Logistics Research (2020) 13:10
DOI_10.23773/2020_10

Research on Decentralized Control Strategies for Automated Vehicle-based In-house Transport Systems – a Survey

T. Schmidt[1], K.-B. Reith[1], N. Klein[1], M. Däumler[2]

Received: 09 July 2019 / Accepted: 19 October 2020 / Published online: 26 November 2020

ABSTRACT

With the application of in-house logistics automated guided vehicle (AGV) systems for transportation three different control problems arise: task assignment, empty vehicle balancing and routing. With an increasing fleet size and especially when considering requirements like flexibility and adaptivity these control problems often become complex to solve using a central controller. The main reasons are the increasing problem size and amount of information. Decentralized control of logistics systems has received a lot of attention during the last years and may help to remedy the shortcomings of central solutions. Decentralized solutions rely on a distributed implementation for decision making and make use of local information.

This survey describes basics as well as general categorizations and reviews existing decentralized control strategies for the mentioned control problems of automated in-house logistics vehicle systems. If existing, decentralized solving approaches for the control strategy problems investigated by the scientific community are listed. Additionally, these control strategies are evaluated to which extend they fulfill the requirements of the decentralized paradigm.

KEYWORDS: automated guided vehicle (AGV) · decentralized control · dispatching · empty vehicle balancing · load-vehicle assignment · routing · task assignment

Thorsten Schmidt[1] (Corresponding Author)
Karl-Benedikt Reith[1]
Nils Klein[1]
Martin Däumler[2]

[1] Technische Universität Dresden
Dresden, GERMANY

[2] Fabmatics GmbH
Dresden, GERMANY

1. INTRODUCTION

1.1. Motivation

In-house logistics deals with logistical challenges within the facilities along supply chains. It employs manual or automated material handling systems for carrying out the core logistical tasks on the operational level. Storage systems, transport systems or sorting systems are examples for the most relevant system types.

Logistics in general and thereby the in-house logistical systems face new challenges which are driven by globalization of company operations, mass customization, shorter product life cycles and consequently lead to rising complexity and dynamics. Depending on the application, logistics systems should have specifications, as follows [see 51, 96]

- flexibility (e.g. modular design)
- (re)configurability & reusability
- high availability
- plug & play configurability (standardization of interfaces and communication)
- scalability & adaptivity
- energy-efficiency and resource-efficiency

Intuitively, manual or semi-automated systems would be the most suitable choice for fulfilling these requirements as they show a high degree of freedom regarding changeability. But usually in-house logistical systems also have to fulfill high throughput requirements and have to cope with high labor costs as well as special product characteristics. Especially companies in industrial nations are faced with increasing health and safety regulations and have to counteract the demographic change with technical, automated solutions [57, 88].

Automated material handling systems however, cannot fully cope with the described requirements [34, 81]. Centralized structures in control systems, especially customer specific hardware configurations and tailored algorithms lead to

- huge efforts for modifications
- increasing test efforts for update procedures
- limited flexibility
- costly hot or warm stand-by systems as the central control system is a single point of failure
- restriction to hardware from the manufacturer

A control unit that relies on global information for decision making requires ongoing status updates of various system units. Depending on the number of units in a system the resulting communication overhead might be a restriction for the system performance. But not only the amount of data and information is challenging. Garey and Johnson [54] and ter Mors [107] point out that the complexity of the control strategy problems (see 1.2) for example in vehicle transport systems make them *NP*-hard. Generally speaking, centrally controlled systems often reach their limits with regard to computing capacity when using real world problem sizes.

Knowing about the weaknesses of state-of-the-art centrally controlled systems and having the increasing challenges of dynamic complexity in logistical systems in mind the concept of "decentralized control" has gained more and more attention in material handling literature during the last years. Decentralized control systems are said to offer a number of advantages compared to centrally controlled systems [45, 89, 105].

Decentralized systems should be easier to implement and configure, due to their modular structure. System changes do not automatically result in high costs. Excellent flexibility, (re)configurability 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.

Generally speaking, in complex systems optimal decision making is hard to achieve. However, decentralized systems try to use local information and rather decentralized decision rules. Due to multiple independent entities interacting with each other and reacting on the system status, it is very complex to predict the overall system behavior that emerges from decentralized systems. As a consequence the performance reached by a decentralized system can hardly be predicted [97]. Optimality is either expected to arise automatically from the local interactions or traded explicitly for the advantages of decentralized control systems which were mentioned above. At best the simplicity of decision making may lead to a higher efficiency than in central systems [91].

The decentralized approaches differ in the level of implementation as we will show later. While some of them seem to be applicable in real-world material handling systems, others still need some research. However, many of the approaches could cope with requirements of future in-house logistical systems. Schreiber [93] pointed out that decentralized control systems have not made the step from research to practical applications in full extent. As a reason Schreiber mentions the qualitative advantages of decentralized systems, which are hard to measure and compare. Up to now, the number of real-world applications is still low. Flämig [49] mentions for example that even newly built vehicle systems are usually centrally controlled and decentralized system are still in research.

However, in the scientific community there is an increasing number of decentralized approaches on controlling in-house vehicle transport systems. We feel that it is therefore necessary to review these decentralized control strategies and categorize them by the vehicle control problems.

On the one hand this survey gives an overview about available decentralized concepts in the field of in-house vehicle transport systems and thereby enhances the development process in other projects. On the other hand, the survey helps to identify questions which have not been answered yet and areas where future research in this field is required.

Similar surveys on decentralized control strategies can be found for example in the sectors of freight transport [102] and distribution logistics [53].

As it is impossible to include all in-house logistical system types in this survey the next section provides a clear definition of our scope. It will also clarify the understanding of decentralized control that we use in this paper.

1.2. Scope of the survey

In this publication we focus on vehicle-based in-house transport systems. Examples for this kind of transport system are carrier-based systems like automated guided vehicle (AGV) systems and overhead hoist transport (OHT) systems.

In the following the scope of this survey is described regarding (a) control strategy problems (b) type of information and implementation (c) considered vehicles & layout restrictions as well as (d) relevant literature.

Control strategy problems

According to Sinriech and Tanchoco [98] the design of vehicle based material handling systems covers the unit load sizing, the layout development, selection of vehicles in type and quantity and the design of an appropriate control system. Within a control system, specific control strategies are responsible for solving a control problem.

For the defined class of automated in-house logistics transport systems three different control strategy problems can be distinguished: (1) load-vehicle assignment (see section 2.1) as a part of dispatching, (2) empty vehicle balancing (see section 2.2), also a part of dispatching and (3) routing (see section 2.3). These control strategy problems will be analyzed in detail in the literature review in section 2.

The assignment of loads and vehicles to each other is the main task of the load-vehicle assignment. In our understanding it is a part of the vehicle dispatching process. The empty vehicle balancing deals with the assignment of idle vehicles to a parking location with the goal of positioning the idle vehicle in a forward-looking manner. If done correctly future response times of the transport system on new arriving transportation tasks are minimized. As will be shown, this control problem is often closely linked to the load-vehicle assignment. Therefore, we consider the empty vehicle balancing as the second part of the vehicle dispatching process.

Generally speaking, the routing is required to define an appropriate route from source (current location) to sink (destination location). "Appropriate" may refer to the shortest route or further criteria as the shortest route is not necessarily the fastest.[1]

Type of information and implementation

The key limitation of scope is the focus on decentralized control strategies. For a clear definition of decentralized control two different aspects have to be distinguished [69].

On the one hand, there is the internal structure and implementation of the control system. Conventional control systems have a hierarchical structure. This means there is one central controller which collects information from the system and is responsible for decision making. In a decentralized implementation the central decision making unit is replaced by several smaller, distributed units that make decisions autonomously. They can be called decentralized or heterarchical units.

On the other hand, the term decentralized can be used to describe the type of information that is used for decision making. It can be distinguished whether a (decentralized) controller uses global system knowledge or locally available information only. In our definition, a decentralized implementation relies on local information.

Nevertheless, it is worth mentioning that there is no strict line (or distinction) between central and decentralized systems with regard to the structure of implementation and the information used. For example, a system can consist of multiple heterarchical units which make almost all necessary decisions autonomously and a single central unit which is only responsible for a minor task. Analogue these entities can rely on local information mainly but still have knowledge about waiting transportation tasks, which is some kind global information. Furthermore, there is no clear definition in literature of what can be regarded as local information. To sum up, there is no strict categorization of systems as either central or decentralized. Instead tendencies towards one or the other paradigm are illustrated in this survey. In the following, this paper uses the term "truly decentralized" for control systems that are

1 In this publication "route" is used as a synonym for "path". A route or a path consists at least of one path element.

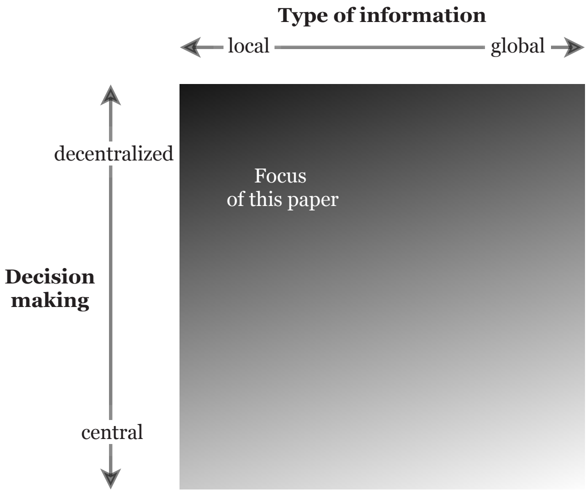

Figure 1: Matrix of decision making and type of information according to [69]. The main focus lies on the decentralized decision making based on local information (black area). As there is no strict distinction partly decentralized approaches (lighter gray area) are touched as well.

implemented in a decentralized way and only use local information for decision making. Despite having a focus on decentralized approaches, in the following literature review there is not a strict limitation to truly decentralized solutions only. A clarification of the scope of regarded systems with different types of implementation and levels of information is visualized in figure 1.

The combination of decision making level and type of information determines the characteristics of the system. Almost any kind of system is possible. However, some of the combinations are not reasonable, like a centrally implemented decision making based on local information only.

Vehicle & layout restrictions

We only consider systems with a homogeneous fleet of vehicles. So, each vehicle has the same technical parameters, like maximum speed and handling time.

The majority of vehicle transport systems in industrial applications are path guided. The representation of the path can be either physical or virtual. Free ranging vehicles are rarely mentioned for big real world applications. Therefore, these systems are not in our scope.

The most important layout components of such an in-house logistics transport system is the path layout itself, which consists of loading and unloading stations, merges and switches. In addition, we consider storage locations, also called "dwell point", "depots" or "home locations", where vehicles can park.

In the regarded context the number of vehicles can usually be seen as the limiting factor. Opposed to vehicle-based systems carrier-based systems as used in baggage handling systems (BHS) at airports usually behave differently due to a higher number of carriers compared to the number of transport requests. The limiting factor in carrier-based conveyor systems is usually the track capacity. Therefore, the focus of this paper does not lie on carrier-based conveyor systems. However, they are rarely mentioned, when there is an approach that in general fits to the scope of this paper and serves as a supplement example.

Relevant literature

In this survey we mainly consider the implementations according to the described scope. Nevertheless, we will make reference to important publications that are out of the scope and to strategies that would not fit exactly in our definition of true decentrality, e.g. because the decentralized decision units still make use of global information to some extent.

The initial impulse for this survey was the dissertation of co-author Klein [69]. This was also the basis for some parts of this work.

2. CONTROL STRATEGY PROBLEMS

Having introduced the exact scope of this paper, this section reviews the three different control strategy problems, namely (1) load-vehicle assignment, (2) empty vehicle balancing, as well as (3) routing with a decentralized-focus (see figure 2).

For each control strategy problem we analyze if decentralized approaches are available and to which extent they fulfill the requirement of using only local information, i.e. if they can be considered to be truly decentralized. Furthermore, the chronological development for popular approaches is also taken into account.

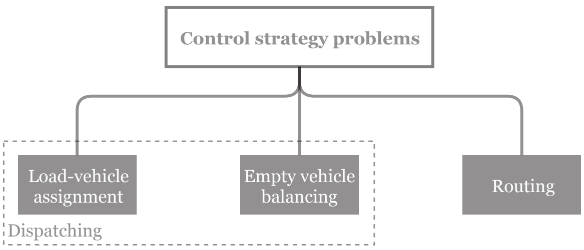

Figure 2: Classification of control strategy problems

Load-vehicle assignment is the challenge of assigning vehicles to new transport requests or vice versa assigning transport requests to vehicles, see section 2.1. The empty vehicle balancing described in section 2.2 is about choosing a vehicle's destination when it has just completed a job and no more tasks are available.

The term "dispatching" is widely used in many contexts, e.g. the control of power systems or production planning. Consequently, the meaning of the term differs and confusion can arise quickly. In the context of in-house vehicle transportation systems we see dispatching as an umbrella term which includes the more specific control problems "load-vehicle assignment" sometimes also known as task assignment, and "empty vehicle balancing". In spite of similarities between the control strategies of load-vehicle assignment and empty vehicle balancing the literature on vehicle-based transport systems distinguishes both questions. While the second one is usually called empty vehicle positioning or empty vehicle balancing, the first one is usually referred to as dispatching. In our opinion the definition of dispatching should include both activities as both deal with a similar question: how should a vehicle behave after it has completed a job. Furthermore, both of these sub-tasks need to be considered subsequently in a real-world implementation.

