by additional control points. Their capacity can be changed similar to the path failure scenario above. The added sources and storage locations are not part of the routing tables before $t_{dist}$. When the new sources have the first demand, the routing tables are updated. The sources become accessible for the vehicles.

As introduced above, the vehicles in each simulation are equally distributed to the storage locations at the beginning of the run. We stick to this procedure. Depending on the used dispatching strategy some modifications are required. For both D3 and D4 the vehicle release process is now applied at the extension sources and storage locations right at the beginning of each simulation. This makes sure that the vehicles leave the storage locations and travel randomly to other storage locations or sources in the system. Once they have left the storage locations, they are not allowed to enter them again until the first demand at the extension sources is available in the system. For D5 the vehicles are automatically pulled away from the storage location when the demand in the systems becomes high enough. Afterwards the extension storage locations are not considered in the distribution of empty vehicles until the first demand at the extension sources is available in the system. The impact that the reduced number of sources has on the nominal maximum storage location capacity is neglected before $t_{dist}$.

Once the sources are accessible after $t_{dist}$, their demand is recognized by D5. The strategy has global information. Afterwards the sources and corresponding storage locations are included directly in the load-vehicle assignment and empty vehicle positioning procedures. For D3 and D4 it is a stochastic process that makes randomly travelling vehicles arrive at the sources and storage locations which were added to the system. The sources and storage locations are not known in the system until the vehicles move there by chance and afterwards spread the information about the new dispatching locations in the system. Therefore two different versions of the disturbance process will be analyzed for D3 and D4. The first version assumes that the information about the new network layout is only spread by the vehicles according to the standard dispatching process. The second version assumes that the updated network information is propagated in the network at $t_{dist}$. The initialization of the decision variables is as follows for the second version of the dispatching algorithms:

- D3: The new dispatching locations are added to the list of known dispatching locations at each sink. The demand of the new sources is initially set to 0 and the fill level estimation for all new storage locations is set to 1,000 for each local forecast.
- D4: The new dispatching locations are added to the list of known dispatching locations at each switch. The decision probability is spread equally to all known dispatch locations for each dispatching table.

For the central dispatching strategy there exists only one version of the algorithm as the availability of global information is a prerequisite for its application.

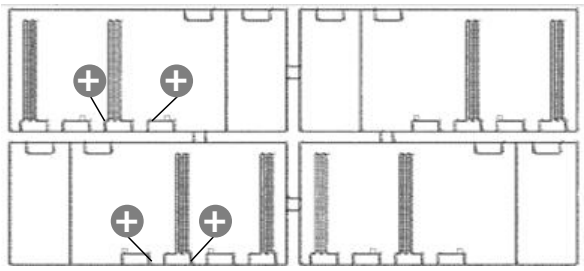
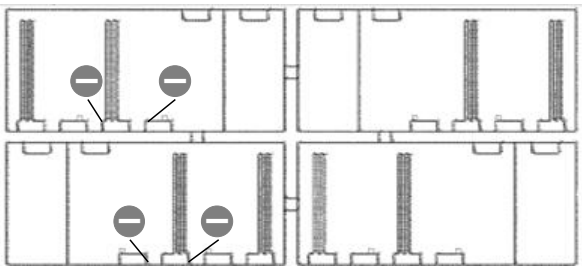
| System extension | System size reduction |
|---|---|
|  |  |
| Location of loop sidings that become accessible at $t_{dist}$. | Location of loop sidings that are no longer accessible after $t_{dist}$. |

Figure 74: Positions of disturbed sources and storage locations in layout scenario 2

**System size reduction**

Similar to the discussion for the system size extension, the relevant options for its reduction are either taking sources and storage locations or sinks off the system. The breakdown or removal of a sink would always have to be covered by an emergency strategy. A destination is no longer available and a decision about alternative delivery locations is required. This has only a limited impact on the dispatching decisions. We therefore again choose to focus on sources and storage locations. Technically, the system size reduction can be interpreted as physically removing sources and storage locations from the system or as stopping the sources' demand patterns at $t_{dist}$.

For the disturbance analysis it is assumed that the demand patterns at two of the sources in the system stop at $t_{dist}$ (see Figure 74). The sources and the corresponding storage locations are no longer accessible for vehicles afterwards. Vehicles which were waiting at the source or storage location have to stay there. The basic AutoMod implementation logic is similar to the extension scenario. The routing tables are updated at $t_{dist}$. For D3 and D4 the same two versions of information propagation are analyzed.

However, the reduction of the system size has several implications which lead to further adjustments of the AutoMod simulation model. For D3 and D4 the following problems occurred and were solved:

| | Problem | Implemented solution |
|---|---|---|
| 1 | • After the reduction of the system size vehicles are dispatched to locations which are no longer in the system and therefore are not contained in the routing tables. | • Affected vehicles continue to travel randomly. |
| 2 | • Vehicles might be "lost" at sources and storage locations which have been taken out of the system. | • None, this situation is accepted. |
| 3 | • Vehicles are released while the source or storage location is taken out of the system. | • Vehicles simply proceed to the successor of their current position. Afterwards they travel randomly. |

Table 18: Adjustments for D3 and D4 in system size reduction scenario

Similarly, several modifications to dispatching strategy D5 were required:

| | Problem | Implemented solution |
|---|---|---|
| 1 | • After the reduction of the system size vehicles are dispatched to locations which are no longer in the system and therefore are not contained in the routing tables. | • Based on their current position the vehicles use the central empty vehicle positioning algorithm to select a new storage location. |
| 2 | • The amount of vehicles stays the same after the system size reduction, but less storage locations are available. | • The maximum capacity for the remaining storage locations is adjusted. |

Table 19: Adjustments for D5 in system size reduction scenario

To deal with the locations which are no longer part of the system is the key aspect of the reduction scenario. For D5 the affected sources are automatically no longer considered in the load-vehicle assignment process as there is no longer a demand. The implementation makes sure that the corresponding storage locations are no longer taken into account for empty vehicle positioning.

For D3 and D4 the situation is more complicated and depends on the type of information propagation which is chosen. If the new network knowledge is not spread when D3 is applied, over time less and less vehicle are dispatched to the sources and storage locations which are extracted from the system. As there is no longer a demand, the sinks do not perceive a vehicle requirement. The forecast process at the sinks will therefore quickly converge to a vehicle requirement of 0 for future periods. At the same time there is no feedback about the storage location fill levels as no vehicles arrive from these storage locations. The estimated fill levels therefore grow to the upper limit of 999,999. If the knowledge of the changed network structure is propagated, similar consequences are initiated, but there is no transition phase. The sources and storage locations are simply removed from the decision tables and from one point in time to the next, no vehicles are dispatched to these locations anymore.

The situation is different for D4. If the new network structure is not propagated, sources and storage locations remain in the dispatching tables. In contrast to D3, there will always remain a residual probability for selecting those locations as the sum of all probabilities in the dispatching table has to equal 1. Although the residual probability is likely to be very small, it will never be 0 and might even grow when other locations receive a high penalty from the feedback functions. For D4 spreading the new topology information is therefore the only method which makes sure that no vehicles are dispatched to the locations which have been taken out of the system. These predicted behaviors are to be verified by simulation results.

**Simulation approach**

In contrast to the simulation experiments which were carried out above, we are not interested in calculating overall KPI for the disturbance scenarios. The relevant question in this context is rather how quickly the systems can adapt to the changed environmental conditions. This ability defines their robustness. The simulation experiments therefore follow a different approach.

The most important performance indicator is still the average service time. But we do not calculate this figure based on stationary simulation output data. Instead we cut the overall simulation into $L$ intervals of 30 minutes. For each of those intervals we calculate an average service time $\bar{Y}_i$ $(1 \leq i \leq L)$ based on the completed load transports. As we are interested in the course of the average service times, we use the replication/deletion approach (LAW 2006) to carry out $K$ simulation runs. An average service time $\bar{\bar{Y}}_i$ $(1 \leq i \leq L)$ for each interval $i$ across all simulation runs $K$ can now be

calculated. The 100(1-$\alpha$) confidence interval for the mean estimation $\bar{\bar{Y}}_i$ of interval $i$ is computed based on the variance $S_i^2$ across all $\bar{Y}_{ij}$ ($1 \leq j \leq K$) of that specific interval:

$$\bar{\bar{Y}}_i(K) \pm t_{K-1,1-\alpha/2}\sqrt{\frac{S_i^2(K)}{K}} \tag{7.1}$$

All other performance indicators are calculated according to the same logic. The relative error of the average service time estimations is used to define the number of required replications. Here it is not calculated with respect to the average service time of each interval but for the average across all intervals. The latter is the more decisive figure when deciding about the relative precision level and it prevents mathematical difficulties for intervals with low or no demand. The precision of the estimations is poor directly before idle periods of the system. During these intervals comparably high variances have to be expected and therefore the confidence intervals are very wide. To limit the simulation effort and as the intervals before idle periods do not have a major impact on the drawn conclusions we require at least 90% of the interval-based mean estimations to have a relative error of less than 15%. A confidence level of 95% is assumed. The numbers of required replications are summarized in the figure below. The achieved precision of the mean service time estimation is summarized in Appendix 2.

| | Path failure scenario | Extension scenario | Reduction scenario |
|---|---|---|---|
| **D3** | 200 | 400 | 200 |
| **D4** | 200 | 600 | 400 |
| **D5** | 200 | 400 | 200 |

Table 20: Required number of replications for disturbance scenarios

As stated above, we only consider layout scenario 2 in combination with the baggage handling data for the analysis of disturbances. We use 168 vehicles in the system as this amount led to stationary configurations in the experiments above. Using the same number of vehicles in each system configuration does not yield comparable average service times. During trial runs the impact of using different vehicle amounts which lead to similar average service times was tested (e.g. 160 vehicles for D3, 280 vehicles for D4, 208 vehicles for D5). The main findings from those simulation runs did not differ with respect to the used number of vehicles. However, using more vehicles increases the runtime of each simulation run. As we do not intend to make comparisons regarding the achieved service time but will only analyze their temporal development, we have decided to use the vehicle amount which minimizes the simulation effort.

The time of the disturbance is set to 367,200 seconds after the start of the simulation. This equals about 2.5 days of the real-world input data (load modification factor of 1.7 is used) and makes sure that the disturbance occurs when a reasonable amount of transport requests are in the system around noon of the third day. Setting $t_{dist}$ to 2.5 days should avoid any initial transient problems as only the data after the disturbance is analyzed. The length of the path failure is set to 30,600 seconds. Afterwards the path starts to work normally again and the system returns to standard working conditions. In a real-world system the downtime of a path can be expected to be less than 1,800 seconds in most cases. We had to choose a longer downtime in order to make the effects of this kind of disturbance visible. As mentioned above, the transport times in our system are rather short compared to the waiting times. But the disturbance in the first place influences the transport time only.

Therefore we had to choose the length long enough not to only influence the transport time but also the waiting time. Otherwise the impact is hardly visible when looking at the service time as the main performance criterion. The chosen 30,600 seconds equal about 5 hours of input data (5 hours x 3,600 seconds/hour x 1.7 as elongation factor).

After $t_{dist}$ 246 half-hour intervals of output data are recorded in total. This equals another three days of real-world data. For the path failure scenario the number of recorded intervals decreases to 229 because the downtime of 30,600 seconds needs to be subtracted. During the following analyses we will often interpret the output data after $t_{dist}$ with respect to the real-world input data sets. According to the chosen $t_{dist}$ the first day starts at noon. Day 2 and 3 both start at midnight and cover 24 hours. The fourth day also starts at midnight but only covers 12 hours. Due to the used input elongation factor, the following conversions apply:

| **Simulation intervals** | | **Real-world equivalent (for time after the disturbance)** |
|---|---|---|
| Path failure scenario | Extension / reduction scenario | |
| 1–24 | 1–41 | Day 1 / Day of disturbance |
| 25–106 | 42–123 | Day 2 / Day after disturbance |
| 107–188 | 124–205 | Day 3 / Second day after disturbance |
| 189–229 | 207–246 | Day 4 / Third day after disturbance |

Table 21: Conversion of simulation intervals to real-world data

The analyses in the next sections usually consider the whole range of 229 (path failure scenario) respectively 246 (extension and reduction scenarios) simulation intervals. Whenever it is judged to be relevant for understanding a certain system property, the time scale might be adjusted.

Similar to the path failure scenario, we found that the extension or reduction by only one source did not lead to clear visibility of the induced performance differences. Therefore sources 1 and 3 are added or taken off the system. The following comparisons distinguish the performance at those sources and the performance at all other sources.

As only a rather short time period is considered (5.5 days) for the simulation runs and the disturbances might lead to higher variability we do not consider the termination criterion for those experiments. We are aware of dealing with possibly non-stationary system characteristics. As the KPI are not calculated across the whole simulation run this property cannot be tested explicitly. But despite the disturbances, all analyzed systems still need to achieve the same throughput in order to enable reasonable comparisons. For the analysis we consider the length of the queues and require that the queue length returns to 0 at least once during the three day interval which is observed. It is obvious that this procedure cannot recognize a long term growth trend of the queue length. But it seems to be sufficient for the dynamic analysis. We will mention, whenever it becomes evident that the required throughput is not achieved.

After the definition of the simulation parameterization, the definition of meaningful reference scenarios for the performance assessment is the next step. The figure below summarizes which system setups are compared.

For the path failure scenario the network has a modified structure from $t_{dist}$ to $t_{norm}$ as one of the paths is not accessible. Before and after the breakdown the network structure is the same. Our comparison focuses on the time after the disturbance and checks how quickly the system recovers

from the disturbance. The standard system configuration without the disturbance is used as reference. This means we compare the average service time values after the disturbance to those values which could have been achieved when the disturbance would not have occurred.

For the other two disturbance scenarios finding an appropriate reference is more difficult. The focus is again the recovery time after the disturbance. For a proper comparison we need to create a reference scenario which is similar to the system which is at hand after the disturbance. We therefore use a standard system which starts its operation initially with all sources / storage locations as a reference for the extension. Its average service times which were achieved after the time of the disturbance are compared to the expanded network. A similar approach is chosen for the reduction scenario. Here we are interested in comparing the performance of a system with only six sources and storage locations after the disturbance. The reference scenario is therefore a system which operates constantly with only six sources and storage locations.

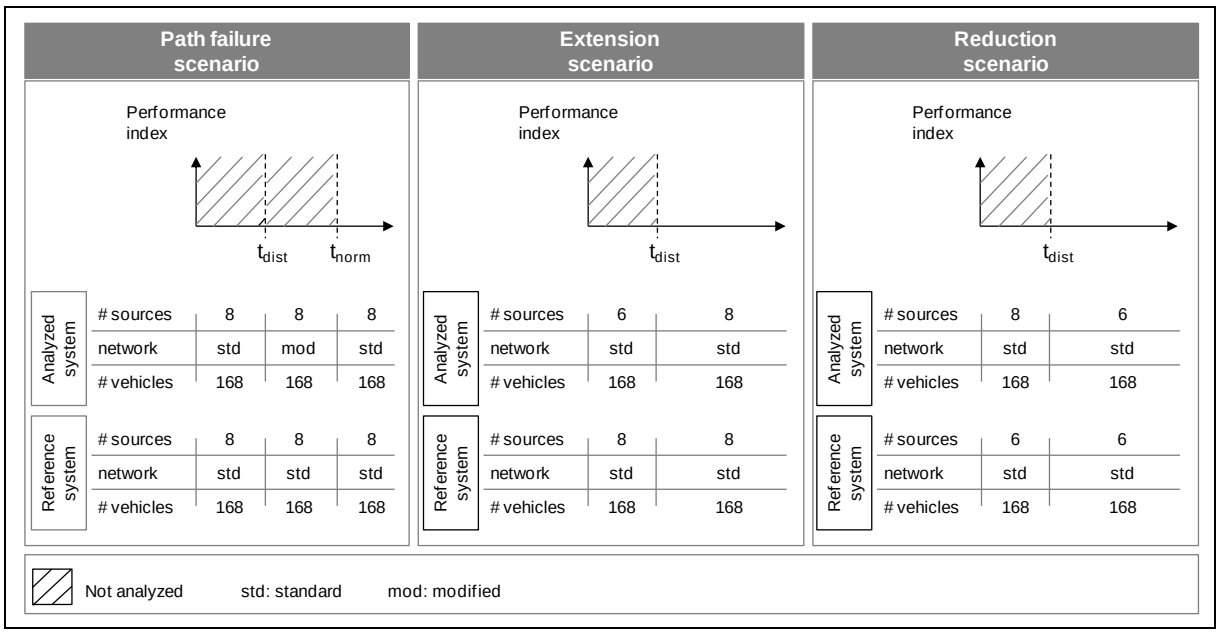



Figure 75: Framework for the evaluation of the disturbance scenarios

Following this approach, the number of vehicles in the system can be kept constant in all scenarios. Although we neglect the performance level which the different systems achieve before the disturbance, it needs to be noted that the system operation before $t_{dist}$ may well affect the performance of the system after the disturbance. As those effects cannot be measured, prevented or excluded, they have to be kept in mind when interpreting the analyses.

After these introductory remarks, the following sections show the results of the simulation runs.

**Results path failure scenario**

Looking at the results it is in a first step important to notice that all dispatching strategies are able to deal with the three disturbance scenarios. Although there might be performance losses, the systems remain operational.

First we analyze the impact of a path failure on the performance of the central dispatching strategy. The figure below indicates that D5 is not very sensitive to the disturbance.

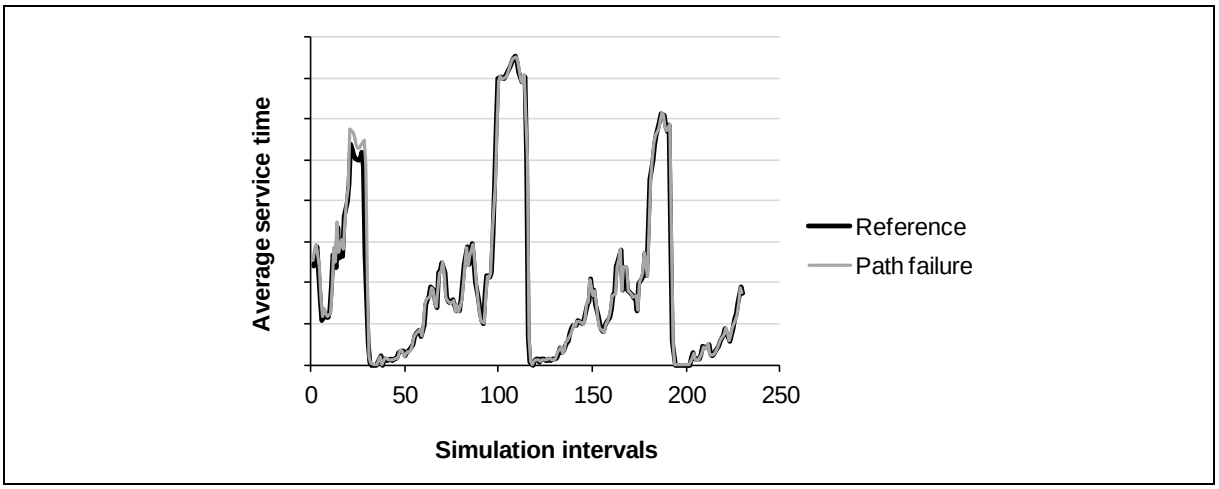



Figure 76: Average service time in path failure scenario for D5

It is the strategy which drove the duration of the chosen downtime. In order to show performance differences at all, the period of the path failure had to be increased to 30,600 seconds. But even with this fairly long downtime, the service time differences between the reference and the failure scenario remain comparably small. Only between the simulation intervals 20 and 30 does a substantial difference become visible. Those intervals equal the end of the day of the disturbance. During this time slightly longer queues which were built up during the disturbance due to lower throughput have to be served. Prolonged service times are the result. The disturbed system therefore also works slightly longer than the reference system until all remaining transport requests of day 1 have been processed and a completely idle state is reached. After the first idle system period, the performance differences disappear for the following days. During the idle period the dispatching algorithm is able to redistribute the vehicles equally in the system. Therefore the system starts operating with the first demands of the next day as if the disturbance would never have occurred.

