

The Impact of Shortest-Path Searches on Dynamic Autonomous Transport Scheduling

Max Gath, Otthein Herzog and Maximilian Vaske

Abstract The requirements of transport processes have become increasingly complex due to shorter transit times, the individual qualities of shipments, and higher amounts of small sized orders. Especially in courier and express services providing same day deliveries, the high degree of dynamics even increases this complexity. To ensure reliable and flexible planning and control of transport processes, we present a reactive and proactive agent-based system to support the dispatching of logistic transport service providers. Beside the application in simulated real-world processes of our industrial partners, this paper focuses on the impact and relevance of shortest-path queries in the system. We compare the application of state-of-the-art algorithms and investigate the effects of high speed shortest-path computations in agent-based negotiations. The results prove that efficient shortest-path algorithms are an essential key component in agent-based control of dynamic transport processes.

Keywords Shortest-path algorithms · Planning and scheduling · Autonomous logistic processes · Multiagent systems

M. Gath (✉) · O. Herzog · M. Vaske

Institute for Artificial Intelligence, TZI—Center for Computing
and Communication Technologies, Universität Bremen, Bremen, Germany
e-mail: mgath@tzi.de

O. Herzog
e-mail: herzog@tzi.de

M. Vaske
e-mail: zottel@tzi.de

Introduction

The growing cost pressure in logistics to offer competitive prices, while increasing the quality of services requires logistic service providers to setup more efficient processes. Especially in courier and express services, the optimization of planning and control processes is a complex problem. In courier and express services small- and medium-sized shipments have to be transported within guaranteed time windows and probably within a few hours. Regarding the dispatching process, the general problem can be mapped to the well-known vehicle routing problem (VRP) (Parragh et al. 2008a) or pickup and delivery problem (PDP) (Parragh et al. 2008b). However, the complexity of process planning is even increased by changing amounts and individual qualities of shipments like weight, volume, priority, and value. Handling this complexity in real situations is further aggravated by the high degree of dynamics that results from unexpected events such as rapidly changing order situations and delays. Moreover, the exact amount and properties of shipments are not known in advance. Actual capacities are only revealed while serving tasks. Picked-up shipments decrease the vehicles' capacities during ongoing tours. To react to changing traffic conditions and delays at incoming goods departments, it is essential to adapt tours and timetables while considering actual capacities.

Transport Processes of Courier and Express Services

We started with a detailed documentation and an analysis of the relevant processes of our industrial partners and revealed the interdependencies between processes and actors by modeling the business processes with the well-established *Business Process Modeling Notation* (BPMN). In order to cover the general planning and control, the collected information has been enriched by interviews with other transport service providers and logistics experts. While the dispatching processes of several forwarding agencies are varying in detail, e.g., by applying different software systems, the general procedure is not varying substantially. First, an information system collects all incoming orders and assigns these orders to predefined tours by a static mapping, e.g., of postal codes to tours. In general, the first assignment neither considers the amount of effectively received orders nor the properties of shipments or available vehicles, but is an essential preprocessing step. Second, this allocation is optimized by the dispatcher. Time critical orders are processed with higher priority and orders of overloaded tours are identified and reassigned to tours with available capacities. After this rough planning a fine-grained planning process is started by each contracted freight carrier. In this step, the freight carrier schedules its trucks and assigns the orders to one of its vehicles. Next, the freight operator determines the route by applying its expertise and additional knowledge, e.g., about preferred time slots of incoming goods

departments. Damaged or time critical orders which cannot be processed within guaranteed time windows are reported to the forwarding agency. Thus, external transport providers may be instructed to transport these time critical goods. In conclusion, the general planning and control processes in groupage traffic reduce the problem complexity by splitting up the overall problem (the assignment of all orders to vehicles while considering relevant constraints) into smaller problems with less complexity. After the preprocessing, each contracted transport company is solving the reduced problem to assign a subset of orders to a subset of vehicles. However, this implies that possible dependencies and optimization potentials between orders of different transport companies are not detected and consequently neglected. Next, we identified the data, which can serve as a basis for the decisions in autonomous processes and analyzed their format, amount, quality, feasibility, relevance, and the point of operation where the data is available. Consequently, processing only data, which is available in current processes, allows for the integration of autonomous logistics processes without any new hardware investments. For instance, software systems should not consider the exact volume, as there is no reliable information available in current information systems, because the volume of heterogeneous goods cannot determine automatically so far. With the data provided by our industrial partners we aggregated several performance indicators to quantitatively describe the current stages of processes and to identify the optimization potential in the business processes, which is related to dispatching. In conclusion, the general problem of our industrial partners refers to the VRP (Golden et al. 2008) which is concerned with determining tours with minimum costs for a fleet of vehicles to satisfy customer requests at different destinations while the start and end point of the tour is the depot.