Once a task is assigned to a certain vehicle (or vice versa), discovering a feasible route and selecting "the best" route from a given source to a given destination is part of the routing process. Furthermore, the subproblem of route execution is responsible for operationally guiding a vehicle and for the avoidance of conflicts. All aspects of routing are described in detail in section 2.3. In order to characterize an approach as decentralized in our understanding, we evaluate the entity of a decision making process and the type of information this entity is using for its decision. It is important to understand that there is not a strict "black and white"-like separation of what can be regarded as central or decentralized. Instead there are many levels between complete centrality and true decentrality, as has already been discussed in section 1.2. This becomes especially evident regarding the information used to make a decision. More information in general and especially non-local information usually leads to a higher grade of centrality and therefore a higher level of complexity. The following static and dynamic information is more or less relevant for the aforementioned vehicle control strategies:

- layout specific information (most of the time static):
  - path layout
  - sinks and sources in the layout
  - parking locations
- vehicle specific information (dynamic, but only one entity):
  - current position
  - velocity, acceleration, deceleration
  - current status – idle or busy
  - other relevant information, like battery level
- system specific information (dynamic, multiple entities, aggregated information):
  - position of other vehicles (traffic jams, closed paths)
  - destination
  - paths of other vehicles

– queues sizes at sources and waiting times
– (set of) open tasks

After this brief introduction the following sections review the mentioned control strategy problems always in the same order: (1) basics, (2) literature review on solving approaches and methods, followed by a (3) summary. The literature review is further grouped by topics or rather solving approaches which in turn are arranged chronologically.

2.1. Load-vehicle assignment

2.1.1. Basics

The first aspect of dispatching is the task assignment. One or multiple vehicles (often also referred to as robots) need to get assigned to multiple locations. The task the vehicles perform can for example be a surveillance task or just a charging process. Khamis et al. [66] provide a general overview of the multi-robot-task-assignment (MRTA). Furthermore the assigned task can also consist of a transportation job. In this case the problem is usually referred to as load-vehicle assignment or transport assignment. One or more loads need to be transferred between two different locations and one or more vehicle are able to perform the transports.

In the load-vehicle assignment the destination of a transport can be taken into consideration, which is especially important when vehicles with higher capacities are used. However, the load-vehicle assignment and the task assignment mainly deal with the similar control problem of creating a feasible procedure to match all queued transport requests or tasks to specific vehicles. The assignment method needs to take the characteristics of the vehicle system into account and usually solves the assignment problem with regard to a certain objective, like minimal response time or maximum throughput for the transportation problem. Due to the similar problem structure between the load-vehicle assignment and the general task assignment both problems will be considered simultaneously in the literature section.

While operating a material flow system there are three different events which can trigger a new dispatching decision and in consequence the assignment process [see 74]:

- Arrival of a new load/task in the system
- a vehicle delivers a load to its destination or finishes its task
- a vehicle reaches its parking location

These triggers reflect two general perspectives on the control problem which Egbelu and Tanchoco [36] define for classifying dispatching rules in general: (a) load or workstation-initiated rules and (b) vehicle-initiated rules. The first trigger is linked to load-initiated or workstation-initiated rules. When a load enters the system – for the first time or after a processing step – it is the loads or its corresponding workstations responsibility to select a vehicle for transportation. In order to find the "best" vehicle for the transport, all available vehicles need to be ranked according to one or multiple criteria. The logic for prioritizing the vehicles is essentially the assignment rule.

The last two triggers on the list – a vehicle finishes a task or reaches a parking location – can be linked to the vehicle-initiated assignment rules. Instead of a load/workstation choosing an idle vehicle, it is the vehicles responsibility to apply a logic in order to choose its next load or task. For a further comparison of the two perspectives see also [70].

Though the two perspectives seem to be based on a different approach, both perspectives are usually combined in real-world applications. In case an arriving load can not find an idle vehicle at once (load initiated rule), it should not continuously be searching for vehicles. These ongoing calculations could slow down the system. However, the simpler approach is to keep the load as an open task and offer it to the next vehicle that becomes idle (vehicle initiated rule).

Various centralized approaches for the load-vehicle and task assignment have been developed over the past years. There is a whole range from simple to complex rules. Le-Anh and de Koster [8] and Vis [108] provide excellent overviews about design and control of automated guided vehicle (AGV) systems, including sections about vehicle dispatching rules. Fazlollahtabar and Saidi-Mehrabad [46] provide an overview of various assignment methods, which they call scheduling.

These approaches all have – by definition – a central instance which performs the assignment. Generally this central controller has access to global information, i.e. the central instance has a complete image of the system status. However, depending on the complexity of the approach, the problem size and the performance that should be reached different quantities of information are taken into account.

According to Le Anh [74] central assignment strategies can be classified according to (a) the number of attributes taken into consideration for a decision (single-attribute rules or multi-attribute rules), (b) the existence of a look-ahead period (dynamic rules), (c) the possibility for job reassignment and (d) the possibility of a sequential decision procedure (hierarchical rules). Due to the complexity of the assignment problem, central control strategies are usually heuristics. Instead of a global optimum, heuristics make a reasonable decision in a short time. In order to find a suitable solution with increasing problem size, limits in the amount of necessary communication and computing times are reached as problems related to the area of load-vehicle assignment are often *NP*-hard as shown by Garey and Johnson [54]. This becomes especially evident in a so-called on-line assignment with an ongoing arrival of new loads or tasks. This often leads to less complex approaches or calculations concerning a lower amount of information, resulting in a lower system performance.

Opposed to that, a decentralized load-vehicle or task assignment is able to remedy some shortcomings of centralized approaches, especially the overall complexity in huge systems. Decentralized approaches are characterized through (a) a decentralized implementation for decision making, i.e. each vehicle or each load calculates independently which task to perform next and (b) the decisions are made based on local (or at least less global) information primarily. Generally speaking, decentralized dispatching and thus a decentralized assignment is discussed less frequently in literature than central dispatching. Additionally, the literature discusses decentralized dispatching usually in very simple systems which can be categorized in three different vehicle-based transport system types due to their path layout: (1) single-loop systems, (2) tandem systems and (3) conventional systems[2] according to Le Anh [74]

In the following we will limit our attention to assignment strategies that work without pre-arrival information and that are commonly used in in-house logistics environments. These strategies are used because of their simplicity and their easy adaptability to dynamic and stochastic situations. But they are mostly heuristics. Nevertheless, there are also authors that evaluate, whether assignment rules can be replaced by dynamic scheduling algorithms, which in our definition include pre-arrival information. Le-Anh et al. [7] provide a good introduction to these techniques that are not subject of this survey.

A literature review of existing decentralized approaches for the task and load-vehicle assignment under consideration of the information types used and the level of decision making can be found in the following section.

2.1.2. Literature review on solving approaches and methods

The major research activities on the topic of load-vehicle and task assignment started in the 1980s [see e.g. 79] and many of the fundamental classifications which are still valid have been developed during the last two decades of the 20th century. Starting with manufacturing systems and warehouses, attention has especially turned to container port terminals. In recent years the topic of load-vehicle assignment gains more attention due to an increasing amount of AGVs in intralogistics systems. Another important area of research is overhead hoist transport (OHT) systems in semiconductor wafer fabrication [see e.g. 71].

Two major approaches could be identified in literature that where labeled as decentralized: (1) layout transformation to enhance the performance of decentralized control strategies and (2) multi-agent systems. Whereas the layout transformation were the first approaches for a decentralized assignment the vast majority of nowadays approaches are somehow related to the idea of agent-based systems.

Layout transformation

The first approaches which applied a decentralized load-vehicle or task assignment focused on single-loop layouts.

Bartholdi and Platzman [12] analyzed a single-loop system and evaluated the performance of a first encountered first served (FEFS) rule for this scenario. Using this rule each vehicle circulates in a given path loop. Whenever a load is encountered by a vehicle at its current location and the vehicle status is idle the load gets picked up and gets delivered as soon as the vehicle reaches the load's destination location. Only local information is used for this approach and the rule is performed by each vehicle individually. Therefore, this approach can be regarded as truly decentralized. The FEFS rule can be regarded as a greedy rule from a vehicle's individual view. Opposed to that, the first come first serve (FCFS) rule is a greedy rule which takes information about the arriving time of multiple loads into account. The authors were able to derive an expected performance level analytically and to verify it with a simulation study. As a result, the described FEFS rule outperformed other simple (but less decentralized) rules like FCFS or longest queue first (LQF).

Inspired by these results several authors tried to develop new approaches during the following years. The idea was to develop extremely simple layouts or to transform conventional layouts into combinations of simple, i.e. single-loop layouts. The resulting layout is far easier to control and should achieve the same performance as the more complex conventional layout. Sinriech and Tanchoco [98] 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 served by only one bidirectional vehicle [99]. Although the authors do not explicitly state this, they use some kind of sequential dispatching which is similar to the FEFS rule. Bozer and Srinivasan [15] and Ross et al. [85] also follow a simplification approach when they introduce their idea of tandem layouts.

Although all those authors spend a lot of effort on measuring the performance and comparing their systems to the conventional version, it needs to be

2 A single-loop systems consist of one guide path loop where several loading and unloading stations are located. One or more vehicles travel in this loop which leads to simple traffic control and dispatching requirements. A tandem systems can be regarded as multiple single-loop systems. Only one vehicle is used in each loop and loads can be passed via transfer points between the single-loops. Conventional systems can refer to virtually any real-world transport system. Compared to the other 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 congestions.

mentioned that the considered conventional layouts often have a very simple structure themselves with only a few vehicles used in each of the systems.

Multi-agent systems (MAS)

Besides the mentioned decentralized approaches which are based on layout transformation the concepts of agents and multi-agent systems started to influence the control literature in the late 1990s. Generally speaking, a multi-agent systems (MAS) consists of two or more agents who interact with each other to achieve a collective goal. Usually this interaction is reached using either direct communication or indirect communication via storing information somewhere. With the agents having the ability to cooperate and to go for some individual or even global goals a reasonable behavior of the entire system should arise from their interaction. General advantages are among others the lack of a single point of failure and the "plug and play"-principle. Further advantages and general features can be found in [92]. For a more detailed study on collective behavior of mobile agents see [117].

MAS usually offer a flexible behavior. As each agent interferes with the (changing) environment independently there is no need for a global strategy tackling each special case. Especially in complex systems it is almost impossible to reach such a flexible behavior with a central control unit. The main disadvantage of MAS is that the exact system behavior is almost unpredictable and usually not optimal. Examples for agents in our context are vehicles, source and destination nodes, or even path segments. According to Weyns et al. [116] "Applying a multi-agent system opens perspectives to improve flexibility and openness of the system: the AGVs can adapt themselves to the current situation in their vicinity, order assignment is dynamic, the system can deal autonomously with AGVs leaving and re-entering the system." Several authors developed agent-based systems in order to exploit the advantages of the decentralized paradigm. Opposed to the layout transformation approaches, agent-based strategies for the load-vehicle and task assignment were also applied on more complex, conventional path layouts.

Generally speaking, many decentralized approaches based on the concept of multi-agent-systems make use of so-called auction algorithms. The basic idea is that the assignment between vehicles and loads/stations are determined using an auction based on certain criteria, e.g. the nearest vehicle gets a transportation task etc. Many of these auction procedures are based on the so-called contract net (CNET) protocol. CNET is an important principle regarding communication of entities in a distributed problem for cooperative task execution and was originally presented by Smith [100]. This protocol describes a negotiation sequence that serves as a basis for all auction-based control strategies, i.e. the protocol regulates how a task is announced in a network, how and when a vehicle can make its offer for the task, how a vehicle finally receives the permission for the task, etc. The CNET protocol was extended with various ideas, f.e. the possibility for vehicles to return previously assigned tasks and in consequence to reassign tasks. Generally speaking, auction-based approaches are robust to inconsistencies between multiple agents regarding the awareness of the system status [22]. Therefore auctions are especially suitable for distributed and decentralized approaches. In the following decentralized agent-based approaches with relation to auctions and CNET are presented.

Fay and Fischer [44] develop a control system for destination-coded vehicles in a baggage handling application. They propose a multi-agent load-vehicle assignment based on the CNET protocol. Loading stations offer their waiting loads on a virtual market. A vehicle can evaluate these offers based on its current 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. Fay and 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 and Holvoet [115] and Weyns et al. [116] also makes use of the multi-agent perspective. The aim of the authors is to test the feasibility of decentralized control systems for AGVs. They refer to the load-vehicle assignment as "transport assignment" and develop two different rules which they call FiTA and DynCNET which are discussed in the following. 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 field of a transport agents attracts idle vehicles whereas AGVs repulse themselves. The vehicles combine the effects of the fields they receive to calculate some kind of field gradient which they follow along predefined paths. The DynCNET is an adaption of the CNET Protocol, where transport agents and AGV agents both have a certain radius in which they search for each other. The transport priority and the distance between vehicle and load are used as criteria for providing proposals. Compared to the standard CNET protocola significant modification has been made: The vehicles are allowed to switch tasks after the initial assignment. Weyns and Holvoet use simulation experiments to analyze the performance of the described rules. In a real-world layout with 56 loading and 50 unloading stations they use 14 AGVs to show that their two approaches perform better than the original CNET protocol regarding to the average waiting times. Both rules of [115] have a distributed implementation of control, hence using the FiTa rule some central control entities are necessary, f.e. a controller, which calculates the filed in every point of the map. Obviously, both rules do not rely on local information only. Weyns and Holvoet calculate the total necessary communication load for their proposed rules, which is about twice as high as the communication level for a more decentralized dispatching strategy like the CNET protocol.

In Choi et al. [22] two decentralized approaches for the task assignment including a communication constraint are presented: the consensus-based auction algorithm, where a single task gets assigned independently to an agent and the consensus-based bundle algorithm, where a sequence of tasks are combined and assigned to an agent at once. Both algorithms rely on two phases. In the auction phase each vehicle can bid on specific tasks. In the consensus phase the vehicles receive information about the results of the auction process via a list of winning bids that gets passed between neighboring vehicles. As a consequence the same situational awareness of the vehicle agents is reached even without a central entity which monitors the system status.

Schwarz et al. [95] also use a multi-agent system for a load-vehicle assignment via a bidding procedure based on the Foundation for Intelligent Physical Agents (FIPA) Contract Net Interaction Protocol [3], a variation of the original CNET protocol. Vehicles are able to bid for orders put out by the stations. An offer is calculated under consideration of the delivery time of the current job and the future position. A vehicle can bid in multiple auctions but can only accept a single job.

Giordani et al. [55] present a two-level multi-agent system framework. In the first level the number of necessary robots for a given number of tasks is calculated based on an iterative auction based negotiation algorithm. In the second level tasks are assigned to certain robots in each time period. The so-called task allocation problem is solved using a distributed version of the Hungarian Method[4]. The decisions are made by agents and the communication constraints only allow communication between neighboring agents. With the help of a simulation study the results of this decentralized approach are compared to a centralized approach. Whereas the presented decentralized solution tends to higher costs due to a worse utilization of resources, it also results in a more robust solution.