For D4 the disturbance also leads to an increase of the average service times. As soon as the throughput requirement of the input data declines at the end of day 1 and the system gets closer to an idle state it is able to recover. The performance returns to the level of the reference system. However, small deviations can be observed during the next day. Those can be traced back to differing distributions of the vehicles to the dispatching locations. By chance, the distribution which was affected by the disturbance might be closer or less close to the distribution of the reference system. Over time, these differences in distribution are leveled and do not affect the performance any longer. Small remaining differences can be neglected. However, it needs to be mentioned that the disturbance has a long-term impact on the throughput and the queue length of the system. Although this is not visible in the figure below, the system does not reach a completely idle state at the end of days 2 and 3. Not all loads which are waiting at source 3 have been processed. But the queues are small and only contain a few loads. No general growth trend can be derived.

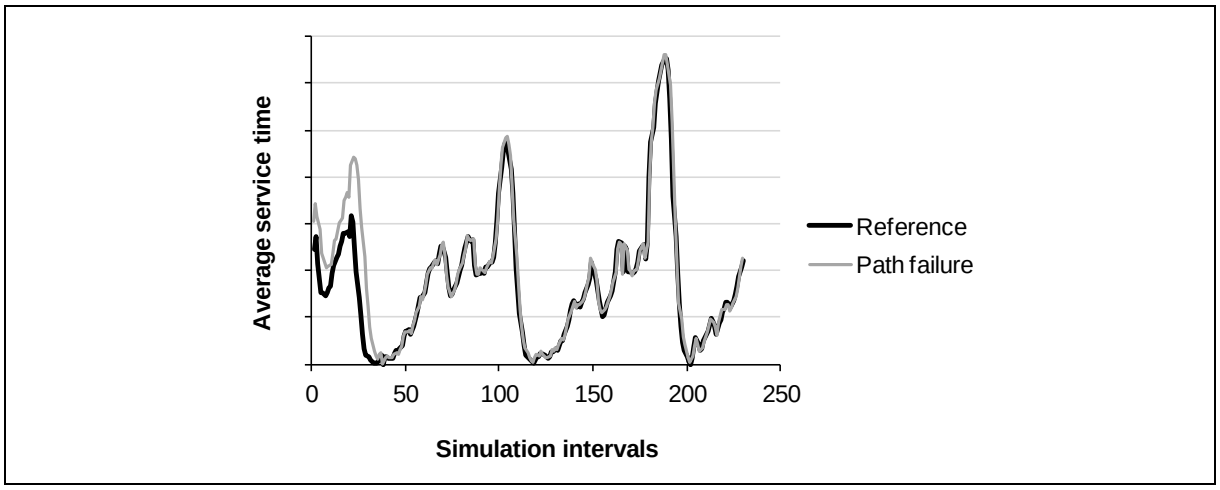



Figure 77: Average service time in path failure scenario for D4

When forecast dispatching is applied, the consequences of the disturbance are much more severe than with D4 and D5. Directly after the disturbance occurs, a steep increase of the average service times can be observed. The reason is a drop of the throughput which occurs while the path is not available. This drop is stronger than for D4 and D5. It can be explained by a delayed feedback process. During the disturbance the dispatching of empty vehicles and arrival of laden vehicles is delayed for all source-sink and sink-source combinations which include stations from the upper and lower left part of the system. The delay of arrivals does in return also postpone the deliveries and the feedback which the sinks receive. The effect reinforces itself and has a big impact on the achieved performance. It is supplemented by the fact that the sources in the left part of the system are responsible for the majority of the overall transport requests. Loads are accumulating at the sources while the throughput is decreased. The low additional demand at the end of the disturbance day allows the system to process those waiting loads and to recover from the disturbance. However, the system is not completely idle between day 1 and 2.

After the initial performance loss has been compensated, another interesting effect can be observed. While the average service times of the compared systems are in a similar range at the beginning of the second day, the disturbed system outperforms the reference system at least during the second half of that day (starting around simulation interval 70). In addition, even the third day performance is slightly affected.

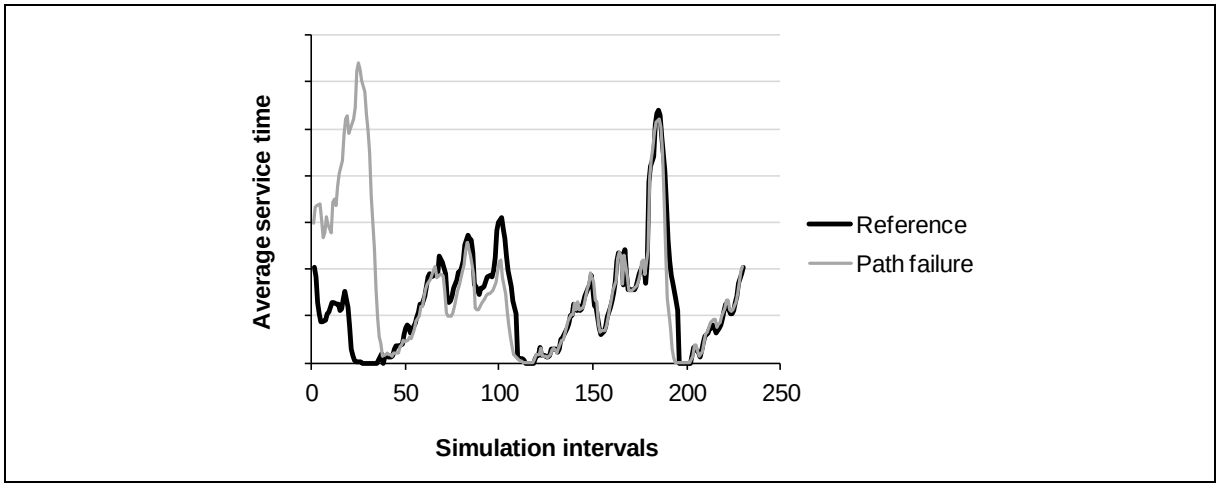



Figure 78: Average service time in path failure scenario for D3

To understand this effect an in-depth analysis of queues and empty vehicle distribution is required. The analysis illustrates the consequences of the throughput decrease during the disturbance which was mentioned above. Source 3 is chosen to explain the behavior of the system. Similar effects, but on a smaller scale and therefore less visible, can be observed for sources 1 and 2.

Due to the mentioned throughput drop between $t_{dist}$ and $t_{norm}$ an extreme increase of the queue length at source 3 can be observed compared to the reference system (see figure below).

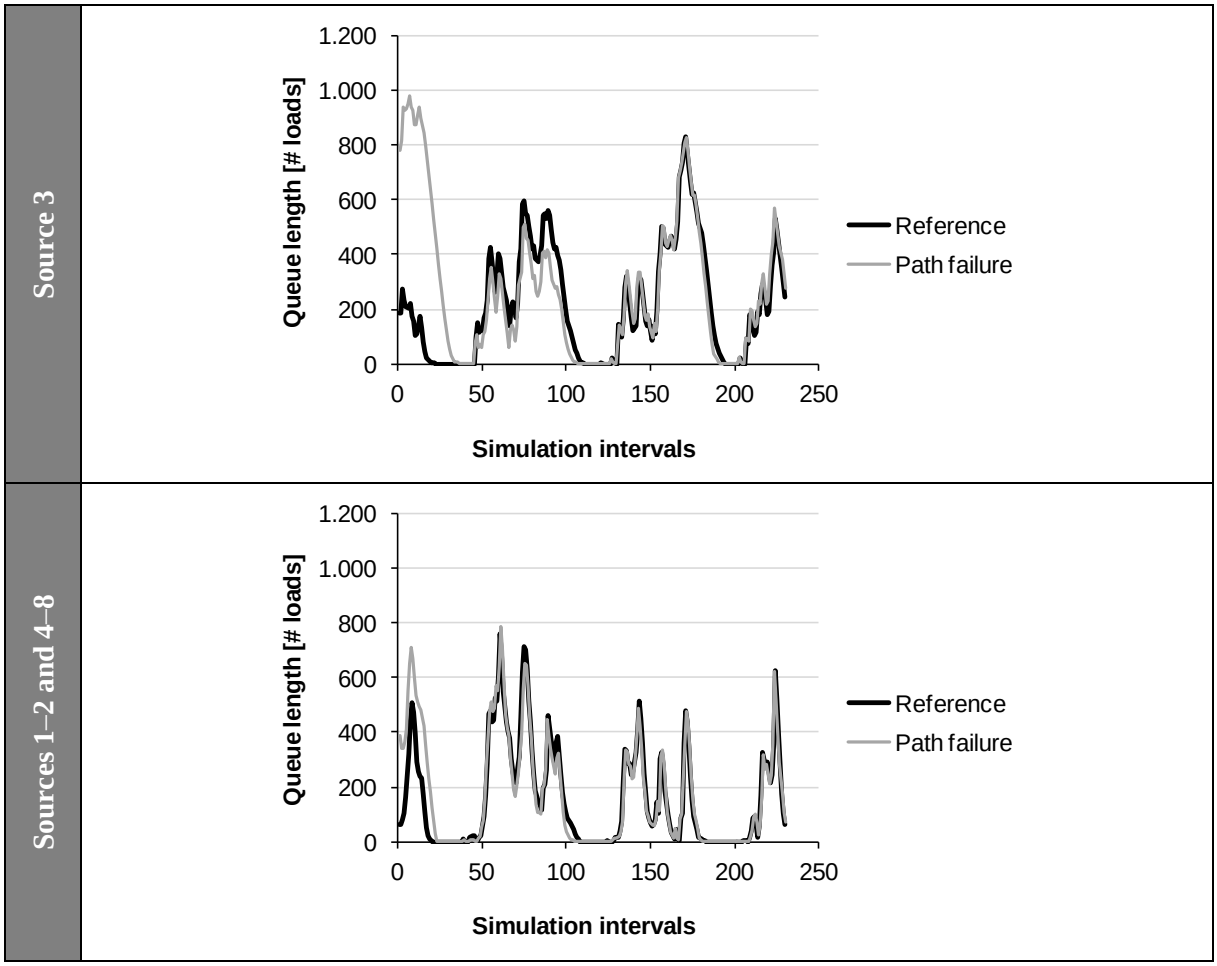



Figure 79: Queue length for D3 in path failure scenario

The pickup and transport of those queuing loads after $t_{norm}$ has the consequence that the demand forecast values for source 3 increase and many vehicles are sent there. Once all the remaining demand has been fulfilled, the vehicles are consequently sent to the storage location which corresponds to source 3. This is the storage location where the latest fill level information is likely to have been received from. Therefore an extreme peak of vehicles at storage location 3 is the consequence during the following low demand period (see intervals 40 to 60 in Figure 80). This peak is afterwards used to serve the demand at source 3. As the throughput at source 3 is the highest in the system, short waiting times here influence the overall average service times in the system. The disturbed system performs better than the reference system. The same effect can to a certain extent be observed for day 3. But the oversteering effects which are induced by the queuing loads during the disturbance disappear over time.

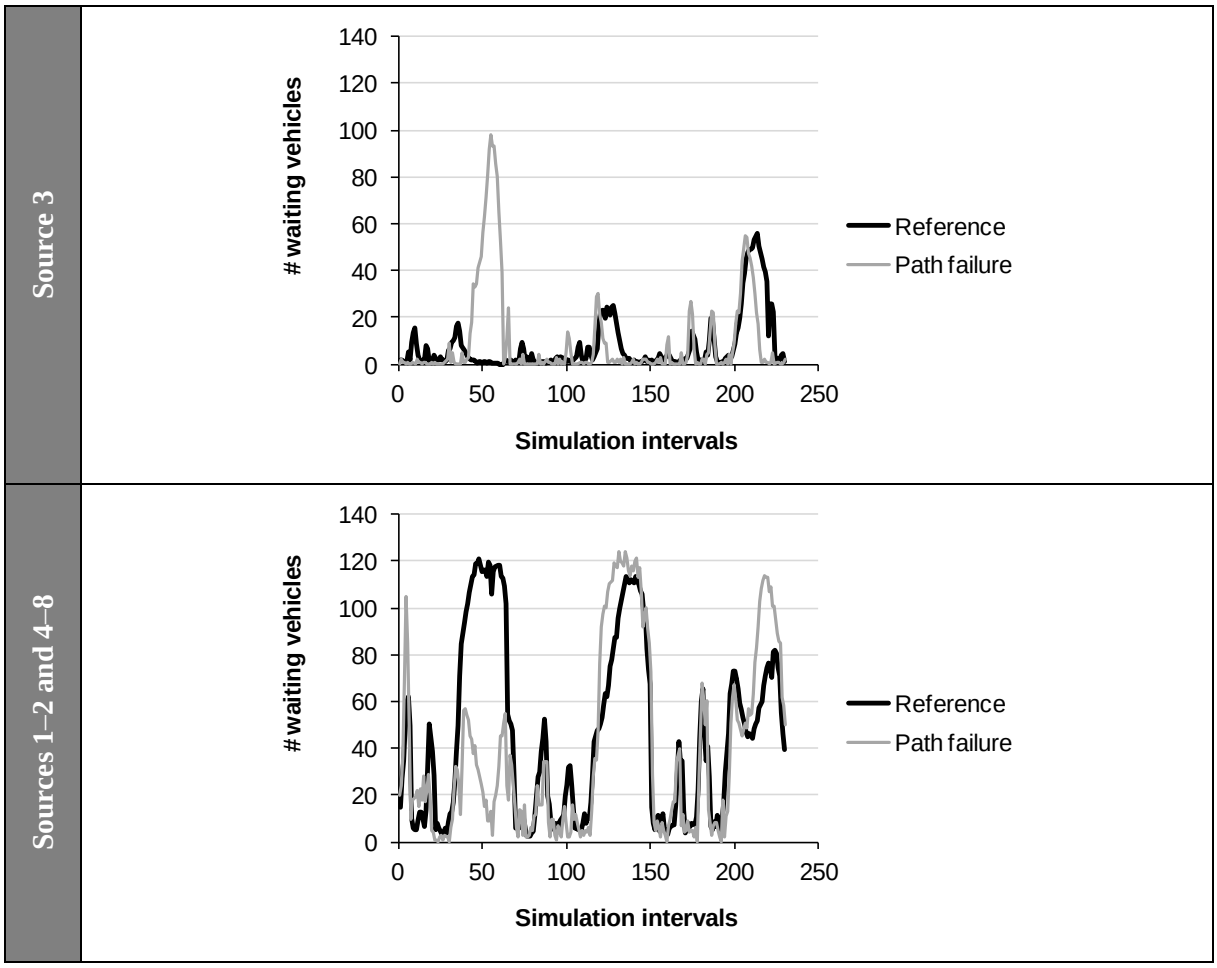



Figure 80: Number of waiting vehicles for D3 in path failure scenario

Summarizing, D5 is not very sensitive to a path failure. The disturbance does not have a big impact on the performance. Once a redistribution of the vehicles was completed during the next idle period the disturbed system performs as well as the reference system. D4 is less robust. The performance loss is initially bigger than for D5. Although major performance differences can only be reported for day 1, a slight impact remains visible during the following days. For D3 the disturbance has the biggest impact. It does on the one hand result in a quick and strong performance loss directly after the disturbance. On the other hand, an improved overall performance during the next two days is the consequence. The oversteering effects seem to disappear over time.

**Results extension scenario**

After the path failure scenario we will now take a look at an extension of the network. The following analyses distinguish the service times for loads which are picked up at sources 1 and 3 and the service times for loads which are picked up at all other sources in the system. For the extension scenario it is important to note the side effects which our analysis approach has. The reference scenario assumes that all sources have a demand during the whole simulation time. Depending on the prior performance there might be queues at the sources 1 and 3 at $t_{dist}$. These queues influence the system behavior during the following simulation intervals. The disturbance scenario on the other hand assumes that source 1 and 3 are added to the system at $t_{dist}$, i.e. there is no prior demand and there are no queues.

Those differences should be kept in mind for the following comparisons. They become visible when looking at the results for D5. The subsequent figures show that the performance curves of the disturbed system approach the reference performance curves from below. In the reference system there

are many loads waiting at source 1 and 3 at time $t_{dist}$. Those have to be processed in the following simulation intervals. There are no initial queues at this source in the disturbed system setup. Therefore the reference performance remains above the performance level of the disturbed system until an idle period is reached and all the waiting loads at source 1 and 3 have been processed. Afterwards there is no difference between the performance of the reference and the disturbed system.

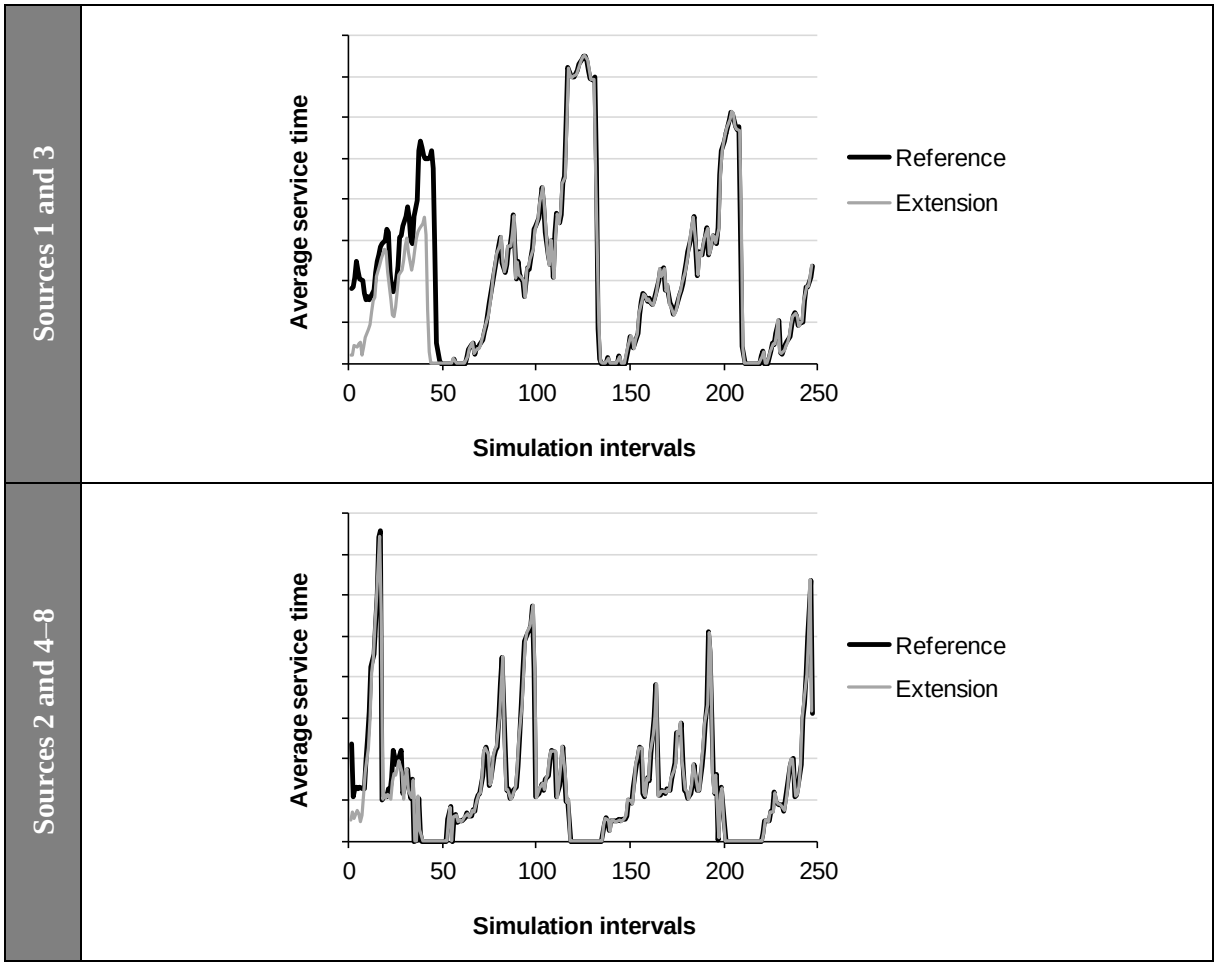



Figure 81: Average service time in extension scenario for D5

But these implications are not limited to source 1 and 3. They can also be found for the rest of the sources. The overall throughput requirement of the reference system was higher than for the disturbed system before $t_{dist}$. Therefore the overall queues at the sources 2 and 4 to 8 tend to be longer in the reference scenario. As soon as the sources 1 and 3 are added to the system, the overall throughput requirement increases in the extended network and the performance of the disturbed system approaches the performance level of the reference system. After about 7 simulation intervals, the performance differences start to become negligible.