Related Work

In forwarding agencies, information technologies support dispatchers in their decision making who still create tours manually on the basis of individual long-term experiences. Indeed, there are several professional transport management systems (TMS) for planning and controlling transport processes like PTV, 4Flow, or EURO-LOG.¹ Moreover, in the past decades, numerous efficient heuristics and metaheuristics have been developed for the transportation domain such as ant systems, tabu search, simulated annealing, and genetic algorithms (Parragh et al. 2008a, b; Golden et al. 2008; Gendreau and Bräysy 2005). However, the dynamics in logistics and individual requirements of the application domain are often neglected. Central planning and control in dynamic and complex logistic processes is increasingly difficult due to the requirements of flexibility and adaptability to changing environments. The goal is to optimize the planning and control processes

¹See: <http://www.ptvgroup.com>; <http://www.4flow.de>; <http://www.eurolog.com>.

by providing operational plans, tours, and routes automatically while considering all relevant data to enable optimal decision making at any point of the operations.

In autonomous logistic processes, the decision making is shifted from central, hierarchical planning, and control systems to decentralized, heterarchical systems (Scholz-Reiter et al. 2004). Intelligent software agents represent logistic entities, e.g., containers or vehicles. Thus, they are able to plan and schedule their way throughout the logistic network autonomously (Schuldt 2011). The agents act on behalf of the represented objects and try to reach the objectives assigned to them by their owners. Consequently, relevant information is directly linked to products. For instance, an agent representing a shipment is aware of its individual weight, volume, and its designated place and time of arrival. As a result, the material flow is directly connected to the information flow, which allows agents to receive and process relevant data immediately. By considering real-time information about the status of the physical world, the quality of the agents' decision-making processes is improved. The agents apply and share this knowledge by communication and negotiation mechanisms with other agents, in order to optimize the efficiency of processes and the resource utilization. Semantic technologies such as domain-specific ontologies, communication protocols, and speech acts are applied to ensure the unambiguous exchange of information. By delegating planning and control processes to decentralized entities, the overall problem is split into smaller problem instances that can be solved optimally. The advantages of applying multiagent systems are high flexibility, adaptability, scalability, and robustness of decentralized systems through problem decomposition and the proactive, reactive, and adaptive behavior of intelligent agents (Wooldridge 2013). Therefore, agent systems are especially applied to open, unpredictable, dynamic, and complex environments. There are many examples of multiagent applications within logistic processes for resource allocation, scheduling, optimization, and controlling. Agent-based commercial systems are used within the planning and control processes of containerized freight (Dorer and Calisti 2005). In-house logistic material handling systems have been implemented with self-controlled and self-configured components (Lewandowski et al. 2013). Agent-based systems have optimized planning and control processes within dynamic environments (Fischer et al. 1995; Harjes and Scholz-Reiter 2013). Other ranges of application have been provided for industrial logistic processes (Skobelev 2011).

Agent-Based Dispatching in Dynamic Transport Processes

In our developed system, agents represent transport vehicles and orders. The agents differ in their individual properties, e.g., represented vehicles vary in their capacities, work schedules, and speed limits. Similarly, each *order agent* carries the unique characteristics of its represented shipment such as the pickup and delivery location, weight, value, time windows, and premium service constraints. The goal of *order agents* is to find a proper transport service provider for transporting the

shipment from the depot to the destination (or vice versa) within given time windows. *Vehicle agents* negotiate and communicate with *order agents* to maximize the number of carried shipments while satisfying all relevant constraints and premium service priorities.

