Logistics Research (2023) 16:10
DOI_10.23773/2023_10

Scheduling Automated Guided Vehicles considering transport load transfers

P. Boden, S. Rank and T. Schmidt

Received: 1 June 2022 / Accepted: 29 August 2023 / Published online: 11 October 2023

ABSTRACT

This paper investigates a scheduling approach for Automated Guided Vehicle Systems, enabling transport load transfers between vehicles during transport execution. The vehicles can use predefined transfer stations to buffer transport loads until the transport continues by a following vehicle. The objective is to improve system performance by decreasing vehicle utilization to yield higher throughput. Transfer operations are planned ad-hoc depending on the current system state.

We describe the task assignment as a Pickup and Delivery Problem with Transfers. Since the problem is classified as NP-hard, an Adaptive Large Neighborhood Search heuristic is given. Test instances and a material flow simulation study evaluate that transport load transfers potential for improvement depends significantly on the characteristics of the transport system.

KEYWORDS: Transport Load Transfers · Automated Guided Vehicles · Task Assignment · Scheduling · Pickup and Delivery Problem with Transfers

Patrick Boden
patrick.boden@tu-dresden.de

Sebastian Rank

Thorsten Schmidt

Chair of Material Handling and Logistics Engineering
TUD Dresden University of Technology,
01062 Dresden

This article is part of a focus collection on "Dynamics in Logistics – Models and Algorithms for Optimisation, Planning and Control."

1 INTRODUCTION

Automated Guided Vehicles (AGV) automate transport processes in production and logistics systems. The strategies for controlling Automated Guided Vehicle Systems (AGVS) are essential to ensure the transport system's performance and reliable equipment supply. The assignment of transport jobs to vehicles and the decision about the processing sequence, referred to as task assignment, have a high impact on reaching these goals. Task assignment for dispatching and scheduling of AGV has been extensively studied in scientific research and for practical application. So far, it premise that a transport job passed to the AGV system will be executed by a single vehicle. The disadvantage is that a potential for increasing efficiency is neglected that could be realized by the technical components of the AGV system.

We consider transport load transfers to enable the exchange of transport loads between vehicles during the transport execution. Transfer operations are planned dynamically (ad-hoc) concerning the system state and not predefined by higher-level control rules. A vehicle drops a transport load at a predefined transfer location (e.g. a shelf) to execute a transfer operation. A following vehicle continues the transport. This flexibility could help reduce the length of vehicle tours through synergy effects, e.g. by combining transports with similar destinations. The ambition is to increase the flexibility of AGVS in the execution of transports to minimize vehicle utilization to yield higher throughput.

Figure 1 compares schedules of a task assignment based on selecting transport tasks by the nearest location to the vehicle position by a fictitious example to demonstrate the basic idea. While transport load transfers are neglected in figure 1 (a), they have been taken into account in figure 1 (b). The scenario is defined by start and end locations for two vehicles (AGV1 start/end, AGV2 start/end) and two transport jobs (Job1 start/end, Job2 start/end). The routes intersect, which leads to a benefit due to a transfer operation if an exchange at the transfer location (TP)

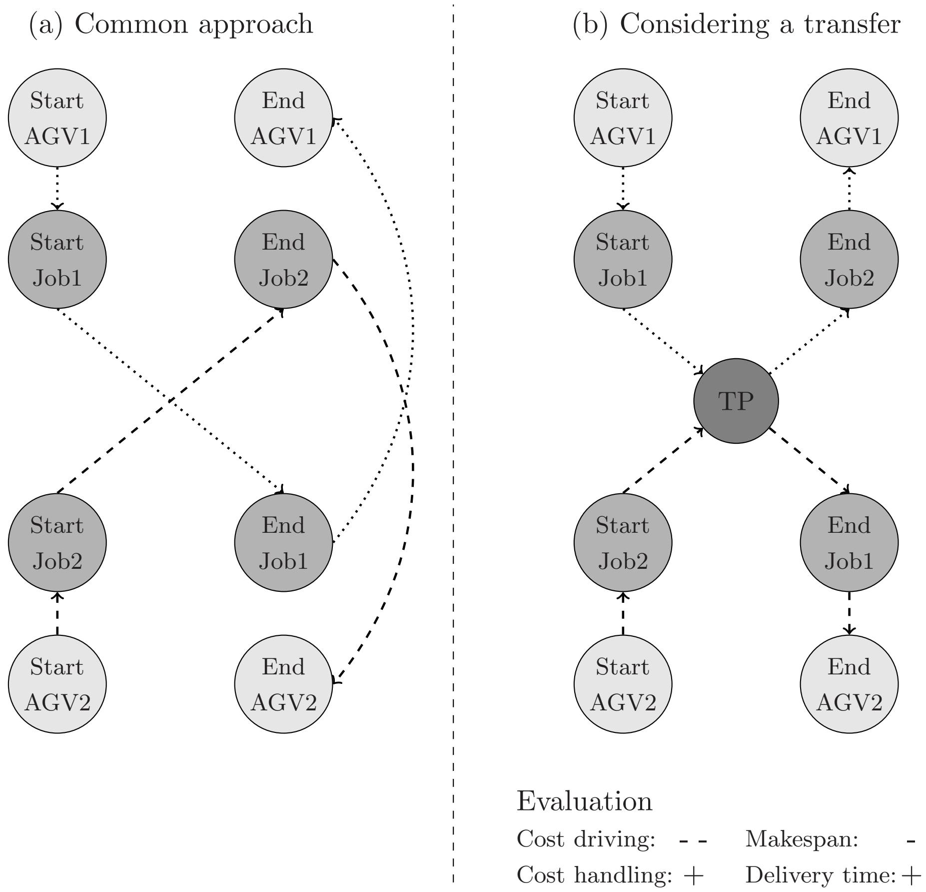

Figure 1: Comparison of a task assignment neglecting (left) and considering (rigth) a transport load transfer

is possible. The example allows a reduction in vehicle driving distance and makespan. On the other hand, it becomes clear that transfers' benefit is highly related to the concrete scenario. For instance, when vehicle handling time becomes large, a transfer operation may not lead to benefits.

Transfers of transport loads between vehicles of an AGV fleet have not yet been comprehensively investigated. Simultaneously, applications in courier services or the transport of people show promising results (see section 2.2) as studies show reduced transport distances, number of vehicles and increased logistics quality. It is still open whether the approaches can be adapted to the control of AGVS and what potentials and limitations are associated with the application in the intralogistics domain.

The objective of this research is to evaluate if transport load transfers can be considered in task assignment by a scheduling approach to improve the efficiency of AGVS. Two key issues must be considered to usefully realize the concept of transport load transfers for AGVS: (1) transfer locations must be located in the layout, (2) for task assignment, transport jobs must be divided into sub-transport jobs and allocated/assigned to the vehicles. Here, we will randomly place the transfer locations and focus on the second aspect first, as we want to show that transfers improve system performance. Optimizing the position of transfer locations will be the subject of future research and may lead to even greater benefits.

For this purpose, we examine task assigning algorithms for AGVS that allow transport load transfers. An exact and a heuristic solution method were adapted from literature and implemented. The exact method is based on a Mixed Integer Programming (MIP) model that a standard solver can process. It formally describes the task assignment task. We take an Adaptive Large Neighborhood Search (ALNS) as heuristic method to enable the calculation in a short time. We use test instances and a material flow simulation study to investigate the impact of transfers on AGVS. We will demonstrate:

- that real-time control of an AGV fleet considering transport load transfers is possible,
- that significant performance improvements can be achieved and
- that the benefit of transfers depends on the characteristics of the transport system.

The remainder of this paper is structured as follows. Chapter 2 gives an overview of the literature about task assignment for AGV and the consideration of transfers in vehicle-based transport systems. The following chapter (chapter 3) discusses relevant assumptions regarding AGVS. Chapter 4 describes the algorithms for the calculation of task assignment decisions. We present an experimental evaluation of the algorithms and general aspects concerning different characteristics of task assignment scenarios by test instances in chapter 5. A case study describes the benefits and limitations of the application of our heuristic by an exemplary system in chapter 6. Chapter 7 concludes.

2 LITERATURE

The following sections describe requirements and approaches to assign tasks in AGVS in the intralogistics domain (section 2.1) and task assignment considering transport load transfers in other domains (section 2.2). Although these systems differ significantly in their requirements, they can still demonstrate how schedules can be calculated.

2.1 Task assignment of AGVS in the intralogistics domain

The literature review in this section refers to relevant literature on the planning and control of AGVS. Typical areas of application for AGV are automated production and warehouse facilities. The automation of transports in such systems requires fleet sizes ranging from one vehicle to several hundred vehicles [see 28]. The selection and adaptation of a task assignment approach for dispatching or scheduling is highly dependent on the specific requirements of the AGV application. These requirements include the number of entities (vehicles and transport jobs), specific constraints (e.g. time windows) and objectives (e.g. short delivery times). Besides that, robustness is essential. Regardless of the number of transport jobs to be allocated to the vehicles, a valid assignment must be found. Commonly, a central instance makes control decisions with knowledge of the current state of the entire system [see 7].

2.1.1 Dispatching

In industrial applications, AGVS are typically controlled by basic dispatching algorithms. We consider dispatching as an approach to generate task assignment decisions with a short planning horizon, where just the next vehicle action (e.g. transport pick-up) is planned. Dispatching algorithms can be selected and combined to consider the specific needs of the logistics environment. Fundamental dispatching approaches are examined in Egbelu and Tanchoco [8]. They describe assigning tasks based on the nearest location to the vehicle generates performant results. A multi-attribute approach, considering multiple criteria for dispatching, is proposed by Jeong and Randhawa [11]. Ho and Chien [10] investigate dispatching approaches for multiple-load AGV (capacity of more than one transport load).

These algorithms allow fast and transparent task assignment even for large AGV fleets [e.g. up to 100 vehicles in 13]. Since we consider detailed planning covering several steps in advance as important for our investigation, we exclude these simplified task assignment approaches for our further consideration.

2.1.2 Scheduling

Metastudies show better overall system performance by applying more advanced planning techniques [see 13]. In literature, vehicle routing approaches like the Pickup and Delivery Problem (PDP) and the Dial a Ride Problem (DARP) are examined for applicability to AGVS. Algorithms such as Branch and Bound or Dynamic Programming are discussed to generate exact/proven optimal solutions. As the PDP and the DARP are considered NP-hard [see 26, 1] no algorithm can solve them efficiently. Even solving small instances can take long computing times [see 4]. Thus, in most cases, PDP and DARP are not suitable for real-time task assignment of AGV. By heuristic approaches, approximate solutions can be generated in a short time. Especially heuristics based on Local Search in combination with meta-heuristics (e.g. Tabu Search or Simulated Annealing) are proposed in the literature. A comprehensive literature study about modeling techniques and algorithms for the PDP is given by Parragh, Doerner, and Hartl [18]. Heuristics based on Local Search and Column Generation were proposed for AGV task assignment in LeAnh [13].

Related to benchmark instances (see Li & Lim), the Adaptive Large Neighborhood Search [ALNS, see 24] proved to show good results for solving the PDP. The ALNS is based on a local search procedure in which several sub-heuristics iteratively destruct and reconstruct the current solution to explore the search space. The approach is called adaptive because its performance chooses the sub-heuristics during the optimization process. A Simulated Annealing approach guides the ALNS. In Ropke [23] a parallelization of the algorithm is discussed to allow shorter calculation times. In Boden et al. [31] we describe the application of the ALNS heuristic for scheduling AGV.

2.2 Task assignment of vehicle based transport systems considering transfers

In this section, the literature overview is based on an analysis of articles on the PDP and the DARP that examine the use of transfers. Transfers have already been intensively investigated to control vehicle systems. Most of these papers are motivated by real-world problems like the transport of people (e.g. school children), distribution logistics (e.g. truck, ship and aircraft transport) [see 16, 30, 21] and crowd-sourced delivery [see 12, 25, 29]. The objectives are reduced transport costs and increased quality of logistics, e.g. through adherence to delivery deadlines. For solving the accompanied task assignment problem, authors often refer to variants of the PDP and the DARP (the Pickup and Delivery Problem with Transfers (PDP-T) and the Dial a Ride Problem with Transfers (DARP-T)). Several exact and heuristic algorithmic approaches are discussed in the literature to generate schedules for these models.

2.2.1 *Exact algorithms*

Various authors provide MIP models for the PDP-T. However, the application of exact algorithms is limited to small problem instances – even the calculation of feasible solutions is challenging [see 21].

A MIP model is given by Cortés, Matamala, and Contardo [5] and solving problem instances with up to 6 transport jobs takes up to 2 minutes when a Branch and Cut algorithm is applied.

Rais, Alvelos, and Carvalho [22] present an MIP model and provide additional formulations to integrate aspects of industrial applications like the possibility to split transport jobs. The authors generated solutions using a standard MIP solver for problem instances with up to 14 nodes (representing start/end and transfer locations). The calculation took up to 5 hours. Their model was adapted by Sampaio et al. [25] to investigate crowd-sourced delivery services and by Shiri, Rahmani, and Bafruei [27] to evaluate the effect of transfers considering general system characteristics of vehicle based transport systems.

A mathematical model for the DARP-T is given by Pierotti and Essen [20]. With a time limit of 1 hour, they were able to generate solutions with up to 4 vehicles and 9 transport requests.

2.2.2 *Heuristics*

In addition to exact/analytical models, heuristic approaches are quite popular and subject of literature to generate solutions considering transfers in a reduced amount of time. Mitrović-Minić and Laporte [16] and Oertel [17] provide Local Search approaches. Mitrović-Minić and Laporte [16] investigate the influence of the instance characteristics on the effect of transfers for courier services. They randomly generated problem instances with a uniform distribution of pick-up and delivery points. Also, variations in time windows constraints (hard/soft), service times at transfer nodes and transfer point positions were considered. They quantified an improvement of up to 10 % in total route length for randomly distributed instances. Pointing out that time window size and the number of jobs have a high effect on the benefit of transfers. Their findings suggest that the characteristics of the transport system may also be relevant for the use of AGV. The heuristic presented by Oertel is guided by a Tabu Search meta-heuristic. It starts with creating a feasible solution. Afterward, this schedule is improved by local exchange steps. The author demonstrates improvements on real-world instances for truck transports with up to 16 trucks in computation times between 5 and 20 minutes. Coltin and Veloso [3] solved the PDP-T by a Very Large Neighborhood Search Approach (VLNS). Transport loads are allowed to be transferred multiple times. Each location can be considered as a transfer point and there are no capacity constraints for transfers. The VLNS starts with an initial solution that is generated greedily, taking random transfers into account. Starting from this solution, a local search procedure is carried out by iteratively removing and reinserting transport jobs. This approach allows computing schedules with up to 4 vehicles and 36 transport jobs in less than 30 minutes. It is demonstrated that schedules considering transfers improve schedules neglecting transfers.

ALNS approaches are given by Sampaio et al. [25] and Masson, Lehuédé, and Péton [15]. These algorithms are guided by a Simulated Annealing meta-heuristic and terminate after a maximum number of iterations. Sampaio et al. demonstrate that solutions for problem instances with up to 100 transport jobs can be found in less than 6 minutes. Qu and Bard [21] extend the ALNS approach by a Greedy Randomized Adaptive Search Procedure. Various sub-heuristics are employed to generate neighborhoods for the local search procedure. The heuristics applied range from fast to time-consuming to be calculated. None of them could be shown as dominant so far. A parallel ALNS is provided by Petersen and Ropke [19] to decrease the time of computation. Each thread performs the local search procedure on a local copy. They can generate solutions for problem instances with up to 1000 transport jobs with a time-limit for computation of 90 minutes.

In Danloup, Allaoui, and Goncalves [6] a genetic algorithm by different functions for crossover and mutation is given. A crossover exchanges sequences of routes between vehicles while mutations manipulate single operations of the schedule. The authors could achieve comparable results compared to the ALNS presented in Masson, Lehuédé, and Péton [15]. The ALNS heuristic is still in focus and serves as a basis for latest scientific publications about this application area [compare to 9, 30].

The result of the evaluated literature shows that the ALNS heuristic is used most frequently and is also used as a benchmark for assessing other algorithms. So far, algorithms (exact and heuristic) with a long computation time are applied, which can be used to solve offline problems. Thus, they can not be applied for real-time control of an AGV system in that configuration. However, it is conceivable that the ALNS heuristic can be used for systems with a limited number of vehicles since meta-heuristics in general and also the ALNS are already investigated to control AGVS without transfer consideration.

2.3 Conclusion

As a result of this literature search, we found that splitting transport jobs via transfers for AGV has not yet been considered in publications focusing to the intralogistics domain. Therefore, it remains open whether and to which extent the advantages reported from other domains (see introduction and section 2.2) can also be realized in intralogistics. However, it can reasonably be assumed that the transfer of transport loads can achieve synergy effects concerning other vehicles' routing or parking positions in order to achieve better performance of an AGV system. For schedule generation, the most commonly applied methods are a solution based on a mathematical model using a solver and the ALNS heuristic. The following sections investigate the application of these techniques to dynamic planning scenarios in the AGV use case.

3 CONCEPT OF TRANSPORT LOAD TRANSFERS IN AGVS

This chapter introduces the concept of transport load transfers. Besides executing a transfer operation, this involves premises about the planning scenario and an exemplary concept for a technical realization. These specifications are applied to the task assignment algorithms and experiments in the following chapters (chapter 4 to 6).

3.1 Concept

Transport load transfer operations are planned considering the actual system status like vehicle positions, running transport tasks and available transfer locations by a central control unit that assigns the transport jobs to the vehicles. To execute a transfer operation, a vehicle is directed to a predefined transfer location which buffers the transport load. Subsequently, a receiving vehicle continues the transport. Transport jobs can be divided into any number of sub-transport jobs. This could help reduce vehicle tour lengths by enabling synergy effects, e.g. combining transports with similar destinations. However, assigning transport jobs to vehicles becomes more complex, as there are far more options for the transport job assignment.

3.2 Assumptions

We focus on AGVS usually applied in automated warehouses and production facilities. To enable the modeling of AGVS in a generic manner rather than specific to individual applications, we consider the following assumptions.

Transport jobs are defined by a combination of a pick-up and a corresponding drop-off location. Pick-ups and drop-offs must be done in specific time windows, whereas a pick-up must be executed first. The transport of a transport load requires capacity on the vehicle. In order to execute transport load transfers, a transport job can be divided into any number of sub-transport jobs. We describe vehicles by the parameters velocity, handling time and capacity. Each vehicle can have individual characteristics (heterogeneous fleet) and starts and ends its tour at individual start and end locations. The vehicles are not restricted to specific transport jobs.

We suppose that scheduling is done by a central control unit based on all relevant information. Therefore, continuous communication to all subsystems is assumed. The objective is to minimize operation time for vehicle driving and handling (in the following, referred to as costs) to reduce vehicle utilization. We consider this as a sensitive indicator to detect even minor effects. Reducing cost leads to increased efficiency in the processing of transport jobs, potentially enabling higher throughput or a reduction in the number of vehicles. All transport jobs need to be considered to generate a valid schedule.

Transfers are allowed at predefined transfer locations. They can be visited multiple times by a vehicle and simultaneously by several vehicles. Transfer locations are not capacitated and there are no precedence constraints for access to transport loads. A transport load is buffered at the transfer location to execute a transfer and then picked up again. Hence, a direct meeting of vehicles to exchange transport loads for transfer is not required. Transfer operations are associated with a handling time and the delivering vehicle must visit the transfer location before the receiving vehicle.

3.3 Algorithmic approaches for task assignment

Roughly, there are three types of algorithmic approaches for AGV task assignment applied:

1. Detailed schedules generated by exact/proven optimal algorithms,
2. detailed schedules generated by heuristic algorithms and
3. selection of the next task to be processed based on criteria that allow fast evaluation.

There are approaches that generate a plan with all outstanding transport tasks (see 1. and 2.) and that select only the next task to execute (see 3.). The solution techniques presented in section 2.2, are type of the first two categories. So far, no publications could be identified for category 3.

Exact solution techniques (e.g. Branch and Bound) and meta-heuristics (e.g. Simulated Annealing) are mainly used for the first two categories. They allow high solution quality with the downside of high and rapidly growing computation efforts for bigger problem sizes. Hence, real-time control with these approaches is possible solely for systems with few vehicles and assigned transport jobs.

Algorithms of the third category generate reasonable results based on criteria fast to evaluate, which potentially allows fast calculation of schedules for larger AGVS.

Because we want to evaluate the potential as accurately as possible, we will focus on the first two categories. We will evaluate algorithms that generate the best possible overall schedules with an extended planning horizon and thus use transfers most beneficially. Therefore, we will use a limited amount of time as the termination criterium for the algorithms for the real-time application. This configuration will allow a new schedule calculation whenever a significant change in the planning conditions happens (e.g. a new transport job appears) to realize online task assignment for AGVS.

The results regarding the benefits of transfers should always be seen in the context of the algorithmic approach used. When using a heuristic, it is conceivable that the approach systematically neglects beneficial transfers. Nevertheless, it is possible to show that transfers can be identified that lead to benefits regardless of whether the optimal value is achieved.

3.4 Technical realization and application

An example of an AGV system allowing transfers is shown in figure 2. The AGV can carry multiple loads as it has two buffer places. It is equipped with a robot arm to handle transport loads independently and needs no further infrastructure. A transfer station could be set up by a usual shelf like it is shown in the picture. This would buffer multiple transport loads and do not need precedence constraints (e.g. last in – first out) for pick-up.

Figure 2: Multiple-load AGV equipped with a robot arm handling at a shelf in