The disturbed system uses parked vehicles to transport the additional demand which arises when source 1 and 3 are added to the system. This process is illustrated in the figure below. Parked vehicles are activated and used to transport loads. The disturbed system has a performance reserve while there are no idle vehicles in the reference system at $t_{dist}$.

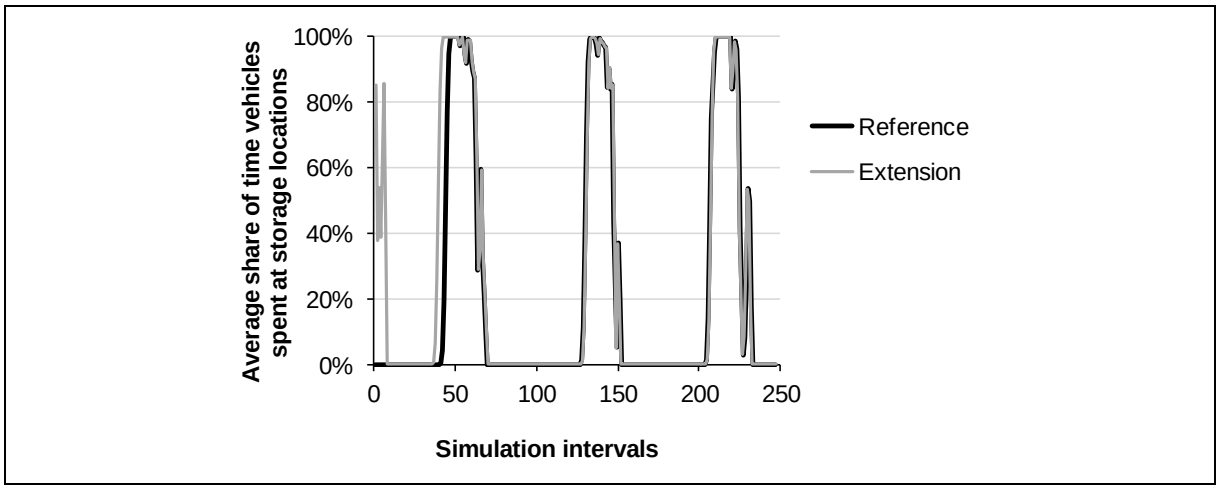



Figure 82: Share of time vehicles spend at storage locations in extension scenario for D5

The behavior of D4 is different to that of D5 in the extension scenario. In a first step we take a look at the impact that the propagation of the network knowledge has on the dispatching performance. The figure below clearly indicates that the throughput at source 1 and 3 grows initially quicker when the information about the new network structure is spread.

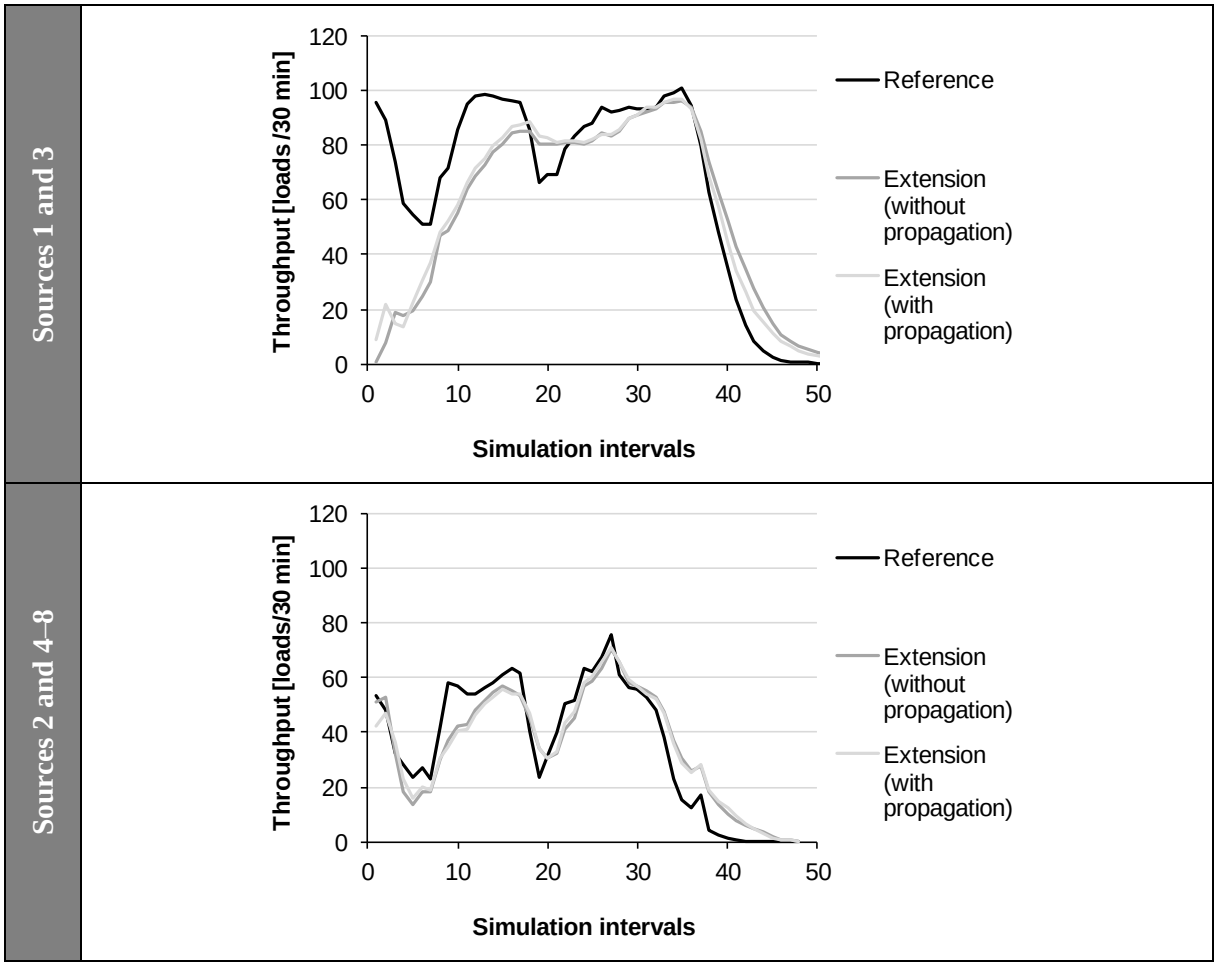



Figure 83: Achieved throughput in extension scenario for D4

The new source does not have to be discovered by chance, but vehicles are sent there on purpose. However, the propagation of the information only influences the throughput during the first two simulation intervals after the extension. Afterwards no major differences can be observed at source 1 and 3. Sending more vehicles to sources 1 and 3 changes the vehicle distribution in the system and leads to lower throughput at the other sources in the system. The system requires some time until it has adjusted its performance to the new demand patterns.

This can also be observed when looking at the average service times (see figure below). The better initial throughput for the strategy with information propagation is reflected in the lower average service times at sources 1 and 3. But in contrast to the findings for D5 the performance of the disturbed system does not only approach the reference performance from below (for both strategy versions), it also exceeds it for the rest of the first day. Not all loads at sources 1 and 3 are processed before the start of the next day.

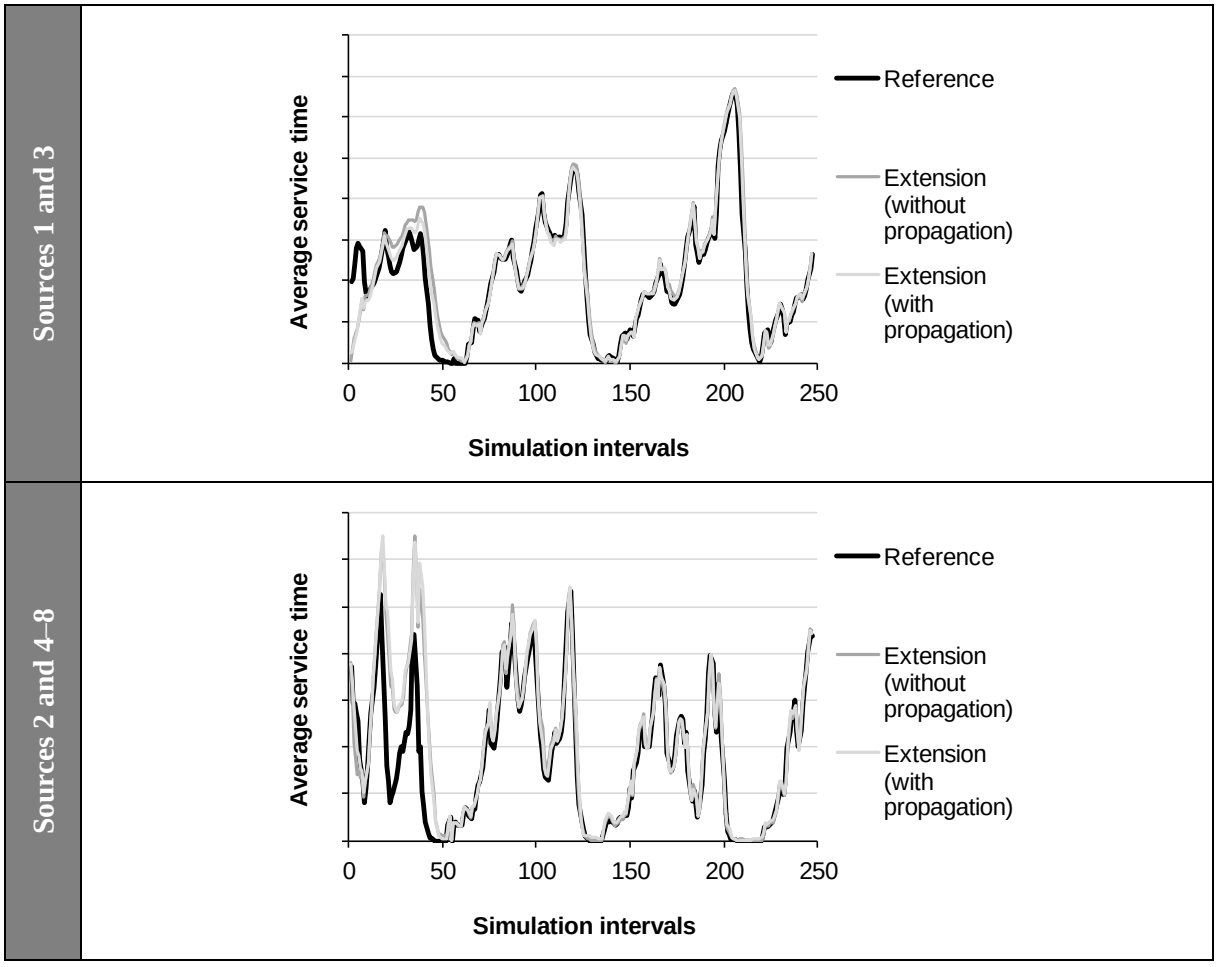



Figure 84: Average service time in extension scenario for D4

Although there are far more loads waiting in the reference system at $t_{dist}$, the performance level of the disturbed system grows above the level of the reference system. Besides the initial throughput differences which might lead to longer waiting times for some of the loads, this exceedance can be explained by the time that D4 needs to activate parked vehicles (see Figure 85 below). For D5 the share of parked vehicles drops to the level of the reference system within about 10 simulation intervals. D4 needs a phase with low demand to get enough vehicles into the load pick-up process. For the remaining time of the first day too few vehicles are available to process all transport requests as fast as in the reference system. D4 is inert and needs until the second day to adjust to the new

situation. In contrast to D5 the feedback-based dispatching strategy is not able to redistribute vehicles within the whole system. Sources only have access to their corresponding storage locations. During the days after the disturbance slight influences remain visible but they become negligible over time. The propagation of network information can slightly improve the performance of D4.

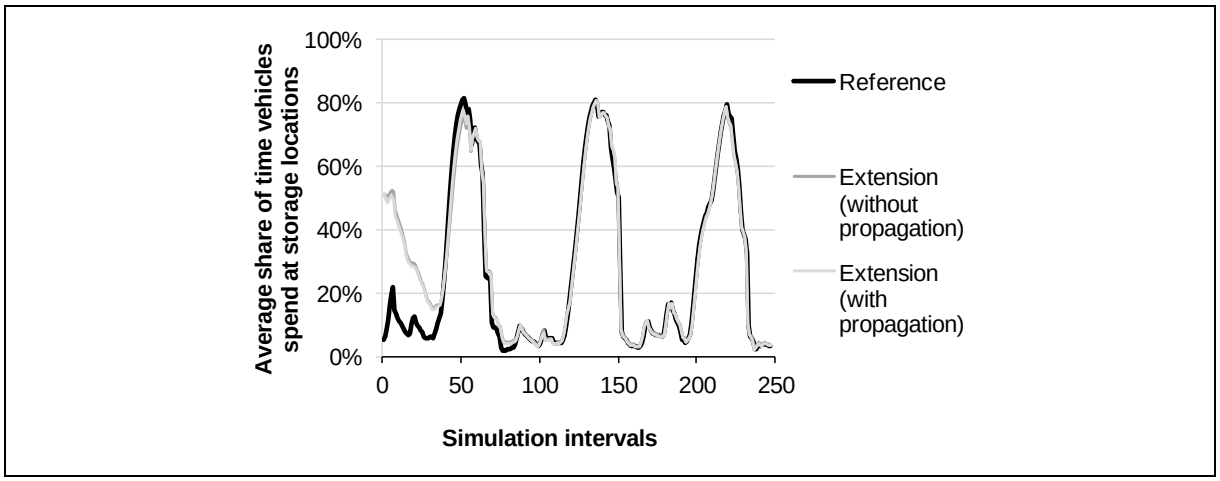



Figure 85: Share of time vehicles spend at storage locations in extension scenario for D4

Looking at the initial throughput which is achieved at source 1 and 3 leads to other conclusions for D3. When forecast dispatching is applied, there is no throughput difference for the system with knowledge propagation and the system without information spread.

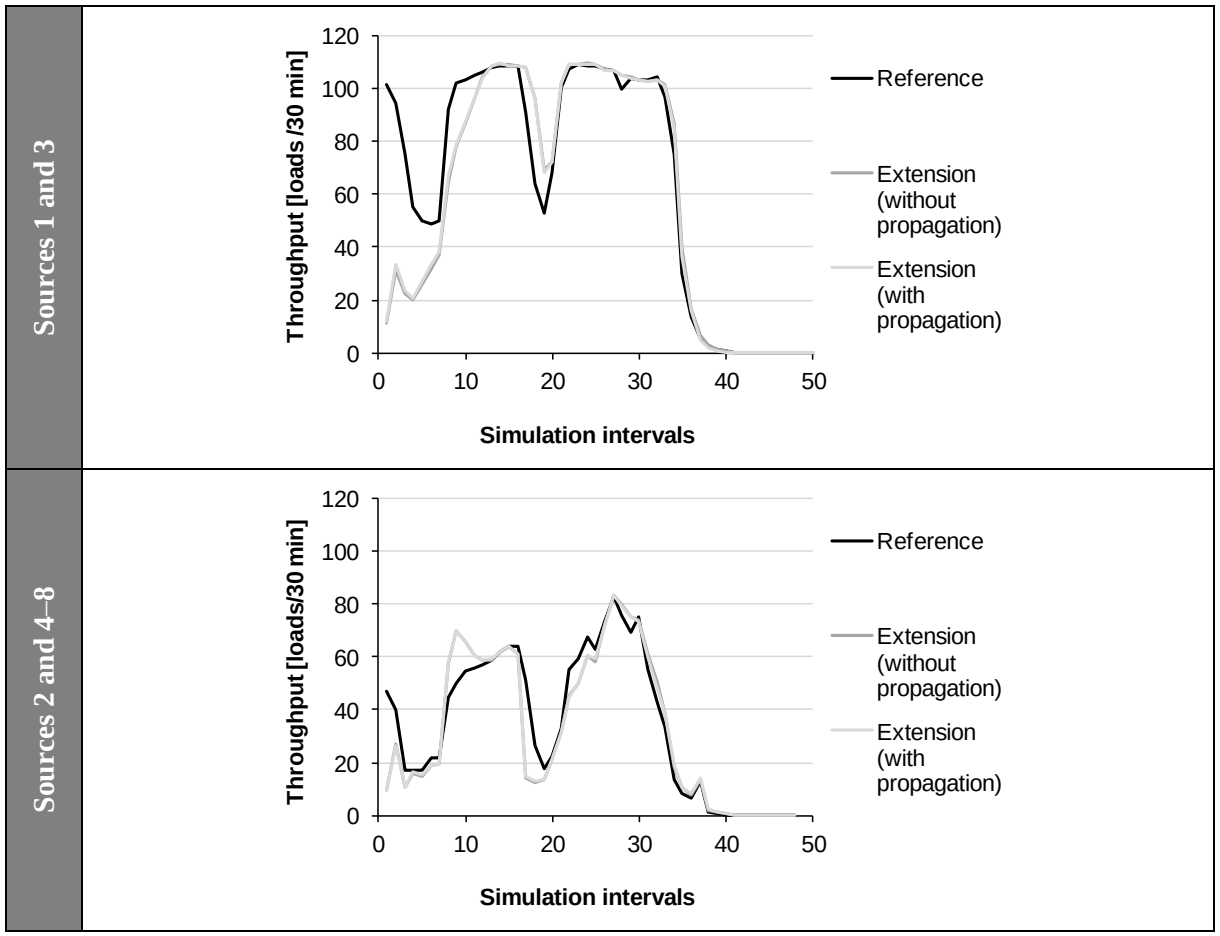



Figure 86: Achieved throughput in extension scenario for D3

The new network information is given to the sinks. These use the updated knowledge for the dispatching of empty vehicles. This means in a first step the new sources have to be selected as dispatching locations during the dispatching process. Afterwards the dispatched vehicles have to arrive at the source to pick up loads. This process simply takes a long time. In the meantime randomly travelling vehicles have already discovered the new sources and storage locations. Therefore, no visible difference between both control approaches exists. The effect is supplemented by the initialization values for the demand forecast and the fill level estimation at the new sources.

As a result, the course of the performance curves for both control strategy versions are almost identical. Slight differences are only visible during absolute peak times of day 1 and 2. Those remaining differences can be traced back to different vehicle distributions due to different storage location fill level estimations.

At all sources the disturbed system reaches the same performance as the reference system after about 8 simulation intervals. Similar to D4, the performance curve of the disturbed system is above the level of the reference system during day 1. But in contrast to D4 those effects are much less severe and mostly limited to sources 1 and 3. It seems that the performance at sources 1 and 3 is slightly worse than the reference system while the performance at the other sources is slightly better.