Another decentralized approach for the load-vehicle assignment based on the CNET protocol is presented by [78]. Machines offer tasks to various AGVs which calculate costs for the transport considering the distance to the machine, the current battery level and even their physical suitability for the given task. A machine is able to contract two vehicles for the same task. The secondary AGV will replace the first vehicle in the event of a failure. Cloud communication is used for sending and receiving information.

In addition to the described approaches, various further examples for the usage of bidding algorithms in a decentralized load-vehicle assignment are given in literature [see e.g. 25, 37, 42, 43, 94].

All in all, it can be stated that auction-based approaches are only partly decentralized. As each agent is doing its calculation independently, the described approaches can be regarded as a usually completely decentralized implementation of the decision making process. However, agents do usually have access to non-local information: vehicles need to know the status of multiple workstations or workstations receive offers from the vehicles. As a consequence in most auction based approaches information is exchanged and aggregated in a widely manner. Only few publications combine a communication constraint with auction-based approaches.

Having described agent-based assignment approaches with relation to auctions and CNET in the following further agent-based approaches for the load-vehicle and the task assignment are presented.

In 2002 Berman and Edan [13] 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 onto 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. No central controller (or instance) is implemented. Instead, each AGV collects information from all workstations. Then, it calculates a decision according to a described dispatching rule and informs the workstation. In combination with routing rules, a conceptual test environment with up to two vehicles was developed. Though having a decentralized decision making and no central unit which aggregates all the system information, each AGV collects the status from multiple workstations and therefore uses non-local information. Consequently, this approach can only be seen as partly decentralized.

Arsie and Frazzoli [10] and Arsie et al. [9] present a no communication algorithm for a problem similar to a task assignment: mobile agents have to visit target points that are generated in a stochastic manner. The authors see their main contribution in a so-called motion coordination strategy that does not rely on communication between the different agents. Nonetheless, optimality is reached under certain conditions. Even though a higher level of communication will not decrease the performance of a system, the authors show that a higher level is not always necessary. They propose the following algorithm: each agent independently calculates its next action and always visits the target nearest to his current

3 For more information about the FIPA Contract Net Interaction Protocol see their specification [48].

4 The Hungarian Method is an optimization algorithm developed by Kuhn in 1955 that solves the assignment problem [72].

position. If there is no target, the agent will move to a position that minimizes the distance to already serviced target points. The authors state that under a so-called light load condition[5] their algorithm is at least as good as other so far published decentralized algorithms and (almost) optimal. Even though there is no communication between the agents and the implementation is completely decentralized, there is still an aggregation of information as each mobile agent knows about open tasks and the locations that need to be served.

In 2009 Lee et al. [76] transfer ideas from a completely different area of research to load-vehicle assignment in port container terminals. Their decentralized algorithm is inspired by T cells[6] that are able to explore their environment and show an immune response to invading antigens. The algorithm aims to minimize the vehicle dwell time and is tested in a simulation model with 12 vehicles, 15 containers and 9 cranes. The algorithm requires communication between the different system entities and thereby leads to an aggregation of information. In addition, the vehicles need to perceive jobs within a certain radius. It remains unclear how this requirement could be fulfilled technologically.

Based on the idea of multi-agent systems, Klein [69] develops four different strategies for a decentralized dispatching which includes load-vehicle assignment and empty vehicle balancing, as explained below (see also section 2.2):

1. Random dispatching rule:
   Vehicles do not have any information about the system. Hence the vehicles start traveling in a random manner until a location with a new transportation task is reached.
2. Static destination dispatching rule:
   Using static destination rule each vehicle has an initial source. Whenever a vehicle completes a transportation task it returns to this home location and only claims the next load there.
3. Forecast dispatching rule:
   In the forecast dispatching sinks aggregate local information over time. In a data collection process the sink keeps the information about a load's source location, whenever a vehicle delivers said load at the sink. Over multiple periods the sink is able to forecast a rough number of transports at a certain source and can therefore forward vehicles accordingly. As no global information is aggregated this forecast must not be regarded as exact calculation, but more like a tendency.
4. Feedback-based dispatching rule:
   The feedback-based dispatching is based on the logic of ant algorithms. At each switch in the system a local probability table tells a vehicle where to go next in the meaning of forwarding. After a vehicle has reached a source and started the next transportation task, it gives feedback (its waiting time) to all switches it passed before. The switches adjust their probability tables accordingly.

In contrast to most other publications Klein lays a special effort on using mostly local information when designing the load-vehicle assignment rules. In our understanding the first three strategies can be seen as truly decentralized, whereas the forth strategy is close to being truly decentralized. All strategies were tested in different layout scenarios (maximum of 18 sources and 18 sinks) with different numbers of vehicles (maximum of 2500) and different transportation data sets. In most of the cases, a central benchmark strategy performs better than the proposed decentralized strategies. Only in small layouts with a high demand variability decentralized strategies were able to outperform the central dispatching strategy. Comparing the decentralized strategies Klein stated that apart from the random dispatching rule no strategy is completely dominating the others in all relevant KPIs especially when taking many vehicles and complex layouts into account.

Ayanian et al. [11] propose an approach based on a dynamic task reassignment as a decentralized solution. Based on an initial random task assignment vehicles within a certain communication range can switch their tasks. Depending on the size of this range, the nature of the problem becomes more or less (de-)centralized. Similar approaches are proposed by Caraballo et al. [17] and Fanti et al. [38]. Caraballo et al. [17] consider a divisible and parallelizable task that has to be accomplished by a team of robotic agents (aerial robots) in a decentralized manner. Within a certain distance (block), information between the agents are shared and tasks are re-allocated. Caraballo et al. identify the block size as an important parameter for the overall performance. Fanti et al. [38] use a so-called "gossip algorithm" under communication constraints. Starting with an unfeasible solution, the algorithm finds a solution for the task assignment problem via communication between the agents. Enhancements are published in [41].

Fanti et al. propose an algorithm for assigning electrical vehicles to charging stations. Each charging station can be seen as an agent which solvesa local integer linear problem considering the charging costs and distance to the vehicles. Therefore, the implementation of the approach can be regarded as decentralized. The stations share and synchronize their individual solution with neighboring charging stations, which leads to an aggregation of information. In an iterative manner the distributed approach minimizes the total time needed for the charging process.

A similar approach is presented by Fanti et al. [39]. Instead of stations assigning vehicles, as seen in Fanti et al. [40], AGVs choose transportation jobs by solving a local integer linear programming problem.

5 a small target generation rate
6 T cell is the short form for T lymphocyte – a subtype of white blood cells.

Vehicles are able to communicate with other vehicles within a certain zone. Following a communication and negotiation protocol, each task gets assigned to an AGV. The vehicles have information about the layout and open tasks including the task position. Therefore, the algorithm uses non-local information in a decentralized implementation. Based on the mentioned assignment problem, Fanti et al. [39] propose a solution for coordinating the vehicles, which we will mention in chapter 2.3.

2.1.3. Summary

Table 1: Literature overview ordered by topics for decentralized load-vehicle and task assignment

Topic	Corresponding literature
Layout transformation	[12, 15, 85, 98, 99]
Multi-agent systems	[9–11, 13, 17, 22, 25, 37–44, 55, 69, 76, 78, 94, 95, 115, 116]

An overview summarizing the literature discussed in this section can be seen in table 1. The vast majority of approaches in the field of decentralized load-vehicle and task assignment are based on multi-agent systems. Within these multi-agent systems many publications are somehow related to auction-based approaches. Noticeable is the low number of truly decentralized solutions, especially with regard to the local information constraint. Some publications that were labeled as decentralized do not have any information constraint at all. Other publications do restrict the communication between different agents, e.g. only neighboring agents are allowed to share information. The range of what is considered as neighboring is discussed less frequently. Nevertheless, most of the approaches make use of a decentralized implementation, i.e. multiple entities calculate solutions independently.

When taking the nature of an assignment procedure into account, the reason for this lack of truly decentralized policies with regard to the use of local information becomes clearer. Allowing the vehicles and workstations to share their current status is a manageable amount of communication, but can result in a high benefit. Analogous, in most publications the status communication between vehicles and workstations or between multiple vehicles is the violation of our definition of true decentrality.

2.2. Empty vehicle balancing

2.2.1. Basics

In the following we will discuss the second part of dispatching: the control of empty vehicles. Terms, like “empty vehicle positioning” or “empty vehicle balancing” are used frequently. The problem tackles the simple question of what to do with an idle vehicle if there are currently no open jobs in the system. In order to minimize the empty vehicle travel time and the future system response time for new pickups, the major objective of vehicle positioning is to find parking locations – also called home locations or dwell points. The empty vehicle positioning can have a strong impact on the overall system performance, especially with a high load profile variability.

Hu and Egbelu [61] and Le Anh [74] identified four approaches which are proposed in automated guided vehicle (AGV) literature:

- Central zone positioning:
  Idle vehicles are always sent to one central zone.
- Point of release positioning:
  Idle vehicles stay at the location, where they completed their last task.
- Distributed positioning:
  This rule is generally similar to the central zone positioning rule but multiple zones are available where the vehicles can be sent to.
- Circulatory loop positioning:
  Instead of parking the vehicles are circulating in the system. One or multiple loops are designed for empty vehicle circulation.

It is obvious that each of the mentioned approaches can be executed centralized or decentralized with either local or global information and the feasibility is highly linked to the path layout. During the layout design process a decision about the empty vehicle positioning policy has to be made. This decision has to be operationalized with an appropriate layout afterwards.

Hu and Egbelu [61] summarize the first three approaches as systems with storage locations. They state that most authors who follow these approaches assume some kind of loop sidings for the home locations. Loop sidings are separated areas near the transportation lanes, where idle vehicles do not delay other vehicles. In contrast to the approaches which use some kind of storage locations, the last approach stands for all control strategies that use specific loops in the system to store empty vehicles.

The first approach can lead to considerably long system response times. Therefore, it is only an option in (a) smaller layouts where the central parking zone is close to the loading stations (b) in systems with enough pre-arrival information (c) in systems where this long response time is no problem, e.g. with a very powerful vehicle system compared to the number of jobs. Parking at the point of release – second approach – can lead to congestion in many cases, if no additional parking zones are integrated in the layout. However, these first two idle vehicle parking concepts do not require complex dispatching processes, as the control rules are straight-forward. One could argue whether the empty vehicle positioning is truly decentralized or not even a dispatching process is required because there is in fact no decision to be made. Idle vehicles are simply sent to a predefined storage location or just wait at their location. Due to their simplicity the first two of the four empty vehicles positioning approaches are hardly discussed in the scientific community.

Looking at the third approach, it becomes clear where the term "empty vehicle balancing" comes from, as vehicles have to be balanced between different zones. This also raises the question of how many vehicles should be in a zone. A decision rule for choosing a parking zone needs to be implemented. The latter approach presented – circulatory loop positioning – is also a simple strategy which enables short response times. The disadvantages can be a higher energy consumption, a danger of blocking other vehicles and a higher risk of congestion.

2.2.2. Literature review on solving approaches and methods

Considering the scope of this paper and especially systems with many vehicles only the third approach is of particular importance. Two relevant sub tasks for the empty vehicle balancing were identified in the literature. On the one hand there is a planning aspect, where optimal dwell point locations have to be selected in an in-house logistics transport system containing several storage locations. On the other hand there is the challenge of distributing a single vehicle to a certain parking position, which can be regarded as a controlling aspect.

In a large part of the literature considered, there is no explanation on which level decisions have been made and how the information is available. Depending on the approach, planning aspects that are influenced by the path layout and vice versa also play a major role, which is pointed out in the following. Similar to what has been said about load-vehicle assignment, many of the publications on vehicle-based systems only consider simple problems.

Planning aspects

Egbelu [35] was among the first authors that consider empty vehicle positioning in a single-loop layout. The single-loop layout is transferred into an equivalent circular layout. Afterwards algorithms for finding optimal dwell point locations are derived in four different setups: single vehicle with unidirectional traffic; multiple vehicles with unidirectional traffic; single vehicle with bidirectional traffic; multiple vehicles with bidirectional traffic. The overall objective is the minimization of maximum response times. Kim and Kim [68] follow the same conceptual approach and analyze a single-loop layout. They use Markov chains to decide about the optimal location for a single central home location. The analytical procedures are valid for a static environment in an unidirectional loop. Chang and Egbelu [19] deviate from the assumption of a static environment. Although they stick to a single-loop layout, they provide a method for determining the home location that minimizes the mean response time for dynamic workloads. Gademann and van de Velde [52] analyze different single-loop systems with variable numbers of vehicles regarding the complexity. They show that the optimal home location problem is much harder to solve for bidirectional loops than for unidirectional loops. Lee and Ventura [75] propose an analytical solution based on a dynamic programming model that minimizes the mean response time. They divide the systems into subsets of stations. Each of the subsets is served by one of the vehicles in uniand bidirectional loops.

Controlling aspect

In contrast to all authors which have been mentioned so far, Hu and Egbelu [61] are the first to consider a conventional layout. For an unidirectional guide path layout the storage locations that minimize the maximum response time or the mean response time are derived in two different models. The performance of an exact and a heuristic approach are compared. In addition to defining the storage locations, the authors provide a procedure for assigning vehicles dynamically to these locations, which is an aspect of controlling. They also try to make less restrictive assumptions about the demand patterns. Bruno et al. [16] provide a similar approach that combines the determination of optimal dwell point locations with a vehicle assignment procedure.

Hallenborg [58] provides a discussion on empty vehicle control in systems with many vehicles. 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 that focuses on distributed vehicle positioning. Each vehicle is assigned to a storage location once it has completed a job. It remains unclear if this is the only dispatching decision or if the load-vehicle assignment is also involved. The travel time to a specific storage location and the current "fill level" are considered for the vehicle dispatching. The comparison of the storage locations requires the usage of global information, like the relative buffer level. The same is true for the determination of the dynamic travel time.