The system starts with the rough planning step by applying a k-means algorithm (Mac Queen 1967). In contrast to a static mapping from postal codes to tours, the algorithm assigns only effectively arrived orders to available vehicles by grouping orders in nearby districts. Consequently, the flexibility of the rough planning step is increased because the system is able to react to daily as well as seasonal fluctuations. After the rough planning step, each *vehicle agent* starts a detailed planning process. On the one hand, the *vehicle agent* considers the represented truck's capacity, the driving times dependent on the type of the road and the respective speed limits, as well as the individual capacities of the shipments such as the weight, priority, time windows, handling times, and the pickup or delivery location. On the other hand, the agent optimizes the objective functions to reduce cost and determine efficient solutions. For instance, the vehicle identifies the shortest-path for visiting all stops. As a result, the agent solves a selective traveling salesman problem (TSP), which is NP hard (Christofides 1976). The design of two different TSP solver is described by Edelkamp et al. (2013) and by Edelkamp and Gath (2013). The input of the solvers is the distance matrix including all the cities, which have to be visited, and the current position of the vehicle. Thus, several shortest-path searches are applied for the computation of this distance matrix. Especially, in dynamic environments with changing traffic conditions and in scenarios in which orders have to be scheduled during operations, it is infeasible to precalculate all the distances offline. This may only be an adequate solution for static problems when all service requests are known in advance. Section "[Shortest-Path Algorithms](#)" describes state-of-the-art and often applied shortest-path algorithms which have been implemented within the decision-making processes of the agents. After the detailed planning step of each *vehicle agent*, several orders may not be serviced by a vehicle. Thus, the responsible agent acts in the same way like agents representing dynamically incoming orders: The agent sends a transport request to available *vehicle agents* and starts a new negotiation. The *vehicle agents* compute proposals by determining their additional cost for transporting the shipment. In order to schedule new orders also while transporting other shipments, the agent considers all relevant changes of the environment and its internal state, e.g., already loaded shipments and the current position of the vehicle. The computed cost is sent back to the *order agent* that chooses the transport provider with the least cost. If it is not possible to satisfy the orders' requirements, a *refuse message* is sent by the *vehicle agent*. To transport a premium service instead of conventional orders, or another premium service with less cost, already accepted orders (that have not been boarded yet) may not be included in the new plan and have to be rescheduled. Affected *order agents* negotiate with other transport service providers again. The agent models consider concurrency aspects within

negotiations, the dynamics of the environment, as well as the interdependencies between planning and the execution of an existing plan. Details of the proactive and reactive agent design are provided by Gath et al. (2013a, b).

Shortest-Path Algorithms

In this section, we present three state-of-the-art algorithms, which were implemented within the decision-making processes of the agents. Section “[Dijkstra-Shortest-Path](#)” describes the well-known Dijkstra algorithm (Dijkstra 1959). Section “[A* Algorithm](#)” continues with the A* algorithm. Section “[Hub-Labeling on Contraction Hierarchies](#)” provides a high speed algorithm optimized for road networks which is based on hub-labeling (Abraham et al. 2012) and contraction hierarchies (CH) (Geisberger et al. 2012). All algorithms are complete and optimal.

Dijkstra-Shortest-Path

Let N denote the set of nodes, E a set of edges and $\text{dist}: E \rightarrow \mathbb{R}$ a distance function of an edge. The Dijkstra shortest-path algorithm (Dijkstra 1959) is probably the best known and most frequently applied algorithm for the computation of a shortest-path P with the minimum distance $\min_P \sum_{e \in P} \text{dist}(e)$ between two nodes $s, t \in N$. Nodes are labeled as visited, reachable, or unvisited and contain a reference to a predecessor node and a distance to s . At first all nodes are unvisited with an infinite distance. While reachable nodes exist and target t has not visited the algorithm which performs the following steps: First, it chooses the reachable nearest node c . Node c is labeled as visited and all unvisited nodes n connected to an outgoing edge are evaluated next. Let d_n denote the distance from s to n . If n is reachable and d_n is smaller than its currently saved distance, mark n as reachable with distance d_n and predecessor c . After termination, if there exists a path from s to t , the distance of t denotes the shortest distance from s to t .

A Algorithm*

The A* algorithm is similar to the Dijkstra algorithm, but applies an additional heuristic which underestimates the real cost from a processed node to the target to push the search in the right direction and avoid the expansion of nodes which are not part of the shortest-path. For instance, on road networks the Euclidian air distances may be used as a valid heuristic. We implemented a memory efficient

version of the A* algorithm which is based on radix heaps. As a result, each node can be processed in constant time and space. Details about this algorithm are provided by Greulich (2013).