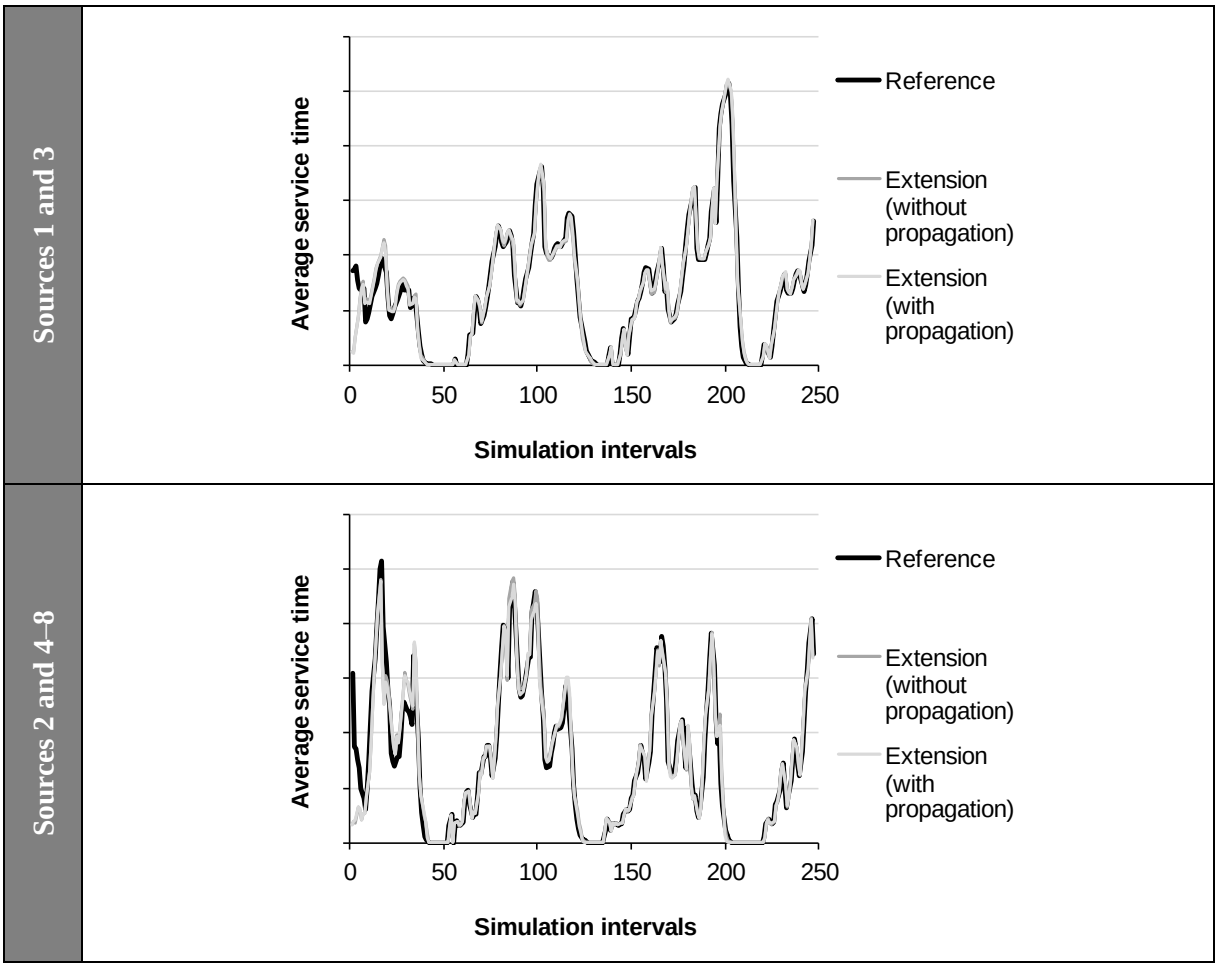



Figure 87: Average service time in extension scenario for D3

Concluding, D5 seems to handle the disturbance best. It keeps the service times at the new sources below the level of the reference system until the first idle period. The performance of the reference system is never exceeded. D3 and D4 both show a more inert behavior. They need more time to adjust

their performance to the new demand requirements. Of the two strategies, D3 adapts quicker to the changes and only slightly exceeds the performance curve of the reference system. D4 needs a period of low demand to catch up completely. In the long term no significant performance differences can be reported for any of the strategies. While the propagation of network information does not have a major impact for D3, it can help to improve the performance of D4. For all comparisons the limitations which were stated at the beginning of this section have to be kept in mind.

**Results reduction scenario**

The final analysis considers the situation when source 1 and 3 are taken off the system. In this scenario performance changes occur rather quickly and are only visible for a short time. Therefore the following figures always show two time scales: One for the first 50 and one for all 246 simulation intervals.

D5 adjusts extremely fast to the lower demand which results from two sources less in the system. Within the third simulation interval after $t_{dist}$ the performance of the disturbed system already drops to the service level of the reference system. As explained for the extension scenario above, these initial differences are caused by the prior operation of the system. The reference system never experienced a demand at source 1 and 3. But in the disturbed system loads had to be picked up at these sources until the disturbance occurs. After the third simulation interval only slight performance differences during peak times are observable. These can be explained by deviating vehicle distributions.

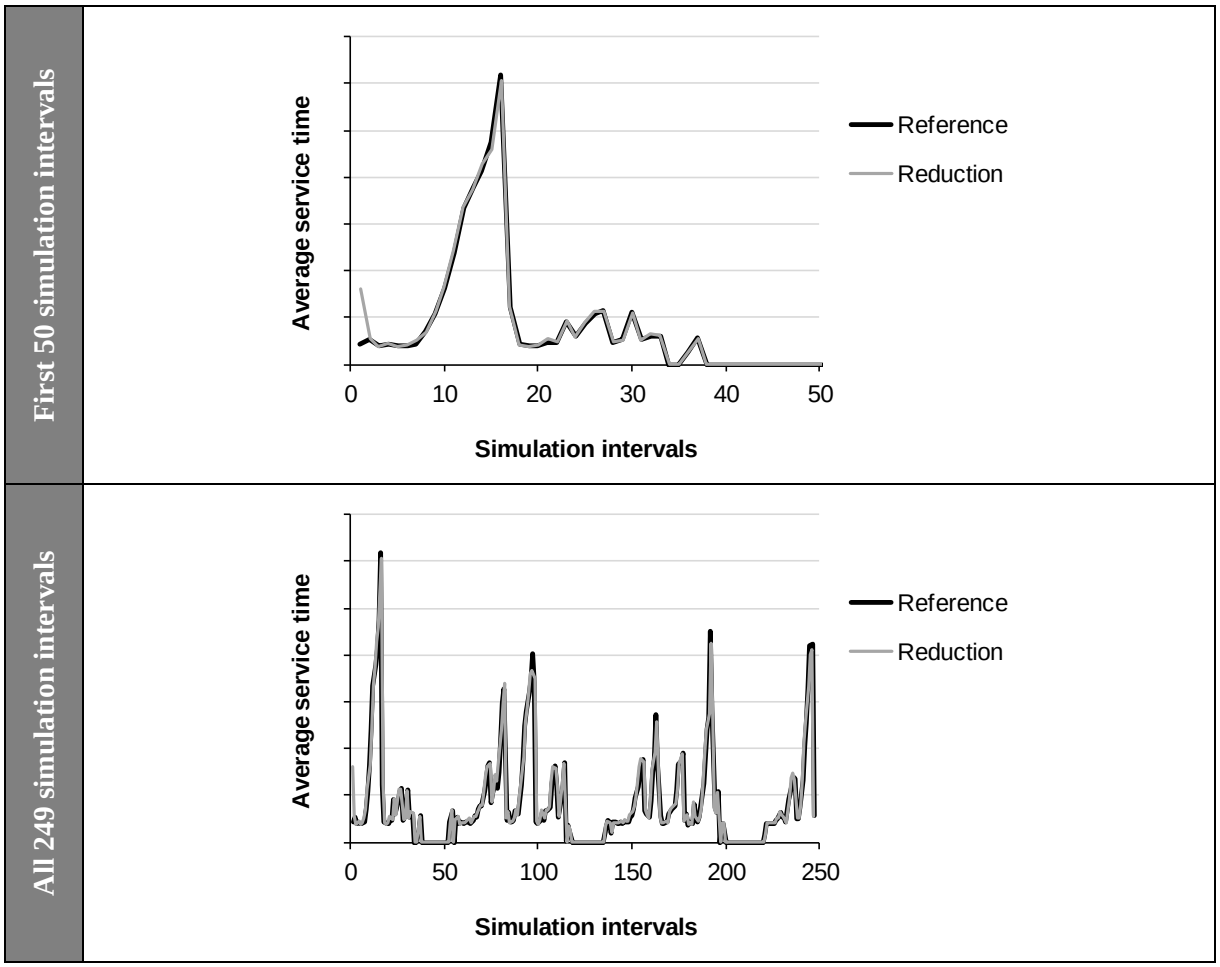



Figure 88: Average service time in reduction scenario for D5

D4 behaves differently. Here, it takes slightly longer until the performance drops to the level of the reference system. After 4–5 simulation intervals the transition is completed. Initially, the difference between the version with and without information propagation is extremely small. But over time it becomes visible. The algorithm without knowledge propagation performs far worse than the version with knowledge propagation. The explanation can be found in the dispatching tables and has already been mentioned. If the knowledge of the changed network layout is not spread among the network nodes, the extracted sources and storage locations remain part of the dispatching tables. The probability of sending vehicles to those locations will therefore never be 0. As the destinations are no longer part of the system, vehicles which have been dispatched towards them cannot be routed there. They have to continue travelling randomly (see Chapter 5.3.1) and contribute to the overall inefficiency. However, the required throughput is reached. The dispatching algorithm which applies information propagation more or less achieves the performance level of the reference system throughout the simulation run. Only smaller differences can be recognized and are negligible.

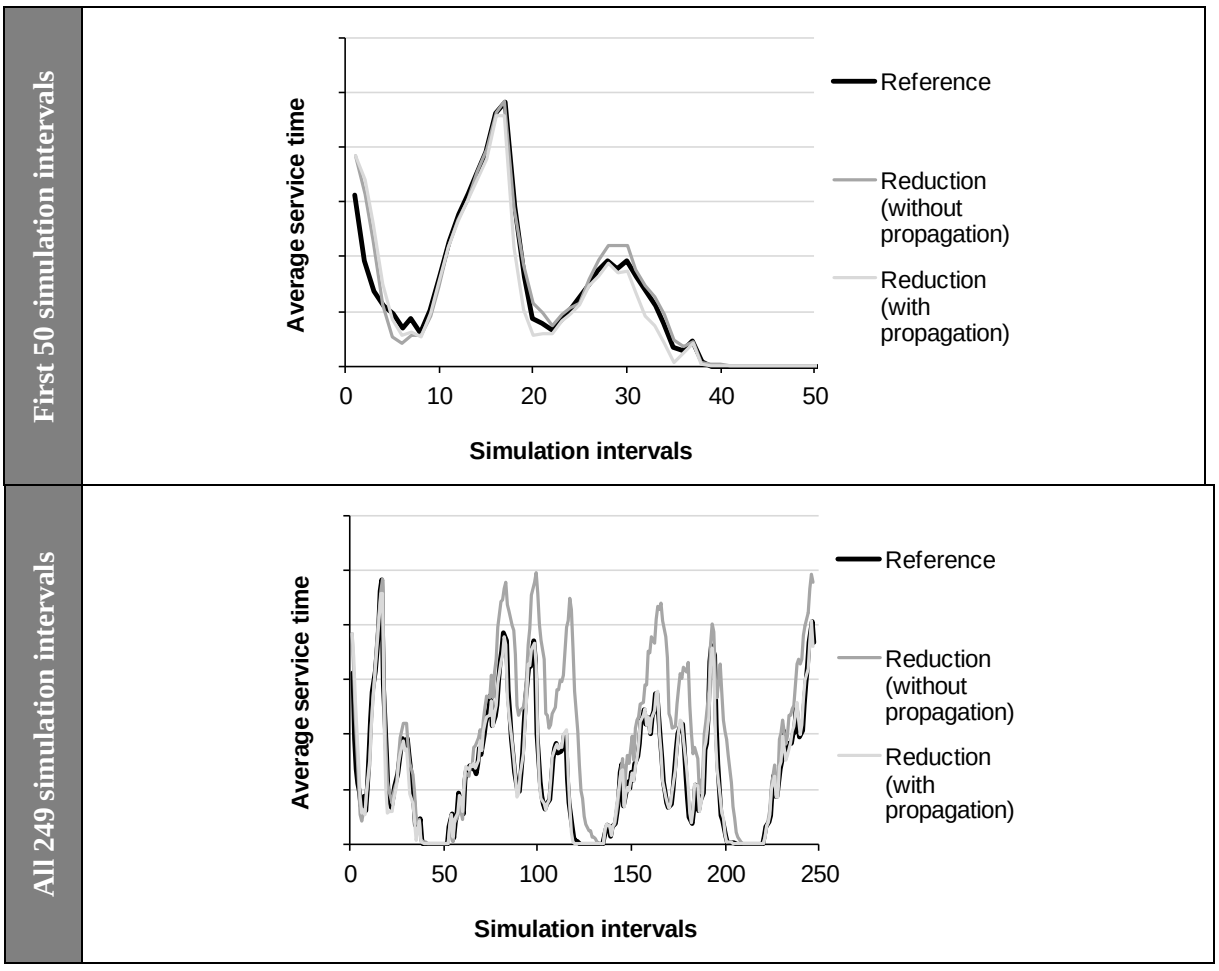



Figure 89: Average service time in reduction scenario for D4

Similar to D4, D3 needs about 4–5 simulation intervals until its performance drops to the level of the reference system. Although it initially experiences shorter queues from the preceding intervals it needs the same time for the adjustment. After the transition period hardly any differences between the three performance curves of reference system, disturbed system both with and without information propagation can be observed.

In addition to the explanation above, the uniformity of the latter two curves can be explained in detail by the following two facts:

- The estimated vehicle demands at all sinks for sources 1 and 3 are quite small when the disturbance occurs. Due to the weighting factor of 0.5 for the estimation formula, the forecast quickly (< 5 forecast intervals) drops to 0 when no more vehicles from source 1 and 3 are received. This does not differ much from setting the demand directly to 0 when the knowledge spread is applied.
- The fill level estimations for storage locations 1 and 3 are at a comparably high level when the disturbance occurs. Therefore only a few vehicles are sent to these storage locations. As no further vehicles are received at the sinks, the fill level quickly grows to the upper boundary of 999,999. This has the same effect as removing the storage locations from the list of known dispatching locations of the sinks. Only if the fill level estimations for all storage locations grew to 999,999 at one of the sinks, would one of the storage locations be chosen randomly for dispatching and it could happen that a vehicle is sent to storage location 1 or 3 again.

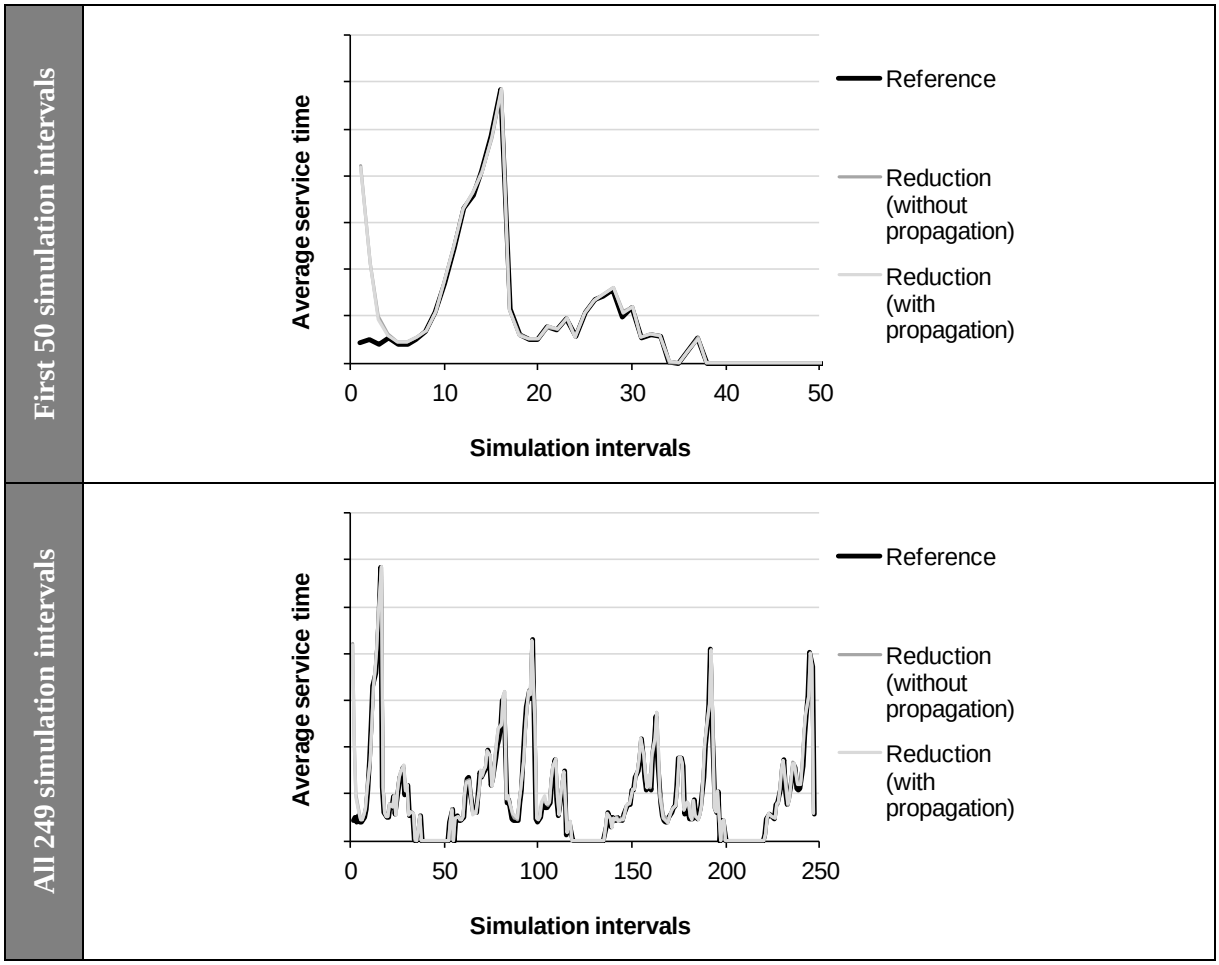



Figure 90: Average service time in reduction scenario for D3

Comparing the three control strategies, D5 reaches the reference performance level faster than D3 and D4. But similar to the extension scenario this also depends on the system performance before $t_{dist}$. While it is important for D4 to implement a propagation of the network information to ensure a stable behavior of the system, this is not required for D3. When forecast dispatching is applied, the system can adapt to the new topology automatically without a big loss of performance.

Summarizing the results from the extension scenarios, the following conclusions can be derived regarding the robustness and adaptivity of the analyzed control strategies. For all statements it has to be considered that the results do not only depend on the adaptivity of the algorithms but also on their performance prior to $t_{dist}$:

- The central and the decentral strategies are able to deal with all disturbance types. However, it needs to be kept in mind that we assume for the purpose of these analyses that the central control system remains operational after the occurrence of the disturbance. This will not be the case in most real-world applications. As already mentioned, a central control system only has limited capabilities for dealing with disturbances. In contrast, the decentral systems are able to remain operational.
- A path failure has only a very small impact on the system performance for D5. After an idle period the system returns to the reference performance. For D4 and D3 the disturbance has long-term implications. Although D4 returns to the reference performance after the first idle period, the disturbance still influences the long-term performance on day 2 and 3. For D3 the impact is even more severe. After an initial performance loss directly after the disturbance, the system shows an oversteering behavior during the following few days which leads to improved service times.
- D5 seems to show the best adaptivity regarding the network extension. The performance of D3 and D4 drops quicker to the level of the reference system. D4 is especially more inert and leads to a temporary performance loss compared to the reference system. The propagation of the updated network topology is not required for D3, but beneficial for D4.
- In the case of a breakdown, D5 adapts fastest to the changed requirements. D3 and D4 need about twice the time. D4 is only able to achieve the reference performance afterwards when the information about the new network topology is propagated in the network. For D3 this is not an essential requirement.