Le-Anh and de Koster [6] make another attempt to analyze dispatching rules for systems with many vehicles. The main assumption for their 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. Therefore, Le-Anh and de Koster follow the fourth of the general approaches which were introduced above. 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 self-developed dispatching rules, they include the best rules which Talbot [104] found in his research. The dispatching rules are compared in two rather simple layouts. One of the layouts contains two loading and two unloading stations. The other layout contains four 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. Using different balanced and unbalanced load scenarios the following dispatching rules are compared: 1. Modified shortest-travel-distance-first rule, 2. entrance control (EC) dispatching rule, 3. multi-attribute dispatching rule (Multi-Att) and 4. modified multi-attribute dispatching rule (Multi-Mod). 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 which is used as a comparative value. 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 this survey, it needs to be pointed out 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 decentralized 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.

In our opinion, empty vehicle balancing as well as vehicle task assignment can be seen as a part of the dispatching procedure, as we pointed out in the introduction to chapter 2 (see also fig. 2). Both are closely linked and influence each other. The approach in the field of multi-agent systems developed by Weyns and Holvoet serves as an example, see section 2.1.1. AGVs driving through the system repulse or attract each other depending on the state (i.e. idle) and pending tasks, see also Weyns et al. [116]. This leads to an automated and implicit distribution of idle AGVs over the system, whereas no dedicated empty vehicle balancing is needed anymore.

A decentralized strategy similar to the described option of circulatory loop positioning is proposed by Klein [69]. The idea is based on the decentralized assumption that a vehicle has no information about the system status, e.g. waiting loads, as it gets idle. In consequence, the empty vehicle drives randomly through the system network. Once it reaches a certain source, the vehicle gets access to the local information of the source, e.g. current transport requests. This strategy shows the close linkage between task assignment and empty vehicle balancing, which we both see as a part of the vehicle dispatching process, as mentioned before. Compared to other dispatching rules, Klein [69] found this strategy to perform worst. Nevertheless, in systems with a huge amount of vehicles, compared to the number of transportation tasks, this strategy can keep the communication and calculation effort to a minimum.

Due to capital-intensive tools in semiconductor industry, the automated material handling system (AMHS) plays an important role. State-of-the-art overhead hoist transport (OHT) systems are designed as unified rail systems, whereby mostly a distinction is made between intrabay and interbay areas. Intrabay means an arrangement of similar tools and interbay is the connection of intrabays. A semiconductor wafer fabrication plant (FAB) can consist of up to hundreds of tools, which are connected by several kilometers of rail tracks including various u-turns and n-shunts [see i.e. 86]. So usually the layout complexity can be considered as very high. A common approach to balance empty vehicles is the use of so called "watermarks", which constrain to number of empty vehicles in areas [see i.e. 67, 86]. Wertz et al. [114] present two approaches for improving delivery times, especially waiting times. First, they modify watermark settings to values which vary dynamically according to the transport frequency in a bay. Second they shift dwell point locations so that other vehicles are almost not obstructed. A simulation was used to evaluate the developed approaches. Kiba et al. [67] show the influence of the low watermark in a certain bay on the average global delivery time and the average global retrieving time with a detailed simulation of a FAB. Chaabane et al. [18] used a discrete event simulation of a wafer fabrication plant to study the influence of low water marks and high water marks on a minimum service policy in the meaning of a minimum number of empty vehicles to serve transport requests quickly. Schmaler et al. [86] present an approach for balancing empty vehicles based on forecast information. They compare three scenarios – no-forecast, limited-forecast and complete forecast – of a FAB with about 300 vehicles in order to show the potential of the approach. One of the goals was also to minimize empty vehicle balancing moves [see also 87].

All in all, the empty vehicle balancing approaches applied in FABs are using a mixture of the distributed positioning approach and the circulatory loop positioning approach. Many decision are made by a mid-level controller which is responsible for an area. Once a vehicle is pushed from another vehicle, it will take another parking position in the same area as a result of the watermark setting which is some kind of local information and decentralized decision making. This push-out of empty vehicles can also be seen as decentralized decision making under local information but the destination is defined with global knowledge about the bays.

Excursus: Sharing systems

The literature on sharing systems – bike sharing or car sharing – has grown considerably in recent years [see e.g. 26, 47]. Depending on the type of system, parallels to vehicle-based transport systems can be made. One of the most popular examples is the rebalancing problem of bikes between stations [see e.g. 20, 84] which is some kind of the distributed positioning rule proposed in 2.2.1. However, there are some significant differences, like cyclically recurring demand patterns pointed out by Vogel et al. [109] or the fact that the bikes cannot move by themselves. Rentals at a certain time or stations can be promoted or avoided by user incentives [5] and specific times of the day – e.g. nocturnal hours – could be used for balancing between all stations. Furthermore, Legros [77] defines the objective in balancing the stations in the meaning of predicting overfull or empty stations which differs to vehicle-based transport systems.

The user based balancing can be seen as a decision making on two levels. On the upper level the rental system operator makes a global decision with full information how to balance the bikes. On the lower level a user takes a ride, which could be a promoted balancing ride but it remains unclear if the user acts the way intended.

2.2.3. Summary

As mentioned above, there is hardly any reference to decentralized empty vehicle dispatching in literature on vehicle-based in-house transport systems. However, three of the four types of different control strategies for the empty vehicle balancing can be seen as decentralized without further effort. With central zone positioning only the information about the layout and the coordinates of the parking zone is necessary. With point of release positioning and circulatory loop positioning, no further central information is needed. The distributed positioning can be implemented either as a central or decentralized version. An overview of the literature in this section can be seen in table 2. After the dispatching strategies, the next section reviews the existing theoretical background for routing problems.

Table 2: Literature overview ordered by topics for empty vehicle balancing

Topic	Corresponding literature
Planing aspect	
AGV	[19, 35, 52, 68, 75]
Controlling aspect	
AGV	[6, 16, 61, 104, 116]
Baggage Handling	[58]
OHT-Systems	[18, 67, 86, 87, 114]
Sharing-Systems	[5, 20, 26, 47, 77, 84, 109]
Other	[69]

2.3. Routing

2.3.1. Basics

Generally speaking, an essential requirement for a routing procedure is the existence of a destination for a vehicle. As the destination is the result of the dispatching procedure the routing process usually succeeds the dispatching process. However, in more advanced controlling processes the information about expected travel times can also be an input factor to the dispatching decision leading to an iterative problem solving process. In consequence, despite having a tendency the order of dispatching and routing is not strictly predefined.

Before we provide a general classification scheme for routing strategies it is necessary to clarify the relationship between the terms routing and scheduling. On the one hand it is a general consensus that routing is about finding a route between a given source and destination. On the other hand the term scheduling is often defined differently and especially the distinction to routing often remains unclear. Some authors consider routing to be responsible for finding routes that are free of congestion, conflicts and deadlocks [see e.g. 83]. Other authors see these requirements as a part of scheduling [see e.g. 103].

Generally, both views are understandable. Due to the technological complexity of the vehicle on the one side and the guidance technology on the other side, controlling a vehicle system on the operational level is not an easy task. Extracting as many decisions from the operating phase to a preceded planning seems to be a reasonable approach for preventing undesirable effects (congestions, deadlocks, etc.). The characteristics and especially the complexity of the analyzed systems seem to be the driving force for either requiring routing to fulfill additional requirements or incorporating it into the scheduling function. The latter view is applicable in systems where all transportation requests are known before the system starts its operation. Then optimization techniques can be used to find the best routes without any interference and reach the highest over-all system performance. In this case the optimization problem can be categorized as a pick-up and delivery problem with time window constraints. In some cases dynamic scheduling approaches with rolling planning horizons might even be able to achieve a continuous optimization and adjustment to changing demands [see 74]. But this can only be accomplished for systems of limited complexity. In many systems scheduling is impossible as no or only limited pre-arrival information is available. In this case dispatching strategies have to be used for the task assignment (see sec. 2.1) or empty vehicle balancing (see sec. 2.2) and the extended understanding of routing as favored by Qiu et al. [83] becomes relevant.

In this paper the term routing is used to describe control procedures which are responsible for the following tasks (see also fig. 3):

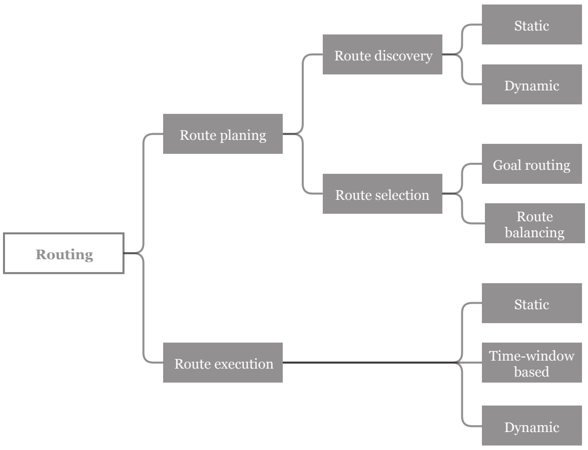

Figure 3: Classification of routing subtasks [compare 73, 83]

- Route planning:
  - Route discovery:
    Finding one or more feasible routes from a source to a desired destination [see 73]
  - Route selection:
    Selecting a specific route, if more than one route is feasible for a source-sink combination [see 73]
- Route execution:
  - Operationally guiding the vehicle from source to destination, which means especially the avoidance of conflicts with other vehicle

A further classification and description of the mentioned routing sub tasks (discovery, selection, execution) is presented in the following. Various (decentralized) approaches from different publications that can be fitted into this classification are discussed in the literature section. It is important to understand that this classification must not be seen as a sequence of necessary tasks to reach a decent routing decision. Not each sub problem individually needs to be targeted by an approach. For example there are approaches which only rely on a route execution without any advanced route planning.

Similar to the control problems of vehicle dispatching and vehicle routing the sub problem of route planning and route execution can be executed as iterative processes. In more advanced routing approaches the current system status as processed by the route execution can be taken into consideration for the next phase of route planning or another iteration of route planning is made when the route execution discovers a conflict. This leads to a more complicated iterative process.

According to ter Mors [107] a sophisticated route planning should ensure a less difficult route execution. Furthermore, ter Mors proves that the coordinated movement of multiple agents in a given map is a *NP*-hard problem.

Route planning

After these introductory comments various general approaches for the sub problems of route planning are presented. In general, the route planning is done before the vehicle starts moving and includes the tasks of (a) discovering routes from a given source to a given destination and (b) selecting a route if more than one route is possible.

Route discovery as a part of route planning

A general classification scheme (see fig. 4) developed by Lau and Woo [73] fits our definition of the term route discovery. Hence, the following paragraphs mainly show their findings.

The major distinction of route discovery strategies is in static and dynamic approaches. This is dependent on the opportunity to update a choice for a calculated path between a specific source and destination. A static route planning calculates the best route only once. Without taking current congestions into account, the calculated route is always used for a specific source-destination combination. To calculate the route in the first place, the route discovery algorithm needs a criterion for determining the best route and for comparing multiple feasible routes. In static approaches usually the shortest distance or time is used for this purpose. Dijsktra algorithms or Floyd-Warshall algorithm are two of the most famous algorithms for computing the shortest paths [33, 50, 113]. In addition to calculating a complete a priori routing table for each source-sink relationship, static route discovery can also be realized based on simple static rules. Similar to a calculated table, these static routing rules will always yield the same result for a given source-destination relation, without taking the current system status into account.

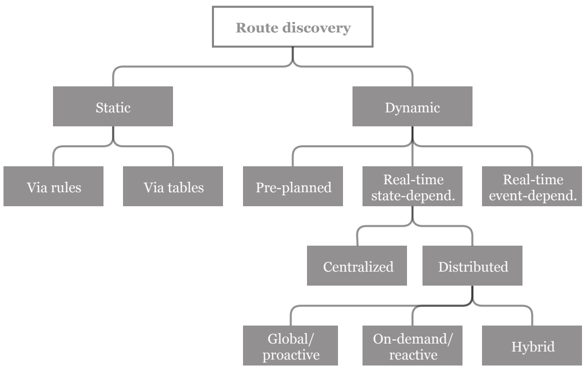

Figure 4: Classification scheme for route discovery strategies [compare 3, 73]

Dynamic routing strategies try to compensate the weakness of static routing strategies by taking the current network status (e.g. congestions) into account. The dynamic pre-planned routing is based on a routing table that is updated according to the network status. The points in time for updating the table are predefined, e.g. after a specific time span, after layout changes or depending on traffic patterns. Therefore, general knowledge about the traffic flow and possible future system behavior is required, to determine these points in time.

With real-time event-dependent and state-dependent approaches updates of the routing tables do not just happen at specific pre-planned time-points but may happen continuously. Using an event-dependent approach, multiple alternative routes are determined for a given source-destination constellation and depending on the current network status one of the routes is chosen. For example, a vehicle would always take the shortest route unless the utilization of this route is above a certain threshold or a path segment breaks down, etc. In contrast, real-time state-dependent routing uses no predefined routing tables or sets of suitable routes. The route is calculated based on the current network status, e.g. the utilization of single paths or recently finished transports are taken into consideration. Depending on the entity which makes decisions and the information that is used for the decision making this strategy can either be seen as centralized or distributed. In a centralized version of state-dependent routing a central unit has global knowledge of the network and in consequence is extremely responsive. In a more distributed algorithm the routing tables are not kept by one central unit. Instead distributed units, e.g. nodes in the graph have distributed versions of the routing tables. Distributed algorithms require a higher degree of information exchange but are said to be more robust to disturbances [see 73]. An alternative to a distributed storage of system information in nodes is aggregating information about the current system status via inter-vehicle communication. Vehicles can share their planned routes with the entire fleet or at least with other vehicles near-by. As a result, the knowledge is more distributed.

Three different forms of distributed state-dependent routing can be distinguished [see 3, 73]:

- Global/proactive routing:
  Global/proactive routing enables an up-to-date view of the network with periodic updates. Examples for this kind of routing are the optimized link state routing (OLSR) or the global state routing (GSR) [see 21, 62].
- On-demand/reactive routing:
  Instead of proactively creating an up-to-date view, these routing approaches are only executed when a routing request is received. Examples of this algorithm type are adaptive distance vector routing (ADVR) and dynamic source routing (DSR) [see 14, 111].
- Hybrid routing:
  Hybrid routing is a combination of features of proactive and reactive protocols. It reduces the overall effort for proactive routing by limiting the regarded area for updates to near-by nodes, often using predened zones. Reactive routing is used for finding routes to remote nodes [see e.g. 23, 56].