Hub-Labeling on Contraction Hierarchies

Since 2011, hub-labeling algorithms (HL) (Abraham et al. 2011) in combination with CH (Geisberger et al. 2012) have become the most efficient approach for shortest-path queries. For instance, shortest-path queries on the whole transport network of West-Europe are processed in less than a millisecond (Abraham et al. 2011, p. 239; 2012, p. 34). First, the CH and hub-labels are computed for each node offline. The general idea of distance labeling algorithms is that the distance between two nodes is only determined by the comparison of their assigned labels. Thus, search queries on the generated labels are efficiently performed online. Hub-labels contain a list with references to several other nodes (the hubs). At the labeling processes, the *cover property* has to be fulfilled: “for any two vertices s and t , there exists a vertex on the shortest s - t path that belongs to both labels (of s and t)” (Abraham et al. 2011, p. 230). Consequently, the cover property ensures that all shortest-paths in a graph may be determined by the labels of the source and target nodes. Especially if the labeling algorithm is applied on nodes saved in CH, this allows memory efficient label representations. In order to build a CH, a preprocessing step is started which extends the original graph g to a larger graph g' . The resulting graph g' contains direct shortcuts between nodes which are represented by paths in g . The algorithm iterates over every node and calculates shortcuts and assigns a distinct level in the CH to the node. For each node a priority value is calculated. The most important value is the *edge difference* between the current graph and the graph resulting from processing that node. The node with the lowest priority is processed next. The performance of the algorithm depends on the sequence of nodes added to the CH (Geisberger et al. 2012). Due to the fact that the sequence relies on the priority of the nodes and the priority changes by the extension of the CH, the priorities are updated continuously after adding a new node to the CH. As the computation of the priorities is cost intensive, it is estimated. The better the approximated values are, the less shortcuts are determined and the more efficient is the memory consumption and the search on the CH. As a result, the priority ordering accelerates the shortest-path searches on the CH because it is goal directed and the shortcuts decrease the number of nodes, which are processed during the search. There are also approaches, which can be applied on time-dependent graphs (Batz et al. 2008) or on dynamically changing graphs (Geisberger et al. 2012).

Evaluation

To describe the system's performance quantitatively and show its applicability, we conclude the results of the case study described by Gath et al. (2013b). For the evaluation, we applied the agent-based event-driven simulation platform PlaSMA.² PlaSMA allows the integration of real-world infrastructures imported from OpenStreetMap³ and supports the continuous simulation of uninterrupted time intervals. Moreover, it includes a geographic information system (GIS) to determine the coordinates of addresses. It is designed for the modeling, simulation, evaluation, and optimization of planning and control processes in logistics. We simulated real-world scenarios based on process data and orders provided by our industrial partners and modeled the road network of their whole business area. In our first investigation, we simulated all orders effectively transported by our industrial partner within two representative weeks. The complete road network infrastructure contains 156,722 traffic junctions and 365,609 roads. It includes all relevant highways, motorways, and inner city roads based on the OpenStreetMap database. Beside its type, also additional information about the distance and the speed limit are included in this infrastructure model. The maximum speed of vehicles is 80 km/h. To cover real-world conditions, a vehicle's speed is reduced to 80 % of the speed limit of the road section. As customers sometimes refuse to accept the delivery of a shipment during operations this is analogously simulated dynamically. The details of the experimental setup are described by Gath et al. (2013b).

Figure 1 shows the amount of shipments which are successfully processed and which are changed between agents in continuous negotiations to optimize the allocation. Note that a shipment may be changed for multiple times if the representing *order agent* finds another vehicle that transports the shipment with less cost.

Within the simulation 1,347,092, TSPs are solved within the decision-making processes of the agents. In relation to the total number of shipments, the high number of changes indicates that the initial allocation is continuously optimized by agent-based negotiations. Thus, the agent system allows for an efficient grouping of packages at pickup and delivery locations. Loads at the same location are transported by a single vehicle, because vehicles generate no additional cost for transporting further shipments from or to an already visited location. In contrast to current processes, the number of shipments transported by external transport providers is thereby reduced by more than 80 %. As a result, the dispatching system reduces the daily costs significantly by increasing the capacity utilization of the own vehicle fleet and by reducing the required number of external transport providers. In addition, the results reveal that a huge number of TSPs have to be solved within the decision-making processes of the agents. For each TSP a distance matrix is computed, which includes information that is only available during daytime operations such as the current position of the vehicle and the locations of new incoming orders. Pre- or offline

²<http://plasma.informatik.uni-bremen.de>.

³<http://www.openstreetmap.org>.