The analysis of disturbances was the last performance assessment in this thesis and offered insights regarding the robustness of the control strategies. Based on all the results which were presented in this chapter the final conclusions for the comparison of central and decentral control can now be derived.

# 8 Limitations

Before we draw our conclusions in the next chapter, it is necessary to briefly review the methodology and the assumptions of this thesis. Both constitute the limitations which have to be kept in mind when discussing the findings and judging about the applicability of decentral dispatching rules in real-world scenarios.

It was argued in the introductory chapter that simulation is the only applicable approach for generating the results of this study. Therefore the findings are not derived analytically. They are not universal statements that can be easily transferred to other applications but are limited to the analyzed simulation model. To decide whether they allow more comprehensive inferences about the nature of decentral control the simulation input parameters need to be assessed.

Firstly, we tried to develop a generic test layout. It is obvious that this layout can hardly represent all layout types which can be found in real-world applications. Transport systems are usually tailored for their specific purpose. For example picking, line feeding and baggage handling systems all require completely different layout shapes. They are customized for the facilities they are used in. However, the test layout contains all station types which can typically be found in transport systems. In addition, loop-based path layouts are very common. Therefore the test layout allows insights into the characteristics of transport systems. The generic structure enables the comparability of different control systems. But it is impossible to derive general statements from the test systems. Before applying it in reality, a new dispatching strategy would always need to be tested with its specific real-world layout.

In the context of the layout, the storage locations are an additional factor which needs to be considered. They are treated in an extremely simplified way. In reality the layout of a storage location would in most cases look different. Instead of one long loop, several parallel lanes would be used which are controlled with sophisticated rules. This leads to shorter travel distances and better access to specific vehicles. But as we only analyze the distance travelled outside the storage locations and do not need access to specific vehicles this deviation from reality does not limit the quality of the results.

The assumption of sufficient storage location capacity has more far-reaching implications. Due to economical reasons the storage locations in a real-world system would always be kept as small as possible. In many cases only a small share of the vehicles could be parked at each storage location. The variable usage and partial excess of the storage location capacity by the different control strategies make the consequences of this assumption a topic for future research.

Besides the layout, the used input data is the next factor which needs to be evaluated. Our experiments have shown that the load data has a huge impact on the achieved performance and on the comparison of central and decentral strategies. Two different data sets are used in this thesis. But based on the findings, it must be expected that other data sets will lead to different results. This topic therefore needs further examination and limits the generalizability. All comparisons are carried out at throughput levels that lead to operational system configurations for all control strategies. The same throughput is used for each combination of layout scenario and input data set. This is the result of our analysis methodology. It needs to be understood that this does not necessarily mean that all control strategies achieve their maximum throughput performance. Some strategies might be able to achieve higher throughputs. The comparison at similar throughput levels simplifies the analysis by excluding one of the performance indicators. In addition, we analyze the systems at comparably high throughput levels. Therefore the waiting times are very long compared to the transport times. This is especially true for configurations with only a few vehicles. In many real-world applications other ratios of waiting and transport time would be present.

A last topic which should be noted with respect to the input data is the equal distribution of the transport requests to the sinks in the system. This assumption is necessary because no real-world information was available. In the experiments it contributes to spreading the overall system load equally and supports the propagation of information for the decentral strategies. It can be expected that this approach enhances the performance of decentral strategies and might not be present in real-world situations.

The mentioned shortcomings can to a certain extent be attributed to the usage of real-world input data. Despite its weaknesses, this approach highly improves the result quality by using realistic load profiles instead of artificial input data.

Looking at the tested control strategies, it will always be possible to develop “better” control rules. Sometimes this might be possible by slightly changing one of the input parameters. Even small changes can have a major impact on the system performance. This thesis is among the first approaches to develop decentral dispatching algorithms. There was almost no theoretical framework for their development. After this pioneering work, faster advances can be expected to follow in the future. It is obvious also that the benchmark strategy can be improved and be bypassed by more sophisticated systems (e.g. multi-attribute rules). Further developments, e.g. to not distribute the vehicles equally to all storage locations, but to adjust the buffer capacity and vehicle assignment to the actual load patterns, seem feasible. Also a learning process is imaginable. Changing either the central or the decentral control strategies will always lead to altered results. Generally, the assessment of control strategies always remains fuzzy unless an optimization approach is used. But the latter is not applicable in this context.

This thesis only focuses on decentral dispatching. It is assumed that a routing algorithm is available and provides the required routing information. The impact that routing and other central or decentral control strategy types have on the system efficiency is thereby knowingly ignored to increase the expressiveness of the conclusions regarding dispatching. Interdependencies between the different control strategy types became visible but were not the main scope of this study.

The last input factor which needs to be considered is the number of vehicles in the system. Here we see a limitation in increasing the amount of vehicles stepwise. This requires interpolating the performance between two system configurations and gives rise to a certain imprecision when deriving the additional vehicle requirements.

The four input factors which were mentioned so far all influence each other. The design of a transport system is therefore usually an iterative process (see Chapter 2.4). We consider the layout to be given externally and only modify the remaining three parameters to achieve operational system configurations. This has particular implications for the vehicle movement. More vehicles in a system decrease the waiting times. But on the other hand, they increase the traffic load which might lead to undesired congestion and deadlocks. In a design process the layout could be adjusted to prevent those consequences. This flexibility is not in our scope and limits the performance of the system.

Further limitations for our results are caused by the pure comparison of average values for the assessment of the control strategies. We try to limit the width of the confidence intervals in order to improve the reliability of the average figures. But the confidence intervals are not explicitly taken into account. To judge explicitly about significant differences, pairwise statistical hypothesis testing would be required. In addition, our analyses sometimes contain output data that has not shown stationarity according to both used statistical tests. For these reasons the computed vehicle and movement requirements should be seen as estimations which indicate a magnitude instead of precise figures. It would also be beneficial to supplement the average service times with quantiles and maximum values. As indicated above, this is difficult with this experimental setup as we use normalized scales and do not have any predefined performance level that is to be achieved. Therefore it is hard to assess which maximum and quantiles would still be acceptable and which would not.

Also for the derivation of the results for the disturbance scenarios a number of assumptions are required. As argued above, it is hard to find a meaningful benchmark for a system that experiences disturbances. We have chosen to create reference systems. But these involve side-effects (see Chapter 7.6.4) which need to be kept in mind when interpreting the results. The performance of the system after the disturbance always depends on its performance before the disturbance. Existing queues and vehicle distribution have an impact. In the context of disturbances it is hard to judge the robustness of a control system. Robustness always depends on a specific application and cannot be defined universally. We have chosen to use the central dispatching as a benchmark. But this requires the assumption that the central rules remain operational after the disturbance. This assumption would be violated in most real-world applications.

For the disturbance scenarios we only consider a comparably short time horizon and accept that a system might not reach stationarity. This requires keeping an eye on the achieved service times and the throughput at the same time. Otherwise no reasonable conclusions can be drawn. It has to be kept in mind that looking at the non-stationary case will never allow analyses which are as powerful and precise as what has been done for the additional vehicle requirements. In general, the analyzed disturbances can only be a first step towards understanding the robustness of decentral control systems. Disturbances were only tested in one specific layout with a defined number of vehicles and one type of input data set. As we have seen, these factors highly influence the system performance. The same is true for choosing $t_{dist}$. Using different input parameter combinations will most probably lead to other results.

Regarding our general assumptions the requirement of only using local information is very strong. In real-world systems it could for example be expected that the system entities quickly get information about the network topology. This information exchange should not require too much communication and can be expected to be feasible. Our limitation to pure local information does therefore definitely

represent a worst case scenario and the lowest decentral performance level which can be expected. However, we have found that the dispatching process would not work without mutual knowledge of sources and storage locations. A minimum of a-priori information is necessary. In addition, D4 slightly violates the local requirement by using stigmergy and giving feedback to the nodes along a chosen path. Although the pure focus on local information is a very demanding objective, it is fulfilled with only a few exceptions.

Summarizing, there are many assumptions and simplifications which limit the expressiveness of the results. But they are the only way to generate the desired results. Only simulation can be used for analyzing this kind of system. Using the generic test layout is the only option for enabling a comparison of control strategy performance in the scientific community. In order to use the generic layout the other input parameters (input data, control strategies, number of vehicles) need to be adjusted accordingly. Despite the mentioned limitations the test environment allows to develop first quantified insights into decentral control approaches and to achieve a better understanding of this system type. The real-world input data enhances the transferability. The applicable limitations just have to be kept in mind when the results are discussed. Based on the findings of this thesis the assumptions and analysis methodology can be improved in future studies.

# 9 Conclusions

The prior sections have shown the findings of our simulation experiments. Having the limitations of this study in mind, the final conclusions can now be derived. The table below summarizes the efficiency results for the decentral dispatching strategies. It states which part of the central benchmark performance (prfm) range each of the strategies is able to cover. The corresponding corridors for additional vehicle requirement and movement are displayed.

| Layout scenario | Control strategy | Input data set 1 | | | | | | Input data set 2 | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Covered prfm range [%] | | Add. vehicles [%] | | Add. vehicle movement [%] | | Covered prfm range [%] | | Add. vehicles [%] | | Add. vehicle movement [%] | |
| | | from | to | from | to | from | to | from | to | from | to | from | to |
| 1 | D2 | 0 | 97.5 | 26 | 35 | -2 | -1 | 0 | 94 | -69 | -41 | -56 | -41 |
| | D3 | 0 | 97.5 | 9 | 45 | 23 | 68 | 0 | 94 | -53 | -18 | -45 | -25 |
| | D4 | 0 | 97.5 | 11 | 43 | 7 | 30 | 45 | 88 | -59 | -51 | -52 | -33 |
| 2 | D2 | 30 | 97.5 | 68 | 113 | 4 | 8 | 33 | 95 | 31 | 84 | -27 | -26 |
| | D3 | 0 | 84 | 28 | 140 | 22 | 48 | 41 | 95 | -18 | -3 | -12 | -2 |
| | D4 | 70 | 84 | 35 | 42 | 46 | 58 | 0 | 60 | 7 | 23 | -1 | 16 |
| 3 | D2 | 55 | 95 | 66 | 124 | 12 | 15 | 0 | 80 | 141 | 213 | -1 | 4 |
| | D3 | 92 | 95 | 66 | 113 | 32 | 37 | 0 | 30 | 53 | 221 | 20 | 24 |
| | D4 | 0 | 84 | 36 | 107 | 42 | 91 | - | - | - | - | - | - |

Table 22: Overview of additional vehicle and movement requirements

Looking at the table it becomes evident that there is not a simple answer to the questions which were raised during the introduction to this thesis. The experiments have quantified the performance impact of only using local information for dispatching. The additional vehicle and movement requirements of

the decentral strategies clearly allow an assessment of their efficiency. But the results are too varying to derive a universal statement that explains the relationship between central and decentral dispatching strategies.

First of all, the decentral strategies are able to achieve the same throughput level as the central strategy. The only exception is D1 which was therefore excluded from the detailed analysis. Purely random dispatching cannot generate the desired throughput.

For the remaining control strategies the simulation experiments have shown that the achievable service time always depends on the used layout and the used input data set. This is true for central and decentral control strategies. The figure below tries to summarize the tendencies which can be derived for the two factors that influence the service time. The performance of the central strategy itself covers a certain range. The additional vehicle and movement requirement depends on the performance level within this range that the decentral strategy is supposed to achieve.

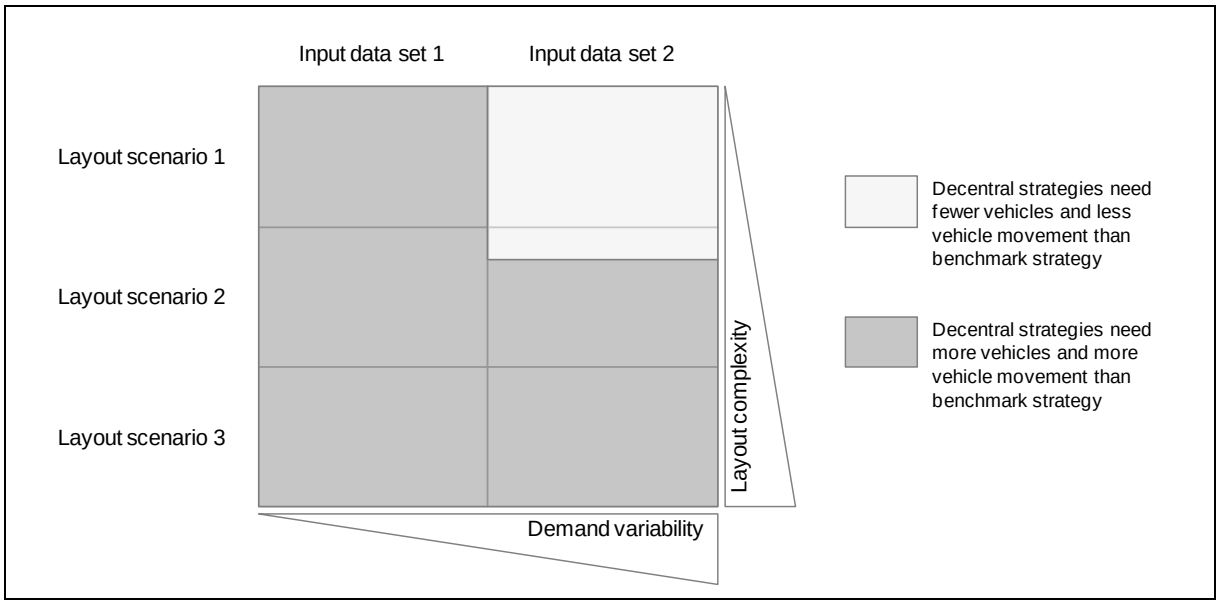



Figure 91: Impact of layout complexity and demand variability on dispatching efficiency

The central benchmark strategy performs worse than the decentral strategies when being applied in simple layouts in combination with input data set 2. Decision making is based on distances and therefore layout-dependent. In combination with a challenging load pattern which requires a high overall throughput and shows demand peaks at disadvantageous sources, the central performance deteriorates. An early load-vehicle assignment and the load-dependent source approach reinforce these results. The decentral strategies therefore require fewer vehicles than the benchmark control rule within the whole central performance range (see upper right corner in figure above). The central dispatching rule works well for all system configurations that process input data set 1. The throughput requirements of this data set are less variable with respect to the different sources. For small layouts the overall throughput is lower than for data set 2. In addition, the central strategy outperforms the decentral approaches in more complex layouts that use the second input data set. The decentral performance corridors diminish and the achieved service times decrease with the complexity of the layout.

The central dispatching rule which has been implemented is a generally accepted heuristic strategy. It does not strive for optimality and shows the mentioned weaknesses in small layouts when input data set 2 is used. The layout scenarios 1 and 2 illustrate that the decentral strategies are less vulnerable to

the disadvantageous load profile characteristics. They achieve a decent performance based on their simple decision rules. But it can be observed that the decentral performance decreases with growing layout size for both input data sets. As complex layouts are the relevant subject to evaluate real-world applicability, we try to answer the question why the decentral performance tends to deteriorate in such systems. The decentral dispatching strategies were developed for the purpose of this study. There are hardly any similar concepts available in the scientific literature. It is therefore crucial to analyze the behavior of these strategies to facilitate future implementations. This evaluation is the core topic for the rest of this chapter.

First, we take a look at dispatching strategy D2. Due to its simplicity it has an exceptional position among the other decentral rules. It only sends the vehicles back to their home locations once they have completed a job. All vehicles always travel via the storage location before they pick up a new load. Therefore the risk of mutual interferences and deadlocks is comparably low. D2 is the only strategy that delivers results for all 30 simulation setups and all vehicles configurations. However, due to the fixed assignment the strategy needs comparably many vehicles to reach a first operational system configuration. It does not try to use any real-time information about the network status and could also be considered to be "simple-minded" and possibly a worst case scenario. D3 and D4 are far more interesting for real-world applications. They both use local information about the current network status. They should therefore be expected to perform better than D2. This is the case for layout scenarios 2 and 3. In these more complex layouts D3 and D4 usually reach a performance which is somewhere between the central dispatching strategy and D2 if only few vehicles are used. But as more vehicles are added to the systems the performance deteriorates. This can especially be observed for input data set 2. Compared to D2 both D3 and D4 strategies always require more vehicle movement than D2. The following paragraphs therefore summarize the characteristics of these strategies and find reasons for their performance limitations compared to D2 and the central dispatching strategy. It will also become evident what induces the additional movement. In a first step the performance loss for systems with many vehicles is explained. Afterwards it is shown how the observed system characteristics restrict the performance in complex layouts.

D3 and D4 both apply procedures that lead to less precise dispatching decisions than D2 and D5. For D3 each sink locally collects information. The information is used to forecast and estimate the vehicle demand at sources and the fill levels at storage locations. Each sink has its own perception of the network and the information is not up to date once it is received. At least the transport time of the arriving load has passed since the information was initially generated. For all these reasons it happens frequently that sinks send vehicles to sources which no longer have a demand. Or the vehicles are sent to inappropriate storage locations. The same is true for D4. Here the dispatching tables collect information. As discussed above, the waiting time information is already outdated once the vehicles provide feedback to all switches they have visited during their last unladen trip. The dispatching tables add a random component to the dispatching decisions. Even for dispatching locations which have not provided feedback lately there is a small probability of being selected.

The lack of precision of the dispatching decisions leads to more vehicle movement. Vehicles are unnecessarily sent to locations where they are not required and need to be repositioned. Due to the resulting inefficiencies more vehicles might be required in the system.

During the development process it was found that the low dispatching precision makes it necessary to include a random component into the algorithms of D3 and D4. This is conceptually similar to using mutation approaches for genetic algorithms and applying techniques to mitigate stagnation in ACO systems (see SIM & SUN 2003). The release process avoids losing vehicles which were sent to disadvantageous sources and storage locations without demand for the remaining system transport

requirements at other locations. Vehicles are released and travel randomly in the network. This gives them the chance to find waiting loads at another source or storage location. They not only contribute to the overall transport performance but also explore the network and might find sources with waiting loads that currently are not contained in the dispatching process at any sink or switch.

But besides their positive characteristics for the performance of the system, the release processes also contribute to the traffic in the network. Together with the vehicle movement which is induced by the imprecise vehicle dispatching and additional vehicles that are required for the inherent inefficiencies, it increases the risk for mutual interferences of the vehicles. In particular for configurations with many vehicles, the probability for deadlocks grows tremendously. Additional reactive control strategies are required to avoid this undesired system behavior. We use load-dependent re-routing for preventing deadlocks. As the vehicles do not necessarily choose the shortest path to their destination this strategy might prevent deadlocks but increases the overall traffic load further. The inefficiencies reinforce themselves.

The dilemma seems to be that the decentral systems limit their own achievable performance. These effects become more severe in complex layouts. The decentral inefficiencies grow with the network size. In more complex layouts more dispatching destinations are available. This reduces the dispatching precision further. The probability of sending a vehicle to an inappropriate location grows with the number of available dispatching locations. It is less likely that the latest information about the status of all dispatching locations is available for decision making. In addition, the network structure is less suitable for randomly travelling vehicles. They need possibly more time to find a dispatching destination because the network is more branched. It can be concluded, that the pure use of local information scales up to the overall dispatching inefficiency when the layout size grows.

Two lines of thought are possible to solve this dilemma of the decentral strategies limiting their own performance. The first proposes a solution to deal with the consequences of the imprecise dispatching. The second approaches the roots of the problem:

1. Adjustment of the layout
   For implementing competitive decentral strategies it seems impossible to simply change the control components of a transport system. Instead it might be necessary to adjust the layout accordingly. The layout needs to be able to deal with the additional vehicle movement that is induced by the decentral control system.
2. Permission to communicate
   The sole usage of local information leads to inefficiencies. Therefore it might be required to allow the exchange of information among the logistics entities. Due to the communication overhead in large systems it seems impossible to allow the information exchange among all logistics entities. But it would be possible to divide the system in zones. Within those zones the logistics entities could exchange information and coordinate their actions. An alternative approach transfers the decision making capabilities to a limited amount of entities (e.g. the switches in the system) and allows the exchange of information among these. Neither approach makes system development easier and will in the end lead to the same questions which were asked in this thesis. However, the information exchange can be limited to a technologically feasible level.