The main focus of this paper are decentralized control strategies. According to the classification scheme in figure 4 the weaker term “distributed” was used instead of “decentralized”. Lau and Woo [73] have chosen this term for non-central routing approaches, because without acquiring additional knowledge about the network from other nodes at all no meaningful local routing decision can be made.

We will show below that such distributed implementations do exist. A completely decentralized routing without any non-local information can not guarantee that a destination will be reached. In this case at least a minimum amount of aggregation of information is required. This limitation of the definition of decentrality has to be kept in mind when assessing the contributions of the authors below. Although these authors might use the concept of “decentralized” routing, they refer to a distributed implementation structure, but generally do not fulfill the requirement of purely using local information.

Route selection as a part of route planning

If there is only a single possible route between a given source-sink relation, the route selection will obviously be trivial. In conventional layouts usually multiple feasible routes or even multiple equivalent routes with regard to the decision criterion (f.e. distance) exist and a specific route needs to be selected. This decision is depending on the objective of the selection.

The problem of load balancing (spreading the vehicles) in decentralized systems has been recognized by several authors and adds a new challenge to the development of such algorithms. The shortest route is not necessarily the best choice as congestion might occur and in the worst case deadlocks endanger the system functionality. These risks can be mitigated by a central controller with global system view that redistributes the traffic flow if necessary. But in a decentralized systems new approaches are required because the overall system status is by definition unknown. Ng et al. [80] propose a load-scattering algorithm based on the average flow on the different alternative paths to a destination. They divide the total load along a certain path by the number of links and thereby derive an average flow figure. From all possible paths a traveling load selects the route with the lowest average flow. Other inspirations for this topic can be found in urban traffic networks [e.g. 4] or ad-hoc mobile networks [e.g. 119].

According to Wardrop [112] the selection decision can be made choosing one of two different principles:

- Goal routing:
  For each vehicle the best (usually shortest or fastest) route is selected.

- Route balancing:
  For each vehicle the route, which results in a global system optimum in the sense of minimal total travel time, is selected.

As it results in a global optimum the second principle is more favorable but also the more complicated goal. This is only possible for small layouts and usually requires extensive pre-calculation – see the already mentioned definition of scheduling according to [74]. Under incomplete information in a decentralized/distributed system, usually the first principle is reachable. Either with or without knowledge about the current system status a single vehicle always selects the route that is presumably best for itself.

Another strategy which was not discussed by Lau and Woo [73] consists of skipping the entire route planning (discovery and selection) process. Lost results can be replaced either with dynamic route execution approaches (see next subsection) or leads to a random vehicle routing process with completely local information.

Route execution

After a feasible route was found and selected using any approach of the route planning the vehicle can start executing the chosen route. The problem arising at this point is the avoidance of blockings and deadlocks with other vehicles. Blockings occur when two or more vehicle want to claim the same physical spot at the same time. Somehow a decision has to be made, which vehicle is allowed to go first. This is especially important at intersections. 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 system 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.

To avoid blockings and deadlocks, each vehicle has to somehow spread information about its intended path. This information can be shared for example with other vehicles, with single path segments and nodes or with a central entity.

Sticking to the requirements of providing conflict-free routes, several types of approaches have been developed [see e.g. 83]:

- Static methods:
  The entire route is blocked for the time a vehicle is moving
- Time-window-based methods:
  Only single path segments are blocked in the time window the vehicle is expected to travel on them
- Dynamic methods:
  Paths are blocked incrementally from path segment to path segment[7] while the vehicle is traveling and considering the current utilization of the next path segment

7 In literature this type of route execution is sometimes also known as “forwarding”.

The former two methods put more emphasis on a 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. However, complex layouts can be divided in less complex sub-layouts and the planning methods can be applied independently for each part. In case of conflicts, the priority on the limiting path segment can be regulated by various strategies, like a first come first serve (FCFS) strategy, by a strategy dependent on the priority of the transported loads or the vehicle destination. Furthermore, global information or some other information exchange is necessary for the static and the time-window based approach [see 74, 83].

In contrast, local information is sufficient for the dynamic methods. At each decision point (usually a switch) a decision is made if the next path segment can be traveled, or if some blockings occur. This method is even capable of incrementally constructing a route to a given destination, if there was no inital route planning. Global information about shortest paths or even the current system status can be stored in the nodes and can therefore be provided for passing vehicles.

Another aspect that is relevant for the dynamic decision is the right of way at merges. Especially in systems with a high number of vehicles, like carrier-based systems at airports this decision can have a huge impact on the system performance. Even though intersection control could be theoretically categorized as a part of dynamic routing execution we consider the field of intersection control not as a part of our scope.

2.3.2. Literature review on solving approaches and methods

There are multiple approaches for more or less distributed routing algorithms, e.g. the sub-problems of route planning and route execution. The next paragraphs illustrate the most influential ideas we found in literature. At first publications that follow the idea of multi-agent systems (MAS) for a distributed routing are presented. Afterwards, we discuss swarm-based approaches for example by reviewing the main ideas of ant algorithms. Finally, we take a look at approaches that transfer ideas and protocols from routing in ad-hoc communication networks to logistics.

Multi-agent systems (MAS)

For a short introduction to multi-agent systems we refer to chapter 2.1.2 or [116].

One of the first attempts to come up with a decentralized routing algorithm using the concept of agent-based systems was made by Taghaboni-Dutta and Tanchoco [103]. They developed an incremental route planner for automated guided vehicle (AGV) systems which does not select a route at the starting node, but rather decides the next node to travel to during the journey. According to the categorizations in figures 3 and 4 the route planning strategy can be seen as a reactive real time state dependent approach, which is executed dynamically. The key of the approach is a procedure, Taghaboni-Dutta and Tanchoco call "selectnextnode", which is incrementally executed by a vehicle that decides which path segment to take next with respect to a conflict-free journey. The input to this procedure consists of local information, like the possible next nodes, but also of global information like the estimated waiting times on subsequent nodes, which are calculated using queuing theory. As the vehicle calculates the decision independently, their conceptual system can be seen as one of the first agent-based routing systems. The authors evaluate the performance of their strategy in two simple layouts and compare the results with a complete-route-planner, which plans the entire conflict-free route before a vehicle starts traveling. They come to the conclusion that a complete-route-planner works better in complex layouts whereas the incremental route planner is equally suitable for simple layouts. However, the incremental route planning is less failure-prone and achieves significantly shorter response times. The incremental route planner combines local and global information to make the routing decisions. Thus, this approach is not completely decentralized.

Nishi et al. [82] have their focus on real-time systems considering a use case of an AGV system in a semiconductor fab. In a distributed manner each AGV calculates its own initial routing plan independently. In a subsequent step these initial plans are shared between the vehicles and are checked for feasibility. As long as the plans are not feasible, penalties are updated and therefore the routing decision gets changed. This so-called rescheduling has to be performed, whenever a new transportation request enters the system.

Lau and Woo [73] explicitly adopt the concept of MAS, where the nodes of a network work as cooperating agents. They introduce a hybrid distributed route planning algorithm for automated material handling systems. The algorithm uses a zone control logic and comprises a route discovery process which discovers feasible routes based on message broadcast, a multi-attribute route selection function and a fault management function in case the chosen route is blocked. The developed routing strategy outperforms several other strategies which the authors compare in a simulation study of a generic loop-based layout. The implementation of the approach can be regarded as decentralized. However, as the node agents are able to share information, the used information is not exclusively local.

Hallenborg [58] develop a MAS for a conveyor-based baggage handling system. The idea can be transferred to vehicle systems. The author focuses especially on the design of the agents and their communication. A route agent is responsible for route selection when a route is requested by another agent in the system. The cost factors, which represent the "virtual lenght" of a path are updated constantly according to the current traffic and the Dijkstra algorithm is used for computing the resulting shortest path. This means that there is still global information aggregated and in consequence the approach does not completely follow a decentralized pattern. The author describes 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. [60] develop a concept for a routing strategy which uses a label correction algorithm for decentralized route updates. The authors claim that this reduces the communication overhead. The routing algorithm is to a certain extent based on ideas from AGV literature. It pre-plans routes and uses this information to consider future path utilization in its route selection. However, the whole paper is conceptual and no performance evaluation is presented.

Among other strategies, Klein [69] proposes and evaluates a strategy that aims to be truly decentralized. As all agents rely exclusively on local information the vehicles and nodes do not have information about the global topology or the system status at all. A path planning for a given source destination combination is therefore not possible and even a dynamic route execution depends on information aggregation in the nodes. Consequently, at each switch a vehicle chooses randomly what path to take next. A destination is only reached by chance. Having obvious disadvantages, Klein calls this a "worst case strategy", with the only advantage of reducing the routing decisions to a minimum. As already mentioned in section 2.2 Klein uses this random routing in some cases for unladen vehicles. All in all, this approach can be seen as a rudimental MAS.

Schwarz et al. [95] convert the algorithm of ter Mors et al. [106], which is based on a graph network to a decentralized algorithm for vehicle routing. A vehicle that wants to reach a destination searches for the fastest way within the graph. Afterwards it reserves corresponding timeslots on each path segment. Other vehicles calculating their routes, take the reservation into account and in case of a conflict have to either take a different way or wait for the edge to be freed. The calculation is therefore done in a decentralized manner, but with global information about all reservations in the graph.

Schwarz [94] proposes a negotiation method between two vehicles in the phase of route execution, which is applied when a vehicle wants to travel on an already blocked path segment. The first vehicle with the reservation checks for alternative routes. In case the loss of taking an alternative route for the first vehicle is lower than the benefit for the second vehicle, the reservation is canceled. However, this may lead to many negotiations in crowded areas or even circle negotiations, as the first vehicle might have to negotiate with further vehicles for its alternative route. To avoid too many negotiations specific requirements are defined when a new negotiation is allowed at all. Due to the long-distance information exchange this approach is partly decentralized.

An analytical approach is presented by Digani et al. [32]. The authors introduce a path planning algorithm on a two layer architecture. The first layer (topological layer) consists of multiple sectors (macro-cells), representing an entire vehicle layout in an abstract manner. The algorithm for global path planning on this layer calculates the macro-cells that need to be crossed on the route to the destination sector. The calculation is done by each vehicle separately using the D*-algorithm[8]. This layer of the algorithm can be categorized as distributed real-time state dependent. The second layer (route map layer) represents the actual path segments within each sector. On this layer the actual route on the path segments is calculated using an A*-algorithm[9] and executed with a focus on conflict and deadlock avoidance. This is reached in a decentralized solution with focus on the current sector only and the different priorities of the vehicles in this sector. Therefore, the AGVs only share local information with neighboring vehicles. So Digani et al. use a decentralized implementation which relies on global and local information, depending on the level of calculations to be done. Even though not stated explicitly, the approach is close to agent-based methods. The approach is further refined in Digani et al. [31] and Digani et al. [30].

Abdenebaoui and Kreowski [1, 2] introduce graph-transformational swarms to model and route in a dynamically changing logistical network. The approach has two phases. In the first phase the layout is prepared in a way that each node is capable of indicating the shortest way to a given destination. Therefore, AGVs can solely follow local information. In the second phase the AGVs are following the shortest paths avoiding collisions. As the vehicles in the routing phase rely on decentralized information only and each vehicle has its own calculation unit the approach can be regarded as decentralized.

In Fanti et al. [39] each vehicle calculates the route to its destination using common algorithms, like the A*-algorithm. Consequently, the vehicle has information about the layout including cost factors. They propose an approach to avoid collisions and deadlocks, which they call coordination problem. Each time unit is divided into two sub units. In the first sub unit vehicles are able to communicate with neighboring AGVs about their routing plans. Usually the vehicle with the longest path has a higher priority. In the second sub unit the vehicles are actually moving. Rules like an AGV always has to complete a pass, before another AGV can enter the pass prevent deadlocks.

Without any specific mentioning of the route planning process and the method used, Zhang et al. [118] propose a cyber-physical system based control approach for a dynamic route execution of multiple vehicles. The vehicles are able to interact with each other within a given distance. The authors develop a car-following method, where AGVs follow other vehicles if there are no intersections. Furthermore, they present a method for overtaking, if a vehicle with a higher priority is following a low priority vehicle. A conflict warning method and an avoidance strategy are developed for intersections, where decentralized base stations monitor the traffic flow. Information is only exchanged within a certain radius. Therefore, this approach can be regarded as almost decentralized.

Swarm-based approaches

Swarm-based approaches are the second type of approaches discussed for decentralized routing. Within swarm-based approaches especially ant-based algorithms or ant colony optimization (ACO) are often said to be useful for decentralized control systems. Their development started in the 1990s and they use an analogy to real-world ant colonies in trying to imitate their indirect communication behavior via pheromone concentration. The idea of communicating indirectly by modifying the environment has been named “stigmergy” in the scientific community [see 28]. Although there are different versions of ACO, they all somehow rely on the stigmergy concept.

In logistics literature, it is often argued that ideas of swarm intelligence and ant behavior should be used for routing in logistics networks. Two different representations of real world system entities as ants can be distinguished. On the one hand artificial ants can be used in a network to explore it and find the shortest paths. Any vehicle could own a number of ants which help them to gather and spread information. The AntNet algorithm [29] will be presented as an example for this kind of algorithm. An IT infrastructure would be required to run the ant algorithm for each vehicle in a decentralized system. These optimization runs are computationally expensive. This can be especially problematic when only limited planning data is available and the real-time requirements are very high. 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. However, the results of this routing strategy cannot be expected to reach the same performance as the first implementation opportunity.

8 The D*-algorithm determines the shortest path between two nodes in a dynamic network. Originally it was described by [101].

9 The A*-algorithm is a popular algorithm that determines the shortest path between two nodes in a static network. It is the basis for the D*-algorithm and was originally described by [59].

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 [58].