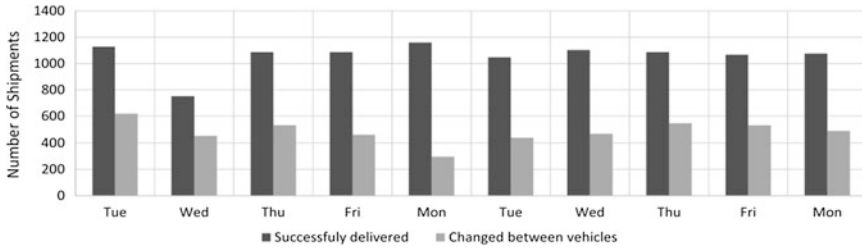


Fig. 1 Shipments which are successfully processed and changed between agents

computation of distance matrices is not feasible because this could result in a table with $|N| \times |N|$ entries. This obviously exceeds the memory requirements.

Figure 2 shows the physical time required for the simulation of a scenario as well as the average number of node expansions required in a single search with different shortest-path algorithms. As the hub-labeling not explicitly expand nodes, we compare the average number of nodes represented by a label. Within the search, this list of nodes has to be processed to identify the shortest way. Note, that all algorithms are well-established shortest-path searches. Likewise to the above-investigated case study, in this scenario we simulated real-world processes with orders provided by our industrial partner. The underlying transport infrastructure contains 85,633 nodes and 196,647 edges. All experiments were performed on a laptop computer with an Intel quad-core i72620-M CPU/2.7 GHz and 16 GB RAM. Memory requirements are not exceeded. Figure 2 clearly indicates that the average number of nodes, which is processed in a single search, is strongly related to the time performance of the overall approach. Although the A* search significantly reduces the number of node expansions, the performance is not increased in the same way. This is due to the time consumption and afford of the heuristic, which is applied at each node expansion to determine the Euclidian distance on the sphere surface of the earth as lower bound cost. Thus, in larger graphs the gap between the number of node expansions and improved time performance of the A* algorithm is decreasing. The results prove a significant impact of the shortest-path.

Algorithm to the computation time of the agent-based dispatching approach even on small infrastructures investigated in this experiment. As the applied TSP solvers

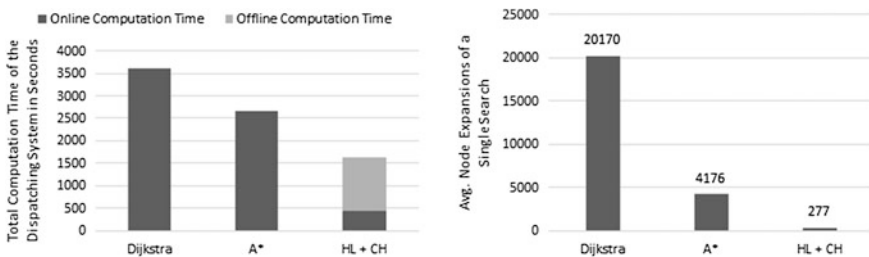


Fig. 2 Comparison of the scenario's computation time and the number of node expansions

are state-of-the-art (Edelkamp and Gath 2013; Edelkamp et al. 2013), this shows that the application of the well-established Dijkstra and A* algorithm is the most time-expensive operation of the agent-based dispatching system. In general, this is remarkable because the Shortest-Path Problem is in P , while the TSP is an NP-hard problem but easier to handle in real-world application. Moreover, other possible bottlenecks of the system such as the performance of the applied agent management system are not relevant. Thus, it is even infeasible in industrial applications with real-time computational requirements, to apply Dijkstra or A* algorithms in agent-based dispatching systems, but necessary to switch to recently developed high performance algorithms for shortest-path computations.

Conclusion

In this chapter, we presented an agent-based approach for dynamic planning and scheduling in transport logistics. The focus is on shortest-path queries, which provide the basis for the agent's decision-making processes. We presented three well-established implementations of shortest-path searches and compared their impact to the run-time performance of the dispatching system. On one hand, the results show that the developed system optimizes the planning and control processes by our industrial partners. The agent-based dispatching system provides high quality tours, and the system significantly decreases daily costs by reducing the required number of external transport providers. On the other hand, the results prove that standard search algorithms preclude the system's industrial application. In conclusion, an efficient high speed shortest-path algorithm, such as hub-labeling on CH, is a key component and essential for the industrial application of agent-based dispatching systems. Future research will focus on even larger graphs, further speed-up techniques, dynamically changing infrastructures, and the integration of the agent-based dispatching system in Industry 4.0 applications.

Acknowledgments The presented research was partially funded by the German Research Foundation (DFG) within the project Autonomous Courier and Express Services (HE 989/14-1) at the University Bremen, Germany. We thank our industrial partners for great cooperation and Stefan Edelkamp for great discussions.