But the lack of dispatching precision in D3 and D4 due to the limitation to local information does not only affect the vehicle movement and the number of required vehicles. It also has an impact on the maximum number of vehicles at each storage location. The dispatching strategies are not completely aware of the current fill levels and it might therefore happen that too many vehicles are sent to a depot.

Therefore the storage location capacity in decentral systems needs to be higher than in central systems. The exceedance of the storage location capacity by decentral control strategies in existing layouts might have a negative impact on the achievable performance. If the dispatching decision making processes cannot be improved, higher storage location capacities or additional circulation strategies are required for empty vehicles. This can on the one hand avoid the exceedance of storage location capacities and contribute to achieving acceptable performance levels. On the other hand, circulation strategies can help to reduce the required investment. The storage location at which the maximum number of vehicles can be expected is hard to predict. Therefore the same capacity has to be available at each source. D5 needs across all simulation setups and vehicle configuration between 25% and 85% less capacity than the decentral strategies. In order to become competitive the decentral requirements need to be reduced considerably. Circulation strategies would increase the traffic load and their influence on the system performance needs to be evaluated first.

Besides the mentioned aspects, our experiments revealed some other features of decentral systems that make their usage challenging. For the decentral control strategies even changing the parameter settings slightly can have a big impact on the achieved performance. The systems struggle with oversteering and understeering effects. They show a higher variability of service times than central systems. Sometimes an oscillating system behavior is observable. These properties can be seen in connection with the theory of complex self-organizing systems. The systems are hard to control because their cause-effect chains are not linear and feedback-loops are contained. The latter might lead to self-enforcing effects.

Our experiments with disturbance scenarios point in the same direction. The decentral strategies are able to adapt to changes in the environmental conditions. Oversteering effects and inertia limit the predictability of the system performance after the disturbance. Decentral control will always lead to delayed system response. In the case of changed load balance (e.g. a new source in the system) it takes some time until the system senses this change and vehicles start to react and adapt their behavior to the new situation.

It is difficult to assess the robustness of the decentral algorithm in comparison with central strategies. For the purpose of this thesis we have assumed that the central dispatching procedure perceives the change of the network status and remains operational. Otherwise the comparison of central and decentral performance would not be possible. But this assumption is violated in most real-world applications. It rather needs to be assumed that the central system is not operational any longer once an entity is added or extracted from the system. In the latter case, decentral algorithms would always be judged to perform better as they remain operational after the disturbance. The assumption of this thesis would lead to the opposite interpretation. A general statement is not possible and always depends on the specific situation and the capabilities of the considered central control system.

Looking at our results from the perspective of an applicability of decentral dispatching rules in real-world scenarios, it seems that there is still a lot of work left to do. The additional vehicle requirements would currently not justify the usage of decentral strategies. In layout scenario 3 depending on the input load data, at least 36% to 53% more vehicles and 20% to 42% more vehicle movement would be required to achieve the lowest central performance level. But real-world scenarios usually involve layouts that are far more complex and therefore an even higher additional vehicle requirement needs to be expected. In addition, the mentioned vehicle amounts only lead to the lowest achievable central performance and would need to grow further for more demanding performance levels. This would lead to an extreme increase in investment and operating costs. Additional disadvantages of the decentral algorithms need to be taken into account. But as shown above, the storage location capacities would currently restrict the application of decentral algorithms in existing layouts anyway. Additional studies

will be necessary to evaluate this aspect. Unless the rather simple rule D2 is accepted, the pure usage of local information seems not to be suitable for real-world applications. In addition, it needs to be kept in mind that this study only quantifies the impact of decentral dispatching. Other decentral control strategy types (e.g. routing) might also influence the additional vehicle requirements.

But this thesis only provides first insights regarding decentral dispatching strategies and helps to develop an understanding. Another topic that needs to be tackled is the derivation of appropriate methodologies for the development of this kind of system. A core question will be if a goal-directed development process is feasible at all. Based on the theory of self-organizing systems their behavior emerges from local interactions. Following this line of thought, the cause-effect chain is not necessarily known and cannot be used for the development of the control system. This would mean that the development process of such systems always remains an unsatisfying trial and error procedure.

What became evident during the experiments is that dispatching strategies cannot be analyzed in isolation. There are various interdependencies with routing and intersection control. Especially for the latter, this thesis contains some new approaches. Load-dependent re-routing has shown its ability to prevent deadlocks. Even in combination with central dispatching it performs well. Better system reliability could be achieved with only very little performance losses and additional movement. Simple and reactive decentral control rules in combination with central dispatching might be more beneficial than using an overall decentral control system.

# 10 Summary and future research

Our study was motivated by four shortcomings regarding decentral control systems which we explained during the introductory chapter of this thesis. The decentral control strategies which are proposed in the scientific literature either use global information or are technologically not feasible due to the induced communication overhead. In addition, their performance is not assessed properly in most publications and they only focus on the routing aspect of the control system. After the definition of the exact scope of this study, the four identified weaknesses were verified with an extensive literature review.

Based on these insights we developed a generic single-loop test layout for an intralogistics transport system. The system is scalable and allows the creation of transport systems with various degrees of complexity by duplicating the basic loop. Implemented in the AutoMod simulation software it can be used to quantify the performance of decentral strategies. At the same time it provides a test environment that can be used by other researchers and makes results of different studies comparable.

In this thesis we used the test layout for implementing and analyzing decentral dispatching strategies. Compared to routing, there exists only a little scientific research on dispatching. We were therefore forced to start from scratch by defining the core tasks that dispatching needs to fulfill with respect to load-vehicle assignment and empty vehicle positioning. The limitation to the usage of purely local information drove the algorithm design. In total we came up with four different decentral strategies which only use local information for making dispatching decisions. As decentral dispatching cannot use global knowledge of the system status, an efficiency loss needs to be expected.

In order to quantify this loss the decentral dispatching strategies were tested in six different scenarios that combined three different layout complexities with two different input data sets. While the layouts were created based on the generic test loop, the input data was taken from two real-world settings. They represent the transport requirements of a buffer replenishment application and a baggage handling system. To only model and test the decentral dispatching strategies would not be sufficient. The results were compared to the performance of a central benchmark dispatching strategy. The central strategy uses global information. Based on the simulation experiments we could derive a

detailed performance comparison. The number of vehicles in the system is treated as a variable input parameter for each of the six simulation setups. Therefore we can create a performance range of the central strategy with respect to the used amount of vehicles. For the decentral strategies it was afterwards determined how many additional vehicles and which increase in vehicle movement are required to reach a certain central performance level. These two dimensions allow a clear judgment of the efficiency of the decentral strategies. The exact quantification of decentral dispatching performance which does not require communication because it only uses local information exactly fills the research gap which we have identified.

In addition to the mentioned experiments, we took a look at disturbance scenarios. The adaptivity and robustness of decentral control strategies are continuously stated as their core advantages. But it is not analyzed in detail what that really means. We have defined three different disturbance types and assessed how they influence the performance of the different control strategies.

All tested control strategies are able to achieve the throughput which is induced by the different input data sets. The only exception is purely random dispatching. It turned out that it is hard to make a universal statement about the efficiency of the decentral control rules. In the case of fairly stable input data, the decentral strategies always need more vehicles than the central strategy. In the case of input data with high variability, this situation changes. In rather simply layouts, the decentral rules perform better than the central rule. They need fewer vehicles and less vehicle movement because their simple decision rules are not as vulnerable to disadvantageous load profile and layout characteristics as the central control approach. But as soon as the layouts get more complex, the decentral performance deteriorates. In complex layouts not all of the decentral strategies can even achieve the performance of the central strategy.

Except one very simple decentral rule that has the disadvantages of requiring comparably many vehicles to reach an acceptable performance, the characteristics of the remaining two decentral dispatching strategies limit their own performance. The forecast and feedback-based procedures lead to a lower precision of the dispatching decisions. It is induced by the pure usage of local information. The lack of precision leads to a self-enforcing increase in the vehicle and vehicle movement requirements that are not feasible within the given layouts. The decentral approaches might require other layouts to achieve a better performance. On the other hand it might be impossible to only use local information in complex systems. An information exchange at least within zones seems a promising approach to achieve a decent performance level. Based on investment and operating costs an application of decentral strategies in real-world situations seems not viable otherwise. In addition, the decentral strategies require a higher storage location capacity than the central rules. They show the unpredictable behavior of self-organizing systems. This can also be concluded from the disturbance scenarios.

The decentral strategies are able to deal with the three analyzed disturbances and remain operational. They seem to adapt slower to the changed environmental conditions than a central strategy could. But the assumption that the central control system is still operational after the disturbance is not applicable in many real-world scenarios. The long-term consequences of the disturbances are hard to predict for the decentral strategies. Oversteering and understeering effects can occur. In case they are relevant they can not only be observed for the next few hours but might even have an influence on the following few days after the disturbance occurred. For feedback-based dispatching the propagation of information about a network size reduction is required in order to ensure a decent performance.

This thesis contributes to the understanding of decentral control strategies but can only be seen as a first step in exploring their characteristics. We have introduced a test environment and developed our

own decentral dispatching strategies. Numerous parameters for analyzing and comparing central and decentral control strategies have been introduced. Future research should use this stage to elaborate the following topics.

Firstly, control strategies should be evaluated in more complex layouts. Based on the basic test loop, bigger layouts can be developed. Analyzing these layouts can help to understand if the performance of the decentral strategies decreases further when even more logistics entities are added to the system. This would be useful to support the impression that we got from our experiments and might supplement our thoughts on the self-limitation of the decentral system performance. The available input data sets can be used to test other throughput levels and different sink distributions. In a similar fashion other input data sets can be evaluated. Once the results from the additional analyses show a solid foundation, tests with real-world layouts should follow. At the same time using more sophisticated central dispatching rules could be considered. This would reduce the problems that we ran into for simple layouts with varying input data sets.

Further testing and improvement of D3 and D4 or the development of new decentral dispatching approaches are other fields of research. As we have shown in this paper there does not exist much scientific literature on decentral dispatching. In order to supplement the results of this paper it makes sense to improve the developed decentral strategies in a first step. The core question is if the dispatching precision can be improved despite only local information being used. The goal needs to be a performance improvement that at the same time reduces the required storage location capacity. Further parameterization tests can help to improve the understanding of decentral systems. For a limited range of simulation setups a “design of experiments” approach could identify all interdependencies between the input factors. Thereby a parameterization closer to optimality can be found. In addition, new thoughts regarding decentral dispatching should be explored by implementing new control rules. These can then be tested using the generic test environment. In this context it is necessary to evaluate which methodologies can support the development of decentral control systems. It needs to be examined if there are appropriate development methods for complex systems with self-organizing properties.

If the dispatching quality of the existing strategies cannot be improved and new approaches suffer from the same problems, it would be fruitful to examine how the storage location capacity influences the achievable performance of the decentral strategies. We have shown that decentral dispatching does in some cases exceed the physically available storage location capacity and might have an impact on the performance. It is therefore required to develop additional control rules which prevent this effect, e.g. by a circulation of idle vehicles. If those rules cannot be found or they show themselves a negative impact on the system performance, it might be necessary to analyze how the usage of the storage locations develops over time and how it can be predicted during the design phase of the system. It needs to be evaluated if decentral systems are economically reasonable in case those exceedances cannot be avoided.

The same prerequisites as for the last paragraph would trigger the exploration of the two research directions which have been mentioned as possible solutions for the decentral dilemma above. It would be interesting to examine how alternative layouts for decentral systems could look and how the exchange of information could be used to improve the efficiency of decentral decision making. This is an extremely broad topic. In this context it makes sense to analyze the results of research projects regarding routing and intersection control. Both control strategy types need to be taken into account when thinking about new layouts and communication. Zone-based routing protocols are already available.

The development of load-dependent routing and other intersection control rules was a by-product of this thesis. They were required to achieve operational system configurations for the decentral control rules. It has been shown that they can also be used beneficially in combination with central control systems. The further evaluation of the impact of these simple reactive control rules in other layout scenarios and in combination with different control systems seems very promising. In the same context the consequences of combining different decentral control strategy types needs to be analyzed. This thesis has only evaluated the performance impact of decentral dispatching. It is required to assess how decentral routing and other decentral approaches influence the system efficiency. At the same time the interdependencies of the different control strategy types should be examined.

# Appendix 1

The following tables show the results from the (Augmented) DF test and KPSS test. They are the basis for judging a system configuration operational. A check mark in the table indicates that the test favors stationarity, i.e. the (Augmented) DF test rejects the null hypothesis and the KPSS test accepts the null hypothesis. All analyses comprise the evaluation of the queue length at all sources in the specific simulation setup. In addition, the precision of the mean service time estimation is shown. It is used as an additional decision criterion. The first operational system configuration is shown in bold letters.

| **Layout scenario 1** | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Input data set 1** | | | | | | | | | | | | | | | | | |
| | | Standard experiments | | | | | | | | | | Additional experiments | | | | | |
| **D2** | # vehicles | 2 | 14 | 28 | 42 | 70 | 98 | 140 | 180 | 222 | 278 | 16 | 18 | 20 | **22** | 24 | 26 |
| | DF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | ✓ | ✓ | ✓ |
| | DF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | ✓ | ✓ | ✓ |
| | ADF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ |
| | Precision[%] | | | 4.4 | 3.0 | 1.0 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | | | | **7.1** | 5.4 | 4.7 |
| **D3** | # vehicles | 2 | 14 | 28 | 42 | 70 | 98 | 140 | 180 | 222 | 278 | 16 | **18** | 20 | 22 | 24 | 26 |
| | DF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | DF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 1 | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Precision[%] | | 6.1 | 13.1 | 2.7 | 2.2 | 1.3 | 1.0 | 1.0 | 1.0 | 6.1 | | **5.3** | 3.6 | 4.2 | 5.0 | 5.1 |
| **D4** | # vehicles | 2 | 14 | 28 | 42 | 70 | 98 | 140 | 180 | 222 | 278 | 16 | **18** | 20 | 22 | 24 | 26 |
| | DF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | DF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Precision[%] | | | 6.2 | 3.1 | 6.8 | 15.2 | 17.3 | 15.0 | 19.0 | 21.6 | | **13.7** | 5.0 | 4.2 | 4.7 | 5.8 |
| **D5** | # vehicles | 2 | 14 | 28 | 42 | 70 | 98 | 140 | 180 | 222 | 278 | 16 | **18** | 20 | 22 | 24 | 26 |
| | DF-test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| | DF-test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF-test 1 | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF-test 2 | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Precision[%] | | 4.5 | 3.8 | 1.1 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | 0.9 | | **4.1** | 4.2 | 4.7 | 4.6 | 4.5 |

**Layout scenario 1**

**Input data set 2**

| | | Standard experiments | | | | | | | | | | Additional experiments | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **D2** | # vehicles | 2 | 14 | 28 | 42 | 70 | 98 | 140 | 180 | 222 | 278 | 30 | 32 | 34 | 36 | **38** | 40 |
| | DF test 1 | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | DF test 2 | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 1 | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ |
| | KPSS test | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Precision[%] | | | | 7.6 | 7.1 | 7.0 | 7.0 | 7.0 | 7.0 | 7.0 | | | | | **7.8** | 7.5 |
| **D3** | # vehicles | 2 | 14 | **28** | 42 | 70 | 98 | 140 | 180 | 222 | 278 | | | | | | |
| | DF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | DF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | ADF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | ADF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | KPSS test | | | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | Precision[%] | | | **10.0** | 27.1 | 8.8 | 11.5 | 10.6 | 10.6 | 10.6 | 10.6 | | | | | | |
| **D4** | # vehicles | 2 | 14 | 28 | 42 | 70 | 98 | 140 | 180 | 222 | 278 | 30 | 32 | 34 | **36** | 38 | 40 |
| | DF test 1 | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | DF test 2 | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | | | ✓ | ✓ | ✓ | | ✓ | | ✓ | | | | ✓ | ✓ | ✓ |
| | Precision[%] | | | | 6.0 | 7.7 | 9.5 | 23.4 | 27.2 | 44.9 | 34.4 | | | | **6.4** | 5.7 | 6.1 |
| **D5** | # vehicles | 2 | 14 | 28 | 42 | 70 | 98 | 140 | 180 | 222 | 278 | | | 54 | 58 | 62 | **66** |
| | DF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ |
| | DF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ |
| | ADF test 1 | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | ✓ |
| | ADF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ |
| | Precision[%] | | | | | 6.6 | 5.7 | 5.4 | 6.7 | 7.0 | 7.0 | | | | | | **6.6** |

**Layout scenario 2**

**Input data set 1**

| | | Standard experiments | | | | | | | | | | Additional experiments | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **D2** | # vehicles | 8 | 56 | 112 | 168 | 280 | 392 | 560 | 720 | 888 | 1112 | 120 | 128 | 136 | **144** | 152 | 160 |
| | DF test 1 | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | ✓ | ✓ | ✓ |
| | DF test 2 | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | ✓ | ✓ | ✓ |
| | ADF test 1 | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Precision[%] | | | | 0.8 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | 0.7 | | | | **3.1** | 1.7 | 1.3 |
| **D3** | # vehicles | 8 | 56 | 112 | 168 | 280 | 392 | 560 | 720 | 888 | 1112 | 64 | 72 | **80** | 88 | 96 | 104 |
| | DF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| | DF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | ✓ | ✓ | ✓ | ✓ |
| | Precision[%] | | | 4.3 | 5.8 | 6.3 | 5.6 | 5.3 | 4.4 | 3.3 | 8.3 | | | **4.4** | 4.1 | 6.5 | 5.2 |
| **D4** | # vehicles | 8 | 56 | 112 | 168 | 280 | 392 | 560 | 720 | 888 | 1112 | **120** | 128 | 136 | 144 | 152 | 160 |
| | DF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | DF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | | ✓ | |
| | Precision[%] | | | | 6.5 | 16.3 | 35.8 | 29.0 | 15.2 | 17.6 | 27.3 | **4.7** | 6.4 | 6.7 | 6.9 | 4.9 | 7.1 |
| **D5** | # vehicles | 8 | 56 | 112 | 168 | 280 | 392 | 560 | 720 | 888 | 1112 | **64** | 72 | 80 | 88 | 96 | 104 |
| | DF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | DF test 2 | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 1 | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Precision[%] | | | 2.9 | 2.5 | 0.9 | 0.8 | 0.7 | 0.7 | 0.7 | 0.7 | **1.8** | 2.1 | 2.5 | 2.4 | 2.5 | 2.5 |

| Layout scenario 2 | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Input data set 2** | | | | | | | | | | | | | | | | | |
| | | Standard experiments | | | | | | | | | | Additional experiments | | | | | |
| **D2** | # vehicles | 8 | 56 | 112 | 168 | 280 | 392 | 560 | 720 | 888 | 1112 | 184 | 200 | 216 | 232 | **248** | 264 |
| | DF test 1 | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | ✓ | ✓ | ✓ |
| | DF test 2 | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | ✓ | ✓ | ✓ |
| | ADF test 1 | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | ✓ | ✓ |
| | ADF test 2 | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | ✓ | ✓ | ✓ |
| | KPSS test | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | ✓ | ✓ | ✓ |
| | Precision[%] | | | | | 5.0 | 3.2 | 3.0 | 3.0 | 3.0 | 3.0 | | | | | **5.5** | 5.0 |
| **D3** | # vehicles | 8 | 56 | 112 | 168 | 280 | 392 | 560 | 720 | 888 | 1112 | **120** | 128 | 136 | 144 | 152 | 160 |
| | DF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | DF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | | | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Precision[%] | | | | 4.9 | 4.3 | 4.2 | 3.5 | 2.9 | 3.1 | 9.1 | **8.7** | 7.0 | 5.6 | 4.9 | 5.6 | 5.2 |
| **D4** | # vehicles | 8 | 56 | **112** | 168 | 280 | 392 | 560 | 720 | 888 | 1112 | | | | | | |
| | DF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | DF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | ADF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | ADF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | KPSS test | | | ✓ | | ✓ | ✓ | ✓ | | | | | | | | | |
| | Precision[%] | | | **10.2** | 10.2 | 6.8 | 6.1 | 7.6 | 7.6 | 7.8 | 19.9 | | | | | | |
| **D5** | # vehicles | 8 | 56 | 112 | 168 | 280 | 392 | 560 | 720 | 888 | 1112 | **120** | 128 | 136 | 144 | 152 | 160 |
| | DF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | DF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 1 | | | | ✓ | ✓ | ✓ | ✓ | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | | ✓ | ✓ | ✓ | ✓ | ✓ | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Precision[%] | | | | 9.6 | 5.3 | 4.8 | 4.2 | | | | **14.6** | 11.4 | 10.5 | 9.7 | 9.5 | 9.7 |

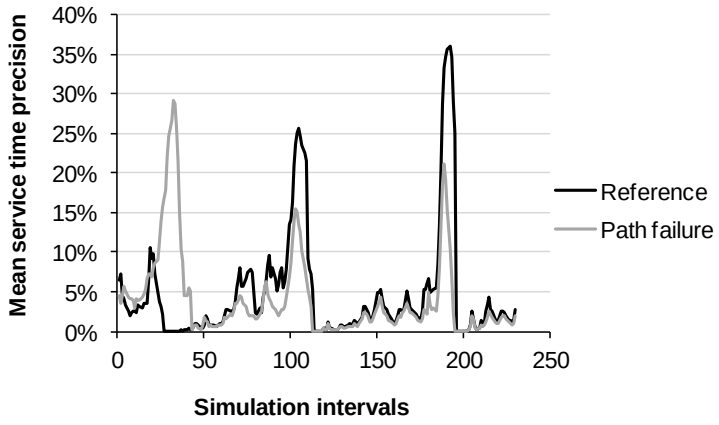
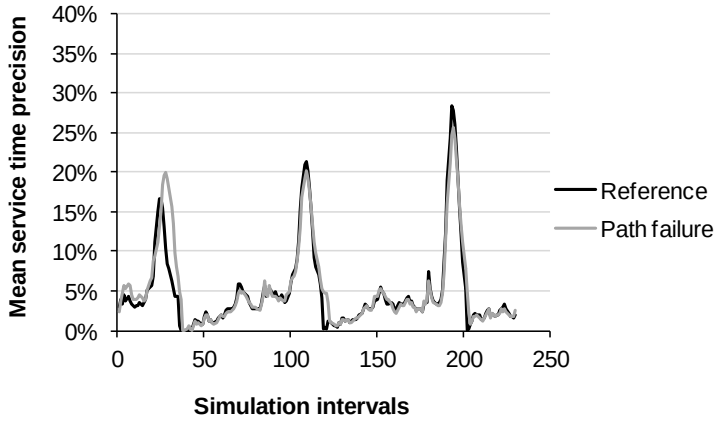
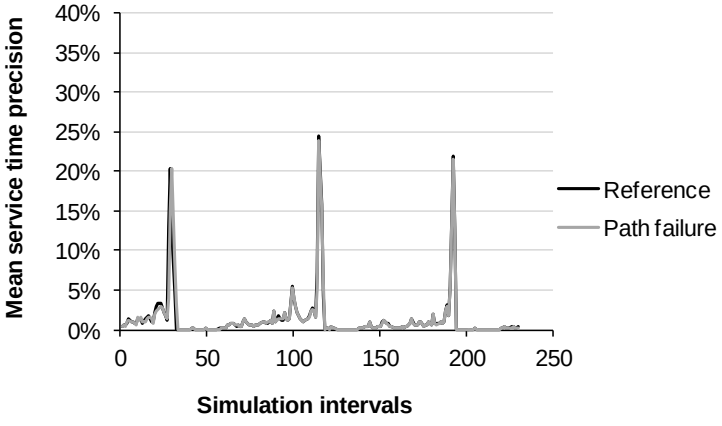
| Layout scenario3 | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Input data set 1** | | | | | | | | | | | | | | | | | |
| | | Standard experiments | | | | | | | | | | Additional experiments | | | | | |
| **D2** | # vehicles | 18 | 126 | 252 | 378 | 630 | 882 | 1260 | 1620 | 1998 | 2502 | **270** | 288 | 306 | 324 | 342 | 360 |
| | DF test 1 | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | DF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 1 | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Precision[%] | | | | 1.9 | 2.6 | 2.2 | 1.9 | 1.8 | 1.8 | 1.8 | **1.3** | 1.1 | 1.2 | 1.3 | 1.6 | 1.8 |
| **D3** | # vehicles | 18 | 126 | 252 | 378 | 630 | 882 | 1260 | 1620 | 1998 | 2502 | 270 | 288 | 306 | **324** | 342 | 360 |
| | DF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | DF test 2 | | | | ✓ | ✓ | ✓ | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | | | ✓ | ✓ | ✓ | ✓ | | ✓ | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Precision[%] | | | | 15.3 | 7.1 | 5.9 | 5.6 | | 16.8 | | | 16.1 | 15.1 | **13.5** | 13.3 | 16.4 |
| **D4** | # vehicles | 18 | 126 | 252 | 378 | 630 | 882 | 1260 | 1620 | 1998 | 2502 | 144 | 162 | 180 | 198 | **216** | 234 |
| | DF test 1 | | | ✓ | ✓ | ✓ | ✓ | | ✓ | | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| | DF test 2 | | | ✓ | ✓ | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 1 | | | ✓ | ✓ | ✓ | ✓ | | ✓ | | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | | ✓ | ✓ | ✓ | ✓ | | ✓ | | | | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | | ✓ | ✓ | | | | | | | | | | | ✓ | ✓ |
| | Precision[%] | | | 6.8 | 8.7 | 7.9 | 7.3 | | 102 | | | | | | | **7.7** | 7.7 |
| **D5** | # vehicles | 18 | 126 | 252 | 378 | 630 | 882 | 1260 | 1620 | 1998 | 2502 | **144** | 162 | 180 | 198 | 216 | 234 |
| | DF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | DF test 2 | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 1 | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | Precision[%] | | | 9.2 | 2.5 | 1.5 | 1.3 | 1.3 | 1.2 | 1.2 | 1.2 | **7.2** | 8.0 | 9.1 | 10.2 | 10.7 | 9.7 |

| Layout scenario 3 | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Input data set 2** | | | | | | | | | | | | | | | | | |
| | | Standard experiments | | | | | | | | | | | Additional experiments | | | | |
| **D2** | # vehicles | 18 | 126 | 252 | 378 | **630** | 882 | 1260 | 1620 | 1998 | 2502 | | | | | | |
| | DF test 1 | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | DF test 2 | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | ADF test 1 | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | ADF test 2 | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | KPSS test | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | |
| | Precision[%] | | | | | **12.6** | 6.8 | 2.8 | 2.8 | 2.7 | 2.7 | | | | | | |
| **D3** | # vehicles | 18 | 126 | 252 | 378 | 630 | 882 | 1260 | 1620 | 1998 | 2502 | 270 | **288** | 306 | 324 | 342 | 360 |
| | DF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | DF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 1 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | | | ✓ | | | | ✓ | | | | ✓ | ✓ | | | |
| | Precision[%] | | | 17.1 | 9.1 | 8.3 | 10.0 | 11.3 | 12.4 | 12.0 | | | **14.9** | 14.0 | 8.9 | 11.5 | 7.8 |
| **D4** | # vehicles | 18 | 126 | **252** | 378 | 630 | 882 | 1260 | 1620 | 1998 | 2502 | 666 | 702 | 738 | 774 | 810 | 846 |
| | DF test 1 | | | ✓ | ✓ | ✓ | | | | | | ✓ | | ✓ | ✓ | | |
| | DF test 2 | | | ✓ | ✓ | ✓ | | | | | | ✓ | | ✓ | ✓ | | |
| | ADF test 1 | | | ✓ | ✓ | ✓ | ✓ | | | | | ✓ | | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | | ✓ | ✓ | ✓ | | | | | | ✓ | | ✓ | ✓ | ✓ | |
| | KPSS test | | | | | | | | | | | | | | | | |
| | Precision[%] | | | **18.2** | 13.0 | 16.3 | 136 | | | | | 14.4 | 24.1 | 16.8 | 34.5 | 141 | 141 |
| **D5** | # vehicles | 18 | 126 | 252 | 378 | 630 | 882 | 1260 | 1620 | 1998 | 2502 | 144 | 162 | 180 | 198 | 216 | **234** |
| | DF test 1 | | | ✓ | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | DF test 2 | | | ✓ | ✓ | ✓ | | | | | | | | ✓ | ✓ | ✓ | ✓ |
| | ADF test 1 | | ✓ | ✓ | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | ADF test 2 | | ✓ | ✓ | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | KPSS test | | ✓ | ✓ | ✓ | ✓ | | | | | | ✓ | | ✓ | ✓ | ✓ | ✓ |
| | Precision[%] | | | 10.8 | 6.5 | 3.5 | | | | | | | | | | | **14.7** |

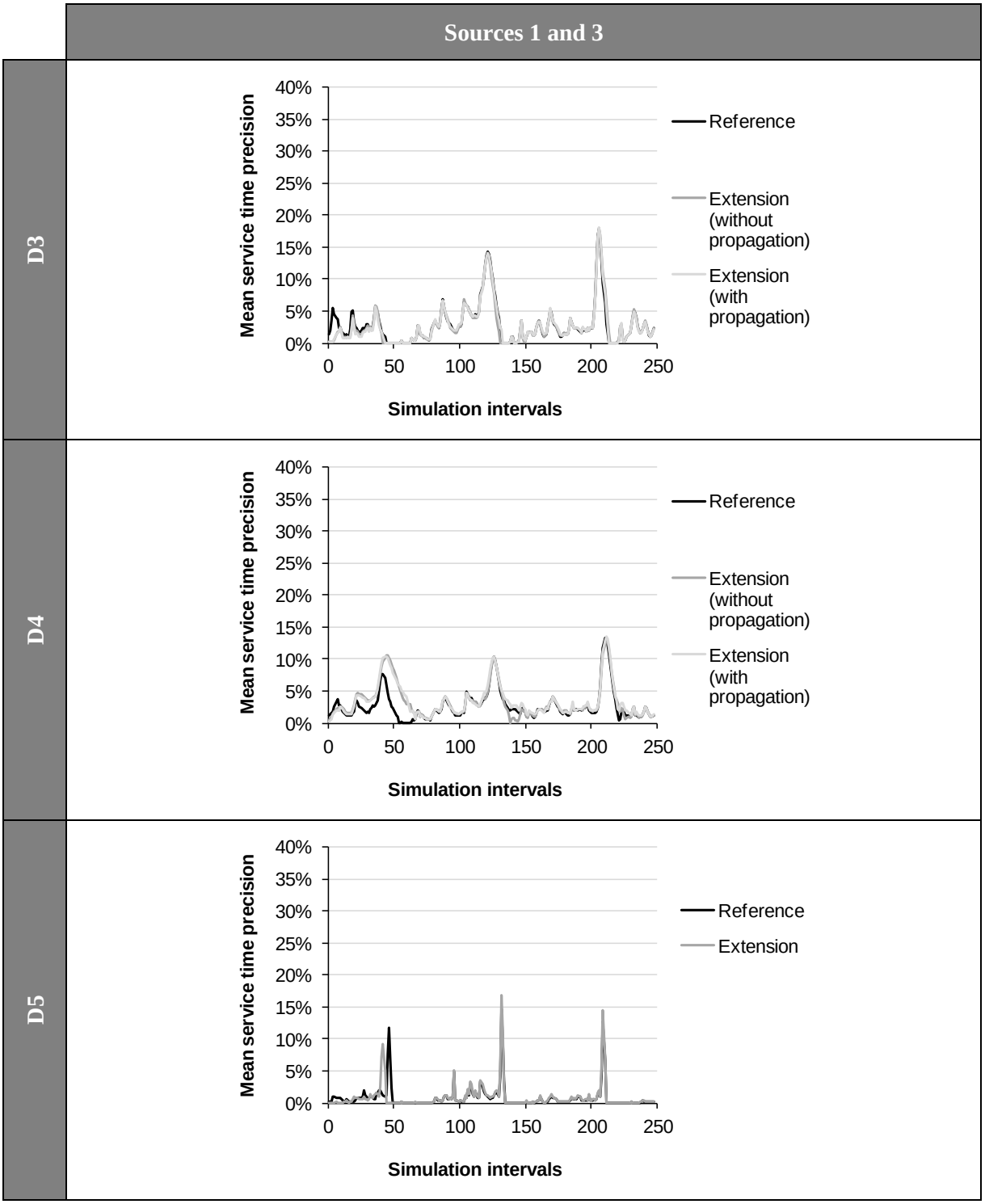
# Appendix 2

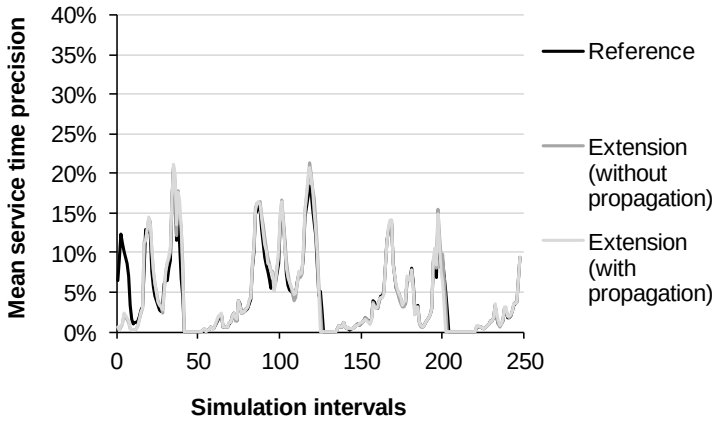
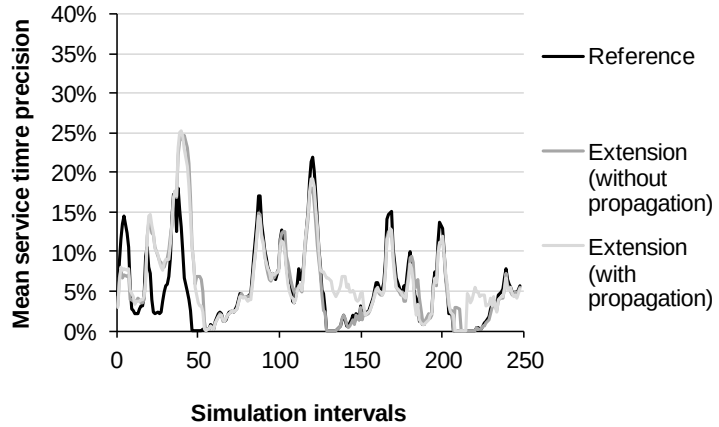
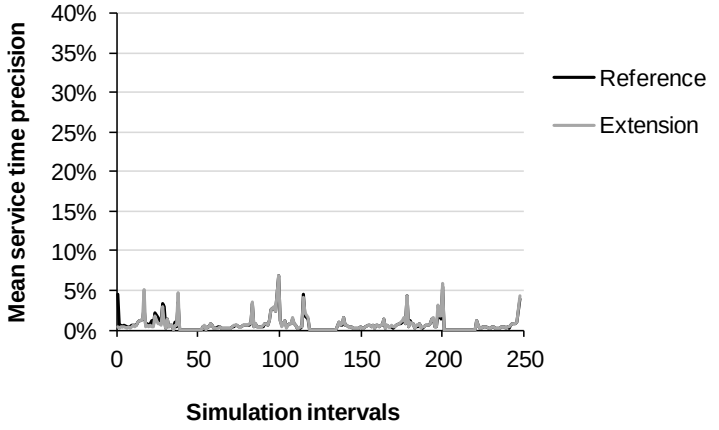
The figures below summarize the achieved precision of the mean service time estimation for the disturbance scenarios. The precision of the estimation for each simulation interval is shown. It becomes evident that the biggest relative errors occur directly before idle periods for most of the scenarios.

**Path failure scenario**

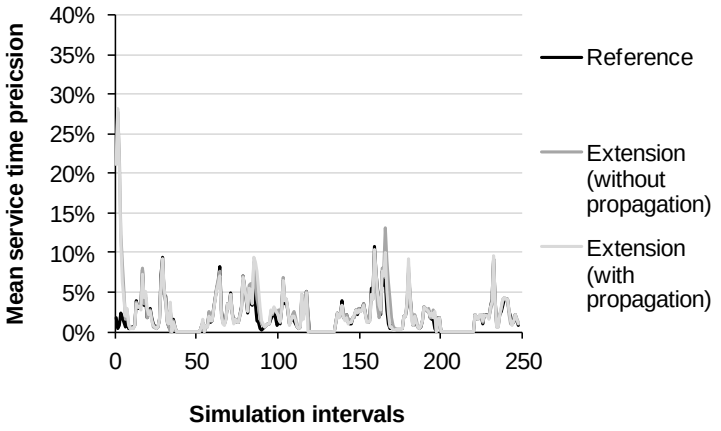
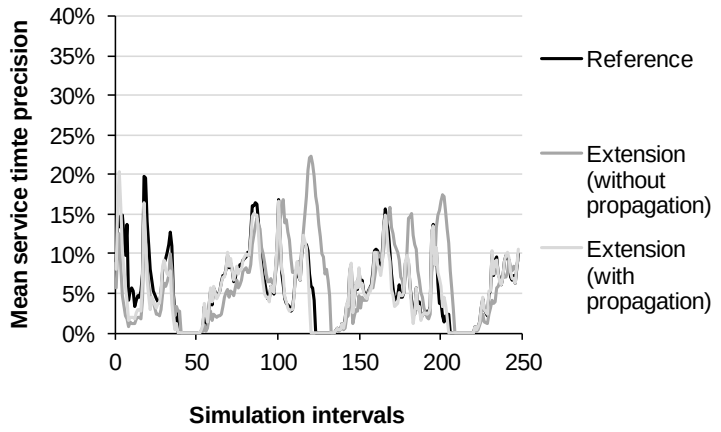
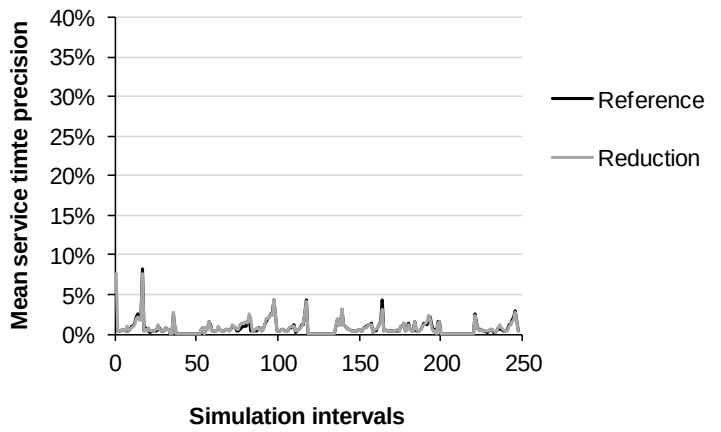
| | |
|---|---|
| D3 |  |
| D4 |  |
| D5 |  |

**Extension scenario**



| | Sources 2 and 4–8 |
|---|---|
| D3 |  |
| D4 |  |
| D5 |  |

**Reduction scenario**

| | |
|---|---|
| D3 |  |
| D4 |  |
| D5 |  |

# Bibliography


Abolhasan, M., Wysocki, T. & Dutkiewicz, E. (2004): A review of routing protocols for mobile ad hoc networks. Ad Hoc Networks 2(1): 1–22.

Arora, S., Raina, A.K. & Mittal, A.K. (2000): Collision avoidance among AGVs at junctions. Pages 585–589 in Proceedings of the IEEE Intelligent Vehicles Symposium 2000. IEEE Press.

Ay, N. (2006): Prinzipien der Robustheit, Acitivity report 2006 of the Max-Planck-Institute for Mathematics in Sciences, Leipzig.

Bartholdi III, J.J. & Platzman, L.K. (1989): Decentralized control of automated guided vehicles on a simple loop. IIE Transactions 21 (1): 76–81.

Berman, S. & Edan, Y. (2002): Decentralized autonomous AGV system for material handling. International Journal of Production Research 40(15): 3995–4006.

Bond, A.H. & Gasser, L. (Eds.) (1988): Readings in Distributed Artificial Intelligence. Morgan Kaufmann Publishers, San Mateo.

Bozer, Y.A. & Srinivasan, M.M. (1992): Tandem AGV systems: a partitioning algorithm and performance comparison with conventional AGV systems. European Journal of Operational Research 63(2): 173–191.

Bozer, Y.A. & Yen, C. (1996): Intelligent Dispatching rules for Trip-based Material Handling Systems. Journal of Manufacturing Systems 15(4): 226–239.

Chang, S.-H. & Egbelu, P.J. (1996): Dynamic relative positioning of AGVs in a loop layout to minimize mean system response time. International Journal of Production Research 34(6): 1655–1673.

Charezma, W.W. & Syczewska, E.M. (1998): Joint Application of the Dickey-Fuller and KPSS tests. Economics letters 61(1): 17–21.

Chisu, R., Kuzmany, F. & Günthner, W.A. (2010): Realisierung einer agentenbasierten Steuerung für Elektrohängebahnsysteme. Pages 263–274 in Internet der Dinge in der Intralogistik (M. ten Hompel & W.A. Günthner, Eds.). Springer Verlag, Berlin.

Claes, R., Holvoet, T. & Weyns, D. (2011): A Decentralized Approach for Anticipatory Vehicle Routing Using Delegate Multiagent Systems. IEEE Transactions on Intelligent Transportation Systems 12: 364–373.

Clausen, U., Kaffka, J., Diekmann, D. & Mest, L. (2011): Impact of different unloading zone locations in transshipment terminals under various forklift dispatching rules. Pages 1658–1667 in Proceedings of the 2011 Winter Simulation Conference (S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, & M. Fu, Eds.). IEEE Press.

Coffman, E.G., Elphick, M. & Shoshani, A. (1971): System deadlocks. ACM Computing Surveys (CSUR) 3(2): 67–78.

Daniluk, D. & Chisu, R., (2010): Simulation und Emulation im Internet der Dinge. Pages 149–166 in Internet der Dinge in der Intralogistik (M. ten Hompel & W.A. Günthner, Eds.). Springer Verlag, Berlin.

De Koster, R., Le-Anh, T. & van der Meer, J.R. (2004): Testing and classifying vehicle dispatching rules in three real-world settings. Journal of Operations Management 22(4): 369–386.

De Koster, R. & van der Meer, J.R. (1998): Centralized vs. Decentralized control of internal transport: a case study. Pages 403–420 in Advances in Distribution Logistics (J.A.E.E. van Nunen, Ed.), Springer Verlag, Berlin.

Deutsches Institut für Normung (DIN) (Eds.) (1994): DIN 19226, Regelungstechnik und Steuerungstechnik, Part 1–6. Beuth Verlag, Berlin.

Di Caro, G. & Dorigo, M. (1998): AntNet: Distributed stigmergetic control for communications networks. Journal of Artifical Intelligence Research 9: 317–365.

Dickey, D.A. & Fuller, W.A. (1979): Distribution of the Estimators for Autoregressive Time Series With a Unit Root. Journal of the Americal Statistical Association 74(366): 427–431.

Dhillon, S.S. & Van Mieghem, P. (2007): Performance analysis of the AntNet algorithm. Computer Networks 51(8): 2104–2125.

Dijkstra, E.W. (1959): A Note on Two Problems in Connexion with Graphs. Numerische Mathematik 1: 269–271.

Dorigo, M. & Di Caro, G. (1999): Ant colony optimization: a new meta-heuristic. Pages 1470–1477 in Proceedings of the Congress on Evolutionary Computation. IEEE Press.

Dorigo, M., Maniezzo, V. & Colorni, A. (1996): Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 26(1): 29–41.

Egbelu, P.J. & Tanchoco, J.M.A. (1984): Characterization of automatic guided vehicle dispatching rules. International Journal of Production Research 22(3): 359–374.

Fay, A. & Fischer, I. (2004): Dezentrale Automatisierungsstrategien für Gepäckbeförderungssysteme. Automatisierungstechnik 52 (7): 335–341.

Fay, A., Jerenz, S. & Seitz, N. (2008): Dezentrale Steuerung von Transportsystemen in Analogie zum Routing in Datennetzen. at-Automatisierungstechnik 56 (6): 284–295.

Fleischmann, B. (2003): Grundlagen: Begriffe der Logistik, logistische Systeme und Prozesse. Pages 3–34 in Handbuch Logistik (D. Arnold, H. Isermann, A.Kuhn, H. Tempelmeier & K. Furmanns, Eds.). Springer Verlag, Berlin.

Frazzoli, E., Pallotino, L., Scordio, V. & Bicchi, A. (2005): Decentralized cooperative conflict resolution for multiple non-holononic vehicles. IEEE Transactions on Robotics 23(6):1170–1183.

Furmans, K., Nobbe, C. & Schwab, M. (2011). Future of Material Handling – modular, flexible and efficient. Conference paper for the IEEE/RSJ International Conference on Intelligent Robots and Systems. San Francisco.

Gademann, A. & van de Velde, S.L. (2000): Positioning automated guided vehicles in a loop layout. European Journal of Operational Research 127(3): 565–573.

Gribble, S.D. (2001): Robustness in complex systems. Pages 21–26 in Proceedings of the 8th workshop on hot topics in operating systems.

Göhring, S. & Lorenz, T. (2010): Agentenbasierte Staplerleitsysteme. Pages 313–327 in Internet der Dinge in der Intralogistik (M. ten Hompel & W.A. Günthner, Eds.). Springer Verlag, Berlin.

Gudehus, T. (2005): Logistik: Grundlagen – Strategien – Anwendungen. Springer Verlag, Berlin.

Günthner, W.A., Chisu, R. & Kuzmany, F. (2008): Internet der Dinge – Intelligent verteilt. F+H Fördern und Heben 9: 494-497.

Hackl, P. (2004): Einführung in die Ökonometrie. Pearson Education, Munich.

Hahn-Woernle, C. (2010): Neue Anforderungen für die Logistik des 21. Jahrhunderts. Pages 9–13 in Internet der Dinge in der Intralogistik (M. ten Hompel & W.A. Günthner, Eds.). Springer Verlag, Berlin.

Hallenborg, K. (2007): Decentralized scheduling of baggage handling using multi-agent technologies. Pages 481–404 in Multiprocessor Scheduling (Levner, E, Ed.). Itech Education and Publishing, Vienna.

Hammel, C., Flemming, A., Peters, K. & Schulze, F. (2008): Anwendung von Methoden aus der Theorie Komplexer Netzwerke für die Optimierung der Layouts von MFS. Pages 81–91 in 4. Fachkolloquium der Wissenschaftlichen Gesellschaft für Technische Logistik (WGTL) (TU Chemnitz, Professur Fördertechnik, Ed.).

Heylighen, F. (2001): The science of self-organization and adaptivity. In Knowledge Management, Organizational Intelligence and Learning, and Complexity (L.D. Kiel, Ed.). In The Encyclopedia of Life Support Systems. EOLSS Publishers, Oxford.

Hippenmeyer, H., Furmans, K., Stoll, T. & Schönung, F. (2009): KARIS – dezentral gesteuert. Hebezeuge – Fördermittel 49: 304–306.

Hofmeister, M., Baier, G. & Gärtner, M. (2010): Strategien für die dezentrale agentenbasierte Steuerung von Materialflusssystemen. Pages 119–139 in Internet der Dinge in der Intralogistik (M. ten Hompel & W.A. Günthner, Eds.). Springer Verlag, Berlin.

Hu, C.-H. & Egbelu, P.J. (2000): A framework for the selection of idle vehicle home locations in an automated guided vehicle system. International Journal of Production Research 38(3): 543–562.

Ickert, L., Matthes, U., Rommerskirchen, S., Weyand, E., Schlesinger, M. & Limbers, J. (2007): Abschätzung der langfristigen Entwicklung des Güterverkehrs in Deutschland bis 2050 – Final report of project 26.0185/2006 for the German ministry of traffic.

Ioannou, P. & Sun, J. (1995): Robust Adaptive Control. Prentice Hall, Upper Saddle River.

Jacquet, P., Mühletaler, T., Clausen, T., Laouiti, A., Qayyum, A. & Viennot, L. (2001): Optimized link state routing protocol for ad hoc networks. Pages 62–68 in Proceedings of IEEE Multi Topic Conference: Technology for the 21st Century. IEEE Press.

Johnson, D.B. & Maltz, D.A. (1996): Dynamic source routing in ad-hoc wireless networks. In Mobile Computing 353: 153–181.

Johnstone, M., Creighton, D. & Nahavandi, S. (2010): Status-based Routing in Baggage Handling Systems: Searching Verses Learning. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 40(2): 189–200.

Jünemann, R. & Beyer, A. (1998): Steuerung von Materialfluss- und Logistiksystemen. Springer Verlag, Berlin.

Kheir, N.A. (1995): Systems Modeling and Computer Simulation. Marcel Dekker Inc., New York.

Kim, K.H. & Kim, J.Y. (1997): Estimating mean response time and positioning idle vehicles of automated guided vehicle systems in loop layout. Computers & Industrial Engineering 33(3–4): 669–672.

Klein, N. (2008): Multiagentensysteme: Theoretische Grundlagen und Einsatz zur dezentralen Steuerung automatisierter Materialflusssysteme. Vdm Verlag Dr. Müller, Saarbrücken.

Koo, P.H. & Jang, J. (2002): Vehicle travel time models for AGV systems under various dispatching rules. International Journal of Flexible Manufacturing Systems 14(3): 249–261.

Koo, P.-H., Jang, J. & Suh, J. (2005): Vehicle dispatching for highly loaded semiconductor production considering bottleneck machines first. International Journal of Flexible Manufacturing Systems 17: 23–38.

Kratky, K.W. (1990): Der Paradigmenwechsel von der Fremd- zur Selbstorganisation. Pages 3–17 in Grundprinzipien der Selbstorganisation (K.W. Kratky, F. Wallner, Ed.). Wissenschaftliche Buchgesellschaft, Darmstadt.

Kwiatkowski, D., Phillips, C.P.B., Schmidt, P. & Shin, Y. (1992): Testing the null hypothesis of stationarity against the alternative of a unit root. Journal of Econometrics 54: 159–178.

Lau, H.Y.K. & Woo, S.O. (2008): An agent-based dynamic routing strategy for automated material handling systems. International Journal of Computer Integrated Manufacturing 21(3): 269–288.

Law, A.M. (2006): Simulation Modeling and Analysis. McGraw-Hill Professional, New York.

Law, A.M. & Kelton, W.D. (1984): Confidence Intervals for Steady-State Simulations, I: A Survey of Fixed Sample Size Procedures. Operations Research 32: 1221–1239.

Le-Anh, T. (2005): Intelligent Control of Vehicle-Based Internal Transport Systems. Ph.D, dissertation, Erasmus University Rotterdam.

Le-Anh, T. & De Koster, M.B.M. (2004): Multi-Attribute Dispatching Rules For AGV Systems with Many Vehicles. Research Paper at the ERASMUS Research Institute of Management.

Le-Anh, T. & De Koster, M.B.M. (2005): On-line dispatching rules for vehicle-based internal transport systems. International Journal of Production Research 43(8): 1711–1728.

Le-Anh, T. & De Koster, M.B.M. (2006): A review of design and control of automated guided vehicle systems. European Journal of Operational Research 171(1): 1–23.

Lehmann, M. (2006): Einsatzplanung von Fahrerlosen Transportsystemen in Seehafen-Containerterminals. Ph.D, dissertation, Berlin University of Technology.

Liebl, F. (1995): Simulation: Problemorientierte Einführung. Oldenbourg Wissenschaftsverlag, Munich.

Mantel, R.J. & Landeweerd, H.R.A. (1995): Design and operational control of an AGV system. International Journal of Production Economics 41(1): 257–266.

Maxwell, W.L. & Muckstadt, J.A. (1982): Design of Automatic Guided Vehicle Systems. IIE Transactions 14(2): 114–124.

Mayer, S. (2009): Development of a completely decentralized control system for modular continuous conveyors. Ph.D, dissertation, Karlsruhe Institute of Technology.

Mayer, S. & Furmans, K. (2010): Deadlock prevention in a completely decentralized controlled materials flow systems. Logistics Research 2:147–158.

Min, T.W., Zhe, L, Yin, K., Hiang, G.C. & Yuong, L.K. (1999): A rules and communication based multiple robots transportation system. Pages 180–186 in Proceedings of the 1999 IEEE International Symposium on In Computational Intelligence in Robotics and Automation (CIRA'99). IEEE Press.

Neumann, K. & Morlock, M. (2002): Operations Research. Hanser Fachbuch, Munich.

Nieke, C. (2010): Materialflusssteuerung heute und ihre Defizite. Pages 15–18 in Internet der Dinge in der Intralogistik (M. ten Hompel & W.A. Günthner, Eds.). Springer Verlag, Berlin.

Nowe, A., Verbeeck, K. & Peeters, M. (2006): Learning Automata as a Basis for Multi-agent Reinforcement Learning. Pages 71–85 in Learning and adaption in multiagent systems (G. Weiß, S. Sen, Eds.). Springer Verlag, Berlin.

Qiu, L., Hsu, W.-J., Huang, S.-Y. & Wang, H. (2002): Scheduling and routing algorithms for AGVs: A survey. International Journal of Production Research 40(3): 745–760.

Olmi, R. (2011): Traffic Management of Automated Guided Vehicles in Flexible Manufacturing Systems. Ph.D, dissertation, University of Ferrara.

Overmeyer, L.; Krühn, T.; Hahn, A. & Pinkowski, J. (2012): CogniLog – Cognitive Logistics for Warehousing. Pages 104–122 in Coordinated Autonomous Systems – Proceedings of the 6th International Scientific Symposium on Logistics 2012 (W. Delfmann, T. Wimmer, Eds.). DVV Media Group, Hamburg.

R Development Core Team (2012): R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna. ISBN 3-900051-07-0, URL http://www.R-project.org/

Rinne, H. (2008): Taschenbuch der Statistik. Verlag Harri Deutsch, Frankfurt.

Ross, E.A., Mahmoodi, F. & Mosier, C.T. (1996): Tandem Configuration Automated Guided Vehicle Systems: A Comparative Study. Decision Sciences 27(1): 81–102.

Sastry, S. & Bodson, M. (2011): Adaptive Control: Stability, Convergence and Robustness. Dover Publications, Mineola.

Schlittgen, R. & Streitberg, B.H.J. (2001): Zeitreihenanalyse. Oldenbourg Wissenschaftsverlag, Munich.

Scholz-Reiter, B., de Beer, C., Böse, F. & Windt, K. (2007): Evolution in der Logistik – Selbststeuerung logistischer Prozesse. Pages 179–190 in 16. Deutscher Materialfuss-Kongress – Intralogistik bewegt – mehr Effizienz, mehr Produktivität. VDI Verlag, Düsseldorf.

Scholz-Reiter, B., de Beer, C., Freitag, M., Hamann, T., Rekersbrink, H & Tervo, J.T. (2008): Dynamik logistischer System. Pages 109–138 in Beiträge zu einer Theorie der Logistik (P. Nyhuis, Ed.). Springer Verlag, Berlin.

Scholz-Reiter, B., Kolditz, J. & Hildebrandt, T. (2009): Engineering autonomously controlled logistic systems. International Journal of Production Research 47(6): 1449–1468.

Scholz-Reiter, B., Rekersbrink, H. & Freitag, M. (2006): Kooperierende Routingprotokolle zur Selbststeuerung von Transportprozessen. Industrie Management 22: 7–10.

Schreiber, S. & Fay, A. (2011): A reference system for the benchmarking of manufacturing control systems. Pages 1–4 in Proceedings of the IEEE 16th International Conference on Emerging Technologies & Factory Automation (ETFA). IEEE Press.

Schwert, G.W. (1989): Test for Unit Roots: A Monte Carlo Investigation. Journal of Business & Economic Statistics 7(2): 147–159.

Seow, K.T., Dang, N.H. & Lee, D.H. (2010): A collaborative multiagent taxi-dispatch system. IEEE Transactions on Automation Science and Engineering 7(3): 607–616.

Sim, K.M. & Sun, W.H. (2003): Ant colony optimization for routing and load-balancing: survey and new directions. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans 33: 560–572.

Sinriech, D. & Tanchoco, J.M.A. (1993): Solution methods for the mathematical models of single-loop AGV systems. International Journal of Production Research 31(3): 705–725.

Sinriech, D., Tanchoco, J.M.A. & Herer, Y.T. (1996): The segmented bidirectional single-loop topology for material flow systems. IIE Transactions 28: 40–54.

Taghaboni-Dutta, F. & Tanchoco, J.M.A. (1995): Comparison of dynamic routeing techniques for automated guided vehicle system. International Journal of Production Research 33(10): 2653–2669.

Talbot, L. (2003): Design and performance analysis of multistation automated guided vehicle systems. Ph.D, dissertation, Universite Catholique de Louvain.

Tarau, A.N. (2010): Model-Based Control for Postal Automation and Baggage Handling. Ph.D, dissertation, Delft University of Technology.

ten Hompel, M., Kamagaew, A. & Stenzel, J. (2012): Cellular Transport Systems in Facility Logistics. Pages 246–254 in Coordinated Autonomous Systems – Proceedings of the 6th International Scientific Symposium on Logistics 2012 (W. Delfmann, T. Wimmer, Eds.). DVV Media Group, Hamburg.

ten Hompel, M., Libert, S. & Roidl, M. (2010): Erarbeitung von Methoden und Regeln zur Gestaltung agentengestützter, dezentraler Steuerungen für den Einsatz in komplexen Materialflusssystemen. Final report on the research project 15313N

ten Hompel, M., Libert, S. & Sondhof, U. (2006): Dezentrale Steuerung für Materialflusssysteme am Beispiel von Stückgutförder- und Sortieranlagen. Logistics Journal 2006.

ten Hompel, M., Schmidt, T. & Nagel, L. (2007): Materialflusssysteme: Förder- und Lagertechnik. Springer Verlag, Berlin.

ten Hompel, M. & Schmidt, T. (2005): Warehouse Management. Springer Verlag, Berlin.

Vahrenkamp, R. (2007): Logistik: Management und Strategien. Oldenbourg Wissenschaftsverlag, Munich.

van der Meer, J.R. (2000): Operational Control of Internal Transport. Ph.D, dissertation, Erasmus University Rotterdam.

Verband Deutscher Maschinen- und Anlagenbau (VDMA) (Eds.) (1994): VDMA-Einheitsblatt 15276: Datenschnittstellen in Materialflusssystemen. Beuth-Verlag, Berlin.

Vis, I.F.A. (2006): Survey of research in the design and control of automated guided vehicle systems. European Journal of Operational Research 170: 677–709.

Weyns, D. & Holvoet, T. (2008): Architectural design of a situated multiagent system for controlling automatic guided vehicles. International Journal of Agent-Oriented Software Engineering 2(1): 90–128.

Zivot, E. & Wang, J. (2003): Modeling Financial Time Series with S-PLUS. Springer Verlag, Berlin.