AntNet was developed to apply ACO for routing in communication networks. See Di Caro and Dorigo [29] for a detailed description of the AntNet algorithm. In the following its procedure is explained briefly, however other versions of ant-based routing exist. (1) 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. (2) At each switch the ant selects its next node based on local, private and heuristic information. The ants do not communicate with each other but use the information that has been stored in the environment, i.e. based on stigmergy. The ant stores its chosen route nodes and additional information, e.g. the travel time. (3) Once it arrives at its destination the ant travels back to the starting point of its journey. It takes the exact same path back. 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. (4) 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 depending on the current destination to be reached. These probability tables represent the pheromone concentration. A direction is taken based on a combination of this probability and the queue length of the next link. Once an ant travels backwards to its initial starting point 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 a communication network [see e.g. 110]. But it needs to be considered that the AntNet algorithm has been developed as an optimization algorithm.

Claes et al. [24] propose an algorithm for decentralized anticipatory routing in road traffic which is based on the ACO. Each vehicle looking for a route in the traffic network has different kinds of artificial ants as helpers. The so-called exploration ants are “asking” each edge about assumed durations for crossing on a certain time point and can therefore calculated the fastest path to the destination. If such a path is defined, the intention ants will inform the single edges about the arrival of the vehicle in a certain time span enabling the edge to present this information for further exploration ants. This implementation is close to the time-window based route execution and the state dependent distributed route planning, we defined above. The information level in this implementation of an ant algorithm, where each vehicle has multiple ants can be seen as either local or global. On the one hand, a vehicle gets global information via exploration ants. On the other hand, these ants are spread and only aggregate information that is stored locally in the edges. As no central unit is calculating any solutions, the implementation can be regarded as decentralized. Even though the approach was originally designed for road traffic, the concept can be applied for vehicle transport systems as well.

Kanamori et al. [65] developed a similar approach in the field of traffic management which is also applicable for in-house logistical systems. Their approach is based on the anticipatory stigmergy model. Having identified the drawback that the stigmergy concept only displays past information, the anticipatory stigmergy concept shares information about future intentions. All routing decisions can be made taking these intentions into consideration.

Ideas from ad-hoc networks

The following two examples from Fay et al. [45] and ten Hompel et al. [105] show concepts that try to transfer the ideas of routing in mobile ad-hoc networks to logistics systems, which is the third approach presented for decentralized routing. Ad-hoc communication networks which change their configuration quickly and constantly are important areas of research. There are huge structural and behavioral similarities between logistics networks and 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. They provide analogies to the dynamics in complex logistics systems. Logistics entities travel in logistics networks like data packets in communication networks.

But besides all similarities, there are also several distinctive characteristics which need to be considered and do not make the development of decentralized logistics routing protocols an easy task [see 90, 105]. Johnstone et al. [64], who develop a dynamic routing policy based on learning agents, or Scholz-Reiter et al. [90], who propose a concept for a general logistics routing protocol are examples for the transfer of communication protocols to the logistics domain. We use the work of Fay et al. [45] and ten Hompel et al. [105] as a further example because they contain more detailed descriptions. As the knowledge of the communication protocols OLSR and DSR is essential for understanding the two examples, 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. [62] and Johnson and Maltz [63].

- optimized link state routing (OLSR):
  The basic idea of link state protocols is network nodes exchanging 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 calculates 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. As stated above, the OLSR can be categorized as a proactive distributed real-time state dependent method for route discovery.

- dynamic source routing (DSR):
  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). When 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 broadcasted 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[10]. 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 will be returned to the source and the routes in the cache will have to be updated. As stated above, the DSR can be categorized as a reactive distributed real-time state dependent method for route discovery.

All in all, the proactive protocol has the advantage that a route is immediately available when required. On the other hand DSR creates less communication overhead.

Fay et al. [45] transfer the ideas of OLSR to a baggage handling application. They 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 decentralized routing strategy. ten Hompel et al. [105] also use a baggage handling system simulation to test their routing strategies. They develop a multi-agent control system based on the concepts of the DSR algorithm. But as several adjustments to the initial baggage handling systems are required to ensure an implementation, they 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. Both authors realize that besides finding a certain route with the help of the adjusted communication protocols, it is also necessary to balance the load if there exist several routes to one destination. ten Hompel et al. [105] 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. [45] 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 multiple routes with more or less equal travel time are available, the load should be distributed equally to them and even routes 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.

Excursus: Free ranging vehicles

A current trend in the sector of AGV routing are so-called free ranging vehicles, which means that vehicles do not follow a physical or virtual guidance. This leads to a better space utilization and a higher layout flexibility, but more complex control strategies are needed. As free-ranging vehicles do not exactly fit the scope of this survey only a single example is mentioned. Demesure et al. [27] propose an agent-based approach for routing free-ranging vehicle in areas without predefined path. The approach is based on two steps. In the first step, a central controller uses global information to calculate the trajectory for a specific AGV. This trajectory shows the intention of the vehicle and is shared with the environment, e.g. the other vehicle. In the second step of the approach, the trajectory of neighboring vehicles are coordinated in a distributed/decentralized manner. If conflicts between multiple vehicles are expected, their trajectories will be updated iteratively according to transport priorities. The second step therefore serves as a decentralized collision avoidance and leads to a huge complexity reduction. Regarding the information used, the approach is based on global and local information – depending on the step of the approach.

10 This can also be an interesting aspect for vehicle-based transport systems. Depending on the network topology, a (virtual) path layout can also consist of equidistant pieces.

2.3.3. *Summary*

Table 3 shows the overview of the literature concerning this chapter. MAS are the most widely used and promising approaches concerning decentralized routing implementations.

Table 3: Literature overview ordered by topics for routing

Topic	Corresponding literature
Multi-agent systems	[30–32, 58, 60, 69, 73, 82, 94, 95, 103, 106, 118]
Swarm-based	[24, 28, 29, 58, 65, 110]
Ad-hoc networks	[45, 64, 90, 105]
Other	[1, 2, 27, 39]

All in all, it can be stated that truly decentralized approaches including a decentralized implementation and local information do almost not exist in routing literature. Enabling a sensitive route discovery, at least the path layout needs to be known either by the vehicle or by path nodes. Advanced tasks, like a conflict avoidance or some kind of route balancing relies on a direct or indirect communication between different entities further violating the decentralized concept. Nevertheless, some approaches rely on some kind of compromise using only regional information for route discovery and vehicle communication. As a consequence advantages of non-decentralized approaches are achieved and the communication effort is still limited.

3. CONCLUSION AND FUTURE RESEARCH

In this paper the available scientific literature on the decentralized control of vehicle systems in in-house logistics has been reviewed. The focus was on online systems that work without any pre-arrival information about future transport requests. The main objective was to analyze existing decentralized concepts and identify fields for future research. In addition to providing a general overview, it was investigated if existing decentralized concepts truly base their decision making process on local information only.

Three different control strategy problems for vehicle-based in-house transport systems were analyzed in detail: (1) load-vehicle assignment, (2) empty vehicle balancing and (3) routing. The first two problems are merged to the dispatching problem. Each control problem was analyzed in a subsection considering basics and major categorizations as well as a segment on relevant literature.

Regarding the load-vehicle assignment the approaches could be divided into layout transformation approaches and multi-agent systems (MAS) with a majority of publications in the field of the latter one. Nevertheless, truly decentralized approaches were mentioned rarely as most approaches use some kind of communication and therefore do not rely on local information only. However, the amount of data that needs to be exchanged for a load-vehicle assignment seems to be manageable in (partly) centralized systems.

The literature about empty vehicle balancing was less extensive. The approaches can be divided into a planing and controlling aspect, which are linked closely. A variety of publications dealt with the balancing of empty vehicles in overhead hoist transport (OHT) systems. All in all, some of the approaches can be regarded as truly decentralized.

It could be seen that most publications on decentralized control of vehicle-based transport systems deal with routing. This is a logical consequence, since the effort in communication and complexity of the routing problem solution can be very high. Approaches are either based on the multi-agent or swarm-based paradigm or try to transfer ideas from ad-hoc communication networks to logistical systems. Decentralized routing concepts generally make use of a distributed implementation, but usually rely at least to a certain extent on information sharing or global information.

In general, many of the mentioned publications are of a rather conceptual nature, which means it is only described how a system should work. A prove of feasibility or advanced performance measuring e.g. in comparison to a central benchmark strategy is often lacking. Furthermore, the complexity of the regarded (simulation) models is often very low. As a consequence, the feasibility of the proposed control strategies in real-world applications often remains unclear.

There is a definite lack of contributions that mention and analyze emergent system characteristics. Based on the decentralized paradigm, these should be observable. This seems to be an important topic, as emergent behavior is not necessarily beneficial for the system performance. It rather makes a system harder to control.

The identified shortcomings lead to the following directions for future research:

- Further analysis of decision making based on local information:
  In general, it needs to be evaluated in detail if the pure usage of local information can be meaningful. Hence, the trade-off between communication overhead for information exchange and system efficiency needs to be analyzed when using different amounts of information.
- Detailed performance analysis:
  It can be stated that typical characteristics of real-world application are usually not sufficiently taken into account. Examples are specific transport pattern or various path networks. Consequently, decentralized concepts need to be analyzed in more realistic test or simulation systems in order

to broadly verify their functionality and assess whether they are applicable in real-world transport systems. In this context the emergent system behavior can be analyzed and comparisons with other control systems, e.g. a central benchmark approach are possible. Therefore, it is important to define objectively clear standard problems for assessing vehicle control strategies on the basis of typical real world systems.

- Basis of decision-making:
  Implementing decentralized systems in real-world scenarios is challenging as there is a lack of methods for applying the design of the control system and choosing the level of (de)centrality. Furthermore, some of the benefits of decentralized systems are also difficult to quantify and measure like robustness, scalability or redundancy. However, these characteristics can have a decisive impact on the performance of a decentralized system and as a consequence they influence the decision for a more centralized or decentralized control system architecture. A procedure (design of experiments) for an extensive comparison of qualitative and quantitative indicators for various control approaches need to be developed. Additionally, this procedure can be applied for the standard problem mentioned above.
- Mixed operations scenarios:
  As vehicle systems are flexible by nature they are often applied in combination with other systems and consequently with various paradigms of controls. As different connected systems will strongly affect each other, this raises the question of strategies for mixed systems.

REFERENCES

1. Abdenebaoui, L. and Kreowski, H.-J. “Decentralized Routing of Automated Guided Vehicles by Means of Graph-Transformational Swarms”. In: *Dynamics in Logistics*. Ed. by M. Freitag, H. Kotzab, and J. Pannek. Cham: Springer International Publishing, 2017, pp. 457–467. doi: 10.1007/978-3-319-45117-6_40 (cit. on pp. 24, 28).
2. Abdenebaoui, L. and Kreowski, H.-J. “Modeling of decentralized processes in dynamic logistic networks by means of graph-transformational swarms”. In: *Logistics Research* 9.1 (2016). doi: 10.1007/s12159-016-0147-6 (cit. on pp. 24, 28).
3. Abolhasan, M., Wysocki, T., and Dutkiewicz, E. “A review of routing protocols for mobile ad hoc networks”. In: *Ad Hoc Networks* 2.1 (2004), pp. 1–22. doi: 10.1016/S1570 – 8705(03) 00043-X (cit. on pp. 19, 20).
4. Adacher, L., Flamini, M., and Nicosia, G. “Decentralized algorithms for multiple path routing in urban transportation networks”. In: *Symposium on Transportation Analysis,(Phuket, Thailand)*. 2007 (cit. on p. 21).
5. Aeschbach, P. et al. “Balancing bike sharing systems through customer cooperation – a case study on London’s Barclays Cycle Hire”. In: *2015 54th IEEE Conference on Decision and Control (CDC)*. 2015 54th IEEE Conference on Decision and Control (CDC). 2015, pp. 4722–4727. doi: 10.1109/CDC.2015.7402955 (cit. on p. 17).
6. Le-Anh, T. and de Koster, M. *Multi-attribute dispatching rules for AGV systems with many vehicles*. 2004 (cit. on pp. 15, 17).
7. Le-Anh, T., de Koster, M., and Yu, Y. “Performance evaluation of dynamic scheduling approaches in vehicle-based internal transport systems”. In: *International Journal of Production Research* 48.24 (2010), pp. 7219–7242. doi: 10.1080/00207540903443279 (cit. on p. 7).
8. Le-Anh, T. and de Koster, M. “A review of design and control of automated guided vehicle systems”. In: *European Journal of Operational Research* 171.1 (2006), pp. 1–23. doi: 10.1016/j.ejor.2005.01.036 (cit. on p. 7).
9. Arsie, A., Savla, K., and Frazzoli, E. “Efficient Routing Algorithms for Multiple Vehicles With no Explicit Communications”. In: *IEEE Transactions on Automatic Control* 54.10 (2009), pp. 2302–2317. doi: 10.1109/TAC.2009. 2028954 (cit. on pp. 11, 13).
10. Arsie, A. and Frazzoli, E. “Efficient routing of multiple vehicles with no explicit communications”. In: *International Journal of Robust and Nonlinear Control* 18.2 (2007), pp. 154–164. doi: 10.1002/rnc.1258 (cit. on pp. 11, 13).
11. Ayanian, N., Rus, D., and Kumar, V. “Decentralized Multirobot Control in Partially Known Environments with Dynamic Task Reassignment”. In: IFAC *Proceedings Volumes* 45.26 (2012), pp. 311–316. doi: 10.3182/20120914-2-US-4030.00029 (cit. on pp. 12, 13).
12. Bartholdi, J. J. and Platzman, L. K. “Decentralized Control of Automated Guided Vehicles on a Simple Loop”. In: *IIE Transactions* 21.1 (1989), pp. 76–81. doi: 10.1080/07408178908966209 (cit. on pp. 8, 13).
13. Berman, S. and Edan, Y. “Decentralized autonomous AGV system for material handling”. In: *International Journal of Production Research* 40.15 (2002), pp. 3995–4006. doi: 10.1080 /00207540210146990 (cit. on pp. 11, 13).
14. Boppana, R. V. and Konduru, S. P. “An adaptive distance vector routing algorithm for mobile, ad hoc networks”. In: Conference on Computer Communications. Vol. 3. 2001, 1753–1762 vol.3. doi: 10.1109/INFCOM.2001.916673 (cit. on p. 20).
15. Bozer, Y. A. and Srinivasan, M. M. “Tandem AGV systems: a partitioning algorithm and

performance comparison with conventional AGV systems”. In: *European Journal of Operational Research* 63.2 (1992), pp. 173–191 (cit. on pp. 8, 13).
16. Bruno, G., Ghiani, G., and Improta, G. “Dynamic positioning of idle automated guided vehicles”. In: *Journal of Intelligent Manufacturing* 11.2 (2000), pp. 209–215 (cit. on pp. 15, 17).
17. Caraballo, L. et al. “The block-information-sharing strategy for task allocation:A case study for structure assembly with aerial robots”. In: *European Journal of Operational Research* 260.2 (2017), pp. 725–738. doi: 10.1016/j.ejor.2016.12.049 (cit. on pp. 12, 13).
18. Chaabane, A. B. et al. \Analyzing the impact of key parameters of vehicle management policies in a unified AMHS”. In: *2013 Winter Simulations Conference (WSC)*. 2013, pp. 3818–3828. doi: 10.1109/WSC.2013.6721741 (cit. on pp. 16, 17).
19. Chang, S.-H. and Egbelu, P. J. “Dynamic relative positioning of AGVs in a loop layout to minimize mean system response time”. In: *International Journal of Production Research* 34.6 (1996), pp. 1655–1673. doi: 10.1080 /00207549608904989 (cit. on pp. 14, 17).
20. Chemla, D., Meunier, F., and Wolfler Calvo, R. “Bike sharing systems: Solving the static rebalancing problem”. In: *Discrete Optimization* 10.2 (2013), pp. 120–146. doi: 10.1016/j.disopt.2012.11.005 (cit. on p. 17).
21. Chen, T.-W. and Gerla, M. “Global state routing: a new routing scheme for ad-hoc wireless networks”. In: *IEEE International Conference on Communications (ICC)*. Vol. 1. 1998, 171–175 vol.1. doi: 10.1109/ICC.1998.682615 (cit. on p. 20).
22. Choi, H.-L., Brunet, L., and How, J. “Consensus-Based Decentralized Auctions for Robust Task Allocation”. In: *IEEE Transactions on Robotics* 25.4 (2009), pp. 912–926. doi: 10.1109/TRO.2009.2022423 (cit. on pp. 9, 10, 13).
23. Choudhury, R. R., Paul, K., and Bandyopadhyay, S. “MARP: a Multi-Agent Routing Protocol for Mobile Wireless Ad Hoc Networks”. In: *Autonomous Agents and Multi-Agent Systems* 8.1 (2004), pp. 47–68. doi: 10.1023/B:AGNT.0000009410.57024.9a (cit. on p. 20).
24. Claes, R., Holvoet, T., and Weyns, D. “A Decentralized Approach for Anticipatory Vehicle Routing Using Delegate Multiagent Systems”. In: *IEEE Transactions on Intelligent Transportation Systems* 12.2 (2011), pp. 364–373. doi: 10.1109/TITS.2011.2105867 (cit. on pp. 25, 28).
25. Colling, D. et al. “Dezentrale Auftragserzeugung und-vergabe für FTF”. In: *Logistics Journal: Proceedings* 2016.10 (2016) (cit. on pp. 10, 13).
26. DeMaio, P. “Bike-sharing: History, Impacts, Models of Provision, and Future”. In: *Journal of Public Transportation* 12.4 (2009), pp. 41–56. doi: 10.5038/2375-0901.12.4.3 (cit. on p. 17).
27. Demesure, G. et al. “Decentralized Motion Planning and Scheduling of AGVs in an FMS”. In: *IEEE Transactions on Industrial Informatics* 14.4 (2018), pp. 1744–1752. doi: 10.1109/TII.2017.2749520 (cit. on pp. 27, 28).
28. Dhillon, S. and Van Mieghem, P. “Performance analysis of the AntNet algorithm”. In: *Computer Networks* 51.8 (2007), pp. 2104–2125. doi: 10.1016/j.comnet.2006.11.002 (cit. on pp. 24, 28).
29. Di Caro, G. and Dorigo, M. “AntNet: Distributed stigmergetic control for communications networks”. In: *Journal of Artificial Intelligence Research* 9 (1998), pp. 317–365 (cit. on pp. 25, 28).
30. Digani, V., Sabattini, L., and Secchi, C. “A Probabilistic Eulerian Traffic Model for the Coordination of Multiple AGVs in Automatic Warehouses”. In: *IEEE Robotics and Automation Letters* 1.1 (2016), pp. 26–32. doi: 10.1109/LRA.2015.2505646 (cit. on pp. 24, 28).
31. Digani, V. et al. “Ensemble Coordination Approach in Multi-AGV Systems Applied to Industrial Warehouses”. In: *IEEE Transactions on Automation Science and Engineering* 12.3 (2015), pp. 922–934. doi: 10.1109/TASE.2015.2446614 (cit. on pp. 24, 28).
32. Digani, V. et al. “Hierarchical traffic control for partially decentralized coordination of multi agv systems in industrial environments”. In: *Robotics and Automation (ICRA), 2014 IEEE International Conference on.* IEEE, 2014, pp. 6144–6149 (cit. on pp. 24, 28).
33. Dijkstra, E. W. “A note on two problems in connexion with graphs”. In: *Numerische mathematik* 1.1 (1959), pp. 269–271 (cit. on p. 19).
34. Draganjac, I. et al. “Decentralized Control of Multi-AGV Systems in Autonomous Warehousing Applications”. In: *IEEE Transactions on Automation Science and Engineering* 13.4 (2016), pp. 1433–1447. doi: 10.1109/TASE.2016.2603781 (cit. on p. 2).
35. Egbelu, P. J. “Positioning of automated guided vehicles in a loop layout to improve response time”. In: *European Journal of Operational Research* 71.1 (1993), pp. 32–44 (cit. on pp. 14, 17).
36. Egbelu, P. J. and Tanchoco, J. M. A. “Characterization of automatic guided vehicle dispatching rules”. In: *International Journal of Production Research* 22.3 (1984), pp. 359–374. doi: 10.1080/00207548408942459 (cit. on p. 6).
37. Erol, R. et al. “A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems”. In: *Applied Soft Computing* 12.6 (2012), pp. 1720–

1732. doi: 10.1016/j.asoc.2012.02.001 (cit. on pp. 10, 13).

38. Fanti, M. P. et al. “Discrete consensus in networks with constrained capacity”. In: *52nd IEEE Conference on Decision and Control.* 52nd IEEE Conference on Decision and Control. 2013, pp. 2012–2017. doi: 10.1109/CDC.2013.6760177 (cit. on pp. 12, 13).
39. Fanti, M. P. et al. “A decentralized control strategy for the coordination of AGV systems”. In: *Control Engineering Practice* 70 (2018), pp. 86–97. doi: 10.1016/j.conengprac.2017.10.001 (cit. on pp. 12, 13, 24, 28).
40. Fanti, M. P. et al. \Assignment of electrical vehicles to charging stations by a distributed approach”. In: *Control Conference (ECC), 2014 European.* IEEE, 2014, pp. 1888–1893 (cit. on pp. 12, 13).
41. Fanti, M. P. et al. “Discrete consensus for asynchronous distributed task assignment”. In: *Decision and Control (CDC), 2016 IEEE 55th Conference on.* IEEE, 2016, pp. 251–255 (cit. on pp. 12, 13).
42. Fauadi, M. H. F., Yahaya, S. H., and Murata, T. “Intelligent combinatorial auctions of decentralized task assignment for AGV with multiple loading capacity”. In: *IEEJ Transactions on Electrical and Electronic Engineering* 8.4 (2013), pp. 371–379. doi: 10.1002/tee.21868 (cit. on pp. 10, 13).
43. Fauadi, M. H. F. M. “Agent-based material transportation scheduling of AGV systems and its manufacturing applications”. PhD thesis. Waseda University, 2012 (cit. on pp. 10, 13).
44. Fay, A. and Fischer, I. “Dezentrale Automatisierungsstrategien für Gepäckbeförderungssysteme (Decentralized Automation Strategies for Baggage Transportation Systems)”. In: *at – Automatisierungstechnik/Methoden und Anwendungen der Steuerungs-, Regelungs-und Informationstechnik* 52.7 (2004), pp. 335–341 (cit. on pp. 9, 13).
45. Fay, A., Jerenz, S., and Seitz, N. “Dezentrale Steuerung von Transportsystemen in Analogie zum Routing in Datennetzen (Decentralized Control of Transport Systems based on Data Routing Mechanisms)”. In: *at – Automatisierungstechnik* 56.6 (2008). doi: 10.1524/auto.2008.0708 (cit. on pp. 2, 26–28).
46. Fazlollahtabar, H. and Saidi-Mehrabad, M. “Methodologies to Optimize Automated Guided Vehicle Scheduling and Routing Problems: a Review Study”. In: *Journal of Intelligent & Robotic Systems* 77.3 (2015), pp. 525-545. doi: 10.1007/s10846-013-0003-8 (cit. on p. 7).
47. Ferrero, F. et al. “Car-sharing services: An annotated review”. In: *Sustainable Cities and Society* 37 (2018), pp. 501–518. doi: 10.1016/j.scs.2017.09.020 (cit. on p. 17).
48. FIPA Protokoll. url: http://www.fipa.org/specs/fipa00029/SC00029H.pdf (visited on 12/12/2018) (cit. on p. 10).
49. Flämig, H. “Autonome Fahrzeuge und autonomes Fahren im Bereich des Gütertransportes”. In: *Autonomes Fahren.* Ed. by M. Maurer et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 377–398. doi: 10.1007/978-3-662-45854-9_18 (cit. on p. 2).
50. Floyd, R. W. “Algorithm 97: Shortest Path”. In: *Commun. ACM* 5.6 (1962), pp. 345–. doi: 10.1145/367766.368168 (cit. on p. 19).
51. Furmans, K., Nobbe, C., and Schwab, M. “Future of Material Handling–modular, flexible and efficient”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems.* 2011 (cit. on p. 1).
52. Gademann, A. J. R. M. and van de Velde, S. L. “Positioning automated guided vehicles in a loop layout”. In: *European Journal of Operational Research* 127.3 (2000), pp. 565–573 (cit. on pp. 15, 17).
53. Gansterer, M. and Hartl, R. F. “Collaborative vehicle routing: a survey”. In: *European Journal of Operational Research* 268.1 (2018), pp. 1–12. doi: 10.1016/j.ejor.2017.10.023 (cit. on p. 3).
54. Garey, M. R. and Johnson, D. S. *Computers and intractability.* 29th ed. New York: wh freeman, 2002 (cit. on pp. 2, 7).
55. Giordani, S., Lujak, M., and Martinelli, F. “A distributed multi-agent production planning and scheduling framework for mobile robots”. In: *Computers & Industrial Engineering* 64.1 (2013), pp. 19-30. doi: 10.1016/j.cie.2012.09.004 (cit. on pp. 10, 13).
56. Haas, Z. J. and Pearlman, M. R. “The performance of query control schemes for the zone routing protocol”. In: *IEEE/ACM Transactions on Networking* 9.4 (2001), pp. 427–438. doi: 10.1109/90.944341 (cit. on p. 20).
57. Hahn-Woernle, C. “Neue Anforderungen für die Logistik des 21. Jahrhunderts”. In: *Internet der Dinge in der Intralogistik.* Ed. by W. Günthner and M. ten Hompel. VDI-Buch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 9–13. doi: 10.1007/978-3-642-04896-8_2 (cit. on p. 2).
58. Hallenborg, K. “Decentralized scheduling of baggage handling using multi-agent technologies”. In: *Multiprocessor scheduling theory and applications.* Vienna: I-Tech Education and Publishing, 2007 (cit. on pp. 15, 17, 23, 25, 28).
59. Hart, P. E., Nilsson, N. J., and Raphael, B. “A formal basis for the heuristic determination of minimum cost paths”. In: *IEEE transactions on Systems Science and Cybernetics* 4.2 (1968), pp. 100–107 (cit. on p. 24).

60. Hofmeister, M., Baier, G., and Gärtner, M. "Strategien für die dezentrale agentenbasierte Steuerung von Materialfusssystemen". In: *Internet der Dinge in der Intralogistik*. Ed. by W. A. Günthner and M. ten Hompel. Berlin: Springer, 2010, pp. 119–139 (cit. on pp. 23, 28).
61. Hu, C.-H. and Egbelu, P. J. "A framework for the selection of idle vehicle home locations in an automated guided vehicle system". In: *International Journal of Production Research* 38.3 (2000), pp. 543–562. doi: 10.1080/002075400189293 (cit. on pp. 13–15, 17).
62. Jacquet, P. et al. "Optimized Link State Routing Protocol for Ad Hoc Networks". In: *Proceedings of IEEE Multi Topic Conference: Technology for the 21st Century*. IEEE Press, 2001, pp. 62–68 (cit. on pp. 20, 26).
63. Johnson, D. B. and Maltz, D. A. \Dynamic source routing in ad hoc wireless networks". In: *Mobile computing*. Springer, 1996, pp. 153–181 (cit. on p. 26).
64. Johnstone, M., Creighton, D., and Nahavandi, S. "Status-based Routing in Baggage Handling Systems: Searching Verses Learning". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40.2 (2010), pp. 189–200. doi: 10.1109/TSMCC.2009.2035519 (cit. on pp. 26, 28).
65. Kanamori, R., Takahashi, J., and Ito, T. "Evaluation of anticipatory stigmergy strategies for traffic management". In: *2012 IEEE Vehicular Networking Conference (VNC)*. IEEE, 2012, pp. 33–39 (cit. on pp. 26, 28).
66. Khamis, A., Hussein, A., and Elmogy, A. "Multirobot Task Allocation: a Review of the State-of-the-Art". In: *Cooperative Robots and Sensor Networks 2015*. Ed. by A. Koubâa and J. Martínez-de Dios. Vol. 604. Cham: Springer International Publishing, 2015, pp. 31–51. doi: 10.1007/978-3-319-18299-5_2 (cit. on p. 6).
67. Kiba, J.-E. et al. "Simulation of a Full 300mm-Semiconductor Manufacturing Plant with Material Handling Constraints". In: *Winter Simulation Conference*. WSC '09. Austin, Texas, 2009, pp. 1601–1609 (cit. on pp. 16, 17).
68. Kim, K. H. and Kim, J. Y. "Estimating mean response time and positioning idle vehicles of automated guided vehicle systems in loop layout". In: *Computers & industrial engineering* 33.3 (1997), pp. 669–672 (cit. on pp. 14, 17).
69. Klein, N. "The impact of decentral dispatching strategies on the performance of intralogistics transport systems". PhD thesis. 2012 (cit. on pp. 3–5, 11–13, 16, 17, 23, 28).
70. Koo, P.-H. and Jang, J. "Vehicle travel time models for AGV systems under various dispatching rules". In: *International Journal of Flexible Manufacturing Systems* 14.3 (2002), pp. 249-261 (cit. on p. 7).
71. Koo, P.-H., Jang, J., and Suh, J. "Vehicle dispatching for highly loaded semiconductor production considering bottleneck machines first". In: *International Journal of Flexible Manufacturing Systems* 17.1 (2005), pp. 23–38. doi: 10.1007/s10696-005-5992-6 (cit. on p. 8).
72. Kuhn, H. W. "The Hungarian method for the assignment problem". In: *Naval Research Logistics Quarterly* 2.1 (1955), pp. 83–97. doi: 10.1002/nav.3800020109 (cit. on p. 10).
73. Lau, H. Y. K. and Woo, S. O. "An agent-based dynamic routing strategy for automated material handling systems". In: *International Journal of Computer Integrated Manufacturing* 21.3 (2008), pp. 269–288. doi: 10.1080/09511920701241624 (cit. on pp. 18–21, 23, 28).
74. Le Anh, T. *Intelligent control of vehicle-based internal transport systems =: Intelligente besturing van interne voertuigtransportsystemen*. ERIM Ph.D. series research in management 51. OCLC: 255012955. Rotterdam: Erasmus Research Inst. of Management (ERIM), 2005. 182 pp. (cit. on pp. 6, 7, 13, 18, 21, 22).
75. Lee, C. and Ventura, J. A. "Optimal dwell point location of automated guided vehicles to minimize mean response time in a loop layout". In: *International Journal of Production Research* 39.17 (2001), pp. 4013–4031. doi: 10.1080/00207540110054605 (cit. on pp. 15, 17).
76. Lee, N. M. Y., Lau, H. Y. K., and Ko, A. W. Y. "An Immune Inspired Algorithm for Solving Dynamic Vehicle Dispatching Problem in a Port Container Terminal". In: *Artificial Immune Systems*. International Conference on Artificial Immune Systems. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2009, pp. 329–342. doi: 10.1007/978-3-642-03246-2_30 (cit. on pp. 11, 13).
77. Legros, B. \Dynamic repositioning strategy in a bike-sharing system; how to prioritize and how to rebalance a bike station". In: European Journal of Operational Research (2018). doi: 10.1016/j.ejor.2018.06.051 (cit. on p. 17).
78. Martín, J. et al. "Decentralized Robot-Cloud Architecture for an Autonomous Transportation System in a Smart Factory". In: *SEMANTICS Workshops* (2017) (cit. on pp. 10, 13).
79. Maxwell, W. L. and Muckstadt, J. A. "Design of Automatic Guided Vehicle Systems". In: *IIE Transactions* 14.2 (1982), pp. 114–124. doi: 10.1080/05695558208975046 (cit. on p. 8).
80. Ng, A. K., Efstathiou, J., and Lau, H. Y. "A Load Scattering Algorithm for Dynamic Routing of Automated Material Handling Systems". In: *Computational Intelligence and Security*. Vol. 2. IEEE, 2006, pp. 1020–1025 (cit. on p. 20).
81. Nieke, C. "Materialflusssteuerung heute und ihre Defizite". In: *Internet der Dinge in der Intralogistik*. Ed. by W. Günthner and M.

ten Hompel. VDI-Buch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–21. doi: 10.1007/978-3-642-04896-8_3 (cit. on p. 2).
82. Nishi, T., Ando, M., and Konishi, M. "Experimental studies on a local rescheduling procedure for dynamic routing of autonomous decentralized AGV systems". In: *Robotics and ComputerIntegrated Manufacturing* 22.2 (2006), pp. 154–165. doi: 10.1016/j.rcim.2005.02.010 (cit. on pp. 22, 28).
83. Qiu, L. et al. "Scheduling and routing algorithms for AGVs: a survey". In: *International Journal of Production Research* 40.3 (2002), pp. 745–760. doi: 10.1080/00207540110091712 (cit. on pp. 18, 21, 22).
84. Raviv, T., Tzur, M., and Forma, I. A. "Static repositioning in a bike-sharing system: models and solution approaches". In: *EURO J Transp Logist* 2.3 (2013), pp. 187–229. doi: 10.1007/s13676-012-0017-6 (cit. on p. 17).
85. Ross, E. A., Mahmoodi, F., and Mosier, C. T. "Tandem configuration automated guided vehicle systems: a comparative study". In: *Decision Sciences* 27.1 (1996), pp. 81–102 (cit. on pp. 8, 13).
86. Schmaler, R. et al. "Simulation based evaluation of different empty vehicle management strategies with considering future transport jobs". In: *Winter Simulation Conference (WSC).* 2017, pp. 3576–3587. doi: 10.1109/WSC.2017.8248071 (cit. on pp. 16, 17).
87. Schmaler, R. et al. "Strategies to Empower Existing Automated Material Handling Systems to Rising Requirements". In: *IEEE Transactions on Semiconductor Manufacturing* 30.4 (2017), pp. 440–447. doi: 10.1109/TSM.2017.2756104 (cit. on pp. 16, 17).
88. Schmauder, M. and Spanner-Ulmer, B. *Ergonomie: Grundlagen zur Interaktion von Mensch, Technik und Organisation.* Hanser, Carl, 2014 (cit. on p. 2).
89. Scholz-Reiter, B., Kolditz, J., and Hildebrandt, T. "Engineering autonomously controlled logistic systems". In: *International Journal of Production Research* 47.6 (2009), pp. 1449–1468. doi: 10.1080/00207540701581791 (cit. on p. 2).
90. Scholz-Reiter, B., Rekersbrink, H., and Freitag, M. "Kooperierende Routingprotokolle zur Selbststeuerung von Transportprozessen". In: *Industrie Management* 22.3 (2006), pp. 7–10 (cit. on pp. 26, 28).
91. Scholz-Reiter, B. et al. "Dynamik logistischer Systeme". In: *Beiträge zu einer Theorie der Logistik.* Ed. by P. Nyhuis. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 109–138. doi: 10.1007/978-3-540-75642-2_6 (cit. on p. 2).
92. Scholz, M. et al. "Software-in-the-loop testbed for multi-agent-systems in a discrete event simulation: Integration of the Java Agent Development Framework into Plant Simulation". In: International Systems Engineering Symposium (ISSE). IEEE, 2017, pp. 1–6 (cit. on p. 9).
93. Schreiber, S. "Entwicklung einer Vergleichs- und Bewertungsmöglichkeit von dezentralen Steuerungsarchitekturen für Produktionssysteme". PhD thesis. Hamburg: Universität der Bundeswehr Hamburg, 2013 (cit. on p. 2).
94. Schwarz, C. "Untersuchung zur Steigerbarkeit von Flexibilität, Performanz und Erweiterbarkeit von Fahrerlosen Transportsystemen durch den Einsatz dezentraler Steuerungstechniken". PhD thesis. Universität Oldenburg, 2014 (cit. on pp. 10, 13, 23, 28).
95. Schwarz, C. et al. "Selbstgesteuerte Fahrerlose Transportsysteme". In: *Logistics Journal* 2013.12 (2013) (cit. on pp. 10, 13, 23, 28).
96. Seibold, Z. and Furmans, K. "Plug&Play-Fördertechnik in der Industrie 4.0". In: *Handbuch Industrie 4.0 Bd.3: Logistik.* Ed. by B. Vogel-Heuser, T. Bauernhansl, and M. ten Hompel. Springer Reference Technik. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, pp. 3–20. doi: 10.1007/978-3-662-53251-5_2 (cit. on p. 1).
97. Shen, W. et al. "Applications of agent-based systems in intelligent manufacturing: An updated review". In: *Advanced Engineering Informatics* 20.4 (2006), pp. 415–431. doi: 10.1016/j.aei.2006.05.004 (cit. on p. 2).
98. Sinriech, D. and Tanchoco, J. M. A. "Solution methods for the mathematical models of single-loop AGV systems". In: *International Journal of Production Research* 31.3 (1993), pp. 705–725. doi: 10.1080/00207549308956752 (cit. on pp. 3, 8, 13).
99. Sinriech, D., Tanchoco, J., and Herer, Y. T. "The segmented bidirectional single-loop topology for material flow systems". In: *IIE Transactions* 28.1 (1996), pp. 40–54. doi: 10.1080/07408179608966251 (cit. on pp. 8, 13).
100. Smith, R. G. \The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver". In: *IEEE TRANSACTIONS ON COMPUTERS* 12 (1980), p. 10 (cit. on p. 9).
101. Stentz, A. "Optimal and efficient path planning for partially-known environments". In: *Proceedings IEEE International Conference on Robotics and Automation* (1994) (cit. on p. 24).
102. Sternberg, H. and Andersson, M. "Decentralized intelligence in freight transport – a critical review". In: *Computers in Industry* 65.2 (2014), pp. 306–313. doi: 10.1016/j.compind.2013.11.011 (cit. on p. 3).
103. Taghaboni-Dutta, F. and Tanchoco, J. M. A. "Comparison of dynamic routeing techniques

for automated guided vehicle system". In: *International Journal of Production Research* 33.10 (1995), pp. 2653–2669. doi: 10.1080/00207549508945352 (cit. on pp. 18, 22, 28).

104. Talbot, L. "Design and performance analysis of multistation automated guided vehicle systems". PhD thesis. Universite Catholique de Louvain, 2003 (cit. on pp. 15, 17).
105. Ten Hompel, M., Libert, S., and Roidl, M. *Erarbeitung von Methoden und Regeln zur Gestaltung agentengestützter, dezentraler Steuerungen für den Einsatz in komplexen Materialflusssystemen.* Forschungsbericht Projekt-Nr. 15313 N. Dortmund: Lehrstuhl für Förder- und Lagerwesen, 2010 (cit. on pp. 2, 26–28).
106. Ter Mors, A. W., Zutt, J., and Witteveen, C. "Context-Aware Logistic Routing and Scheduling". In: (2007), p. 8 (cit. on pp. 23, 28).
107. Ter Mors, A. W. "The world according to MARP: Multi-Agent Route Planning". PhD thesis. Delft: Technische Universiteit Delft, 2010 (cit. on pp. 2, 19).
108. Vis, I. F. \Survey of research in the design and control of automated guided vehicle systems". In: *European Journal of Operational Research* 170.3 (2006), pp. 677–709. doi: 10.1016/j.ejor.2004.09.020 (cit. on p. 7).
109. Vogel, P., Greiser, T., and Mattfeld, D. C. "Understanding Bike-Sharing Systems using Data Mining: Exploring Activity Patterns". In: *Procedia – Social and Behavioral Sciences* 20 (2011), pp. 514–523. doi: 10.1016/j.sbspro.2011.08.058 (cit. on p. 17).
110. Wang, J. et al. "HOPNET: a hybrid ant colony optimization routing algorithm for mobile ad hoc network". In: *Ad Hoc Networks* 7.4 (2009), pp. 690–705 (cit. on pp. 25, 28).
111. Wang, L. et al. "Multipath source routing in wireless ad hoc networks". In: *Canadian Conference on Electrical and Computer Engineering.* Vol. 1. 2000, 479–483 vol.1. doi: 10.1109/ CCECE.2000.849755 (cit. on p. 20).
112. Wardrop, J. G. "Road paper. some theoretical aspects of road traffic research." In: *Proceedings of the Institution of Civil Engineers* 1.3 (1952), pp. 325–362. doi: 10.1680/ipeds.1952.11259 (cit. on p. 21).
113. Warshall, S. "A Theorem on Boolean Matrices". In: *J. ACM* 9.1 (1962), pp. 11–12. doi: 10.1145/321105.321107 (cit. on p. 19).
114. Wertz, R. et al. "Improved Empty Vehicle Balancing in Automated Material Handling Systems". In: *Tenth International Conference on Computer Modeling and Simulation (uksim 2008).* Tenth International Conference on Computer Modeling and Simulation (uksim 2008). 2008, pp. 732–733. doi: 10.1109/ UKSIM.2008.101 (cit. on pp. 16, 17).
115. Weyns, D. and Holvoet, T. "Architectural design of a situated multiagent system for controlling automatic guided vehicles". In: *International Journal of Agent-Oriented Software Engineering* 2.1 (2008), pp. 90{128 (cit. on pp. 9, 10, 13, 16).
116. Weyns, D. et al. "Decentralized control of automatic guided vehicles: applying multi-agent systems in practice". In: ACM Press, 2008, p. 663. doi: 10.1145/1449814.1449819 (cit. on pp. 9, 13, 16, 17, 22).
117. Yang, P., Freeman, R. A., and Lynch, K. M. "Multi-Agent Coordination by Decentralized Estimation and Control". In: *IEEE Transactions on Automatic Control* 53.11 (2008), pp. 2480–2496. doi: 10.1109/TAC.2008.2006925 (cit. on p. 9).
118. Zhang, Y., Zhengfei, Z., and Jingxiang, L. "CPS-Based smart control model for shopfloor material handling". In: I*EEE Transactions on Industrial Informatics* 14.4 (2017), pp. 1764–1775 (cit. on pp. 24, 28).
119. Zou, F. et al. "Load Balance Routing using Packet Success Rate for Mobile Ad Hoc Networks". In: IEEE, 2007, pp. 1624–1627. doi: 10.1109/ WICOM.2007.409 (cit. on p. 21).