References

- Abraham I, Delling D, Goldberg AV, Werneck RF (2011) A hub-based labeling algorithm for shortest paths in road networks. In: Experimental algorithms. 10th international symposium SEA 2011. Lecture notes in computer science, vol 6630. Springer, Berlin, pp 230–241
- Abraham I, Delling D, Goldberg AV, Werneck RF (2012) Hierarchical hub labelings for shortest paths. In: Algorithms—ESA 2012. 20th Annual European symposium, Ljubljana, September 2012. Lecture notes in computer science, vol 7501. Springer, Berlin, pp 24–35

- Batz GV, Delling D, Sanders P, Vetter C (2008) Time-dependent contraction hierarchies. In: Proceedings of the 11th workshop on algorithm engineering and experiments, New York
- Christofides N (1976) Worst-case analysis of a new heuristic for the travelling salesman problem. In: Technical Report 388, Graduate School of Industrial Administration, Carnegie-Mellon University
- Dijkstra EW (1959) A note on two problems in connexion with graphs. *Num Math* 1:269–271
- Dorer K, Calisti M (2005) An adaptive solution to dynamic transport optimization. In: Proceedings of the 4th international conference on autonomous agents and multiagent systems, ACM Press, pp 45–51
- Edelkamp S, Gath M (2013) Optimal decision making in agent-based autonomous groupage traffic. In: Proceedings of the 5th international conference on agents and artificial intelligence, vol 1, pp 248–254
- Edelkamp S, Gath M, Cazenave T, Teytaud F (2013) Algorithm and knowledge engineering for the TSPTW problem. In: IEEE symposium on computer intelligence in scheduling, Singapore
- Fischer K, Müller JP, Pischel M (1995) Cooperative transportation scheduling: an application domain for dai. *J Appl Artif Intell* 10:1–33
- Gath M, Herzog O, Edelkamp S (2013a) Agent-based dispatching in groupage traffic. In: IEEE symposium on computer intelligence in production and logistics systems, Singapore
- Gath M, Herzog O, Edelkamp S (2013b) Agent-based planning and control for groupage traffic. In: Proceedings of the 10th international conference and expo of emerging technologies (to appear)
- Geisberger R, Sanders P, Schultes D, Vetter C (2012) Exact routing in large road networks using contraction hierarchies. *Transp Sci* 46:388–404
- Gendreau M, Bräysy O (2005) Vehicle routing problem with time windows, part I: route construction and local search algorithms. *Transp Sci* 39:104–118
- Golden B, Raghavan S, Wasil E (eds) (2008) The vehicle routing problem: latest advances and new challenges. Springer, New York
- Greulich C (2013) Enhanced shortest path computation for multiagent-based intermodal transport planning in dynamic environments. In: Proceedings of the 5th international conference on agents and artificial intelligence, vol 2, pp 324–329
- Harjes F, Scholz-Reiter B (2013) Agent-based disposition in event logistics. *Res Logistics Prod* 2:137–150
- Lewandowski M, Werthmann D, Gath M, Lawo M (2013) Agent-based control for material handling systems in in-house logistics. In: European conference on smart objects, systems and technologies
- Mac Queen J (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley symposium on mathematical statistics and probability, vol 1, pp 281–297
- Parragh SN, Doerner KF, Hartl RF (2008a) A survey on pickup and delivery problems, part I: transportation between customers and depot. *J für Betriebswirtschaft* 58(1):21–51
- Parragh SN, Doerner KF, Hartl RF (2008b) A survey on pickup and delivery problems part II: transport between pickup and delivery locations. *J für Betriebswirtschaft* 58(2):81–117
- Scholz-Reiter B, Windt K, Freitag M (2004) Autonomous logistic processes: new demands and first approaches. In: Proceedings of the 37th CIRP international seminar on manufacturing systems, pp 357–362
- Schuldt A (2011) Multiagent coordination enabling autonomous logistics. Springer, Berlin
- Skobelev P (2011) Multi-agent systems for real time resource allocation, scheduling, optimization and controlling: industrial applications. In: Mařík V, Vrba P, Leitão P (eds) Holonic and multi-agent systems for manufacturing. Springer, Berlin, pp 1–14
- Wooldridge M (2013) Intelligent agents. In: Weiss G (ed) Multiagent systems. A modern approach to distributed artificial intelligence. MIT Press, Cambridge, pp 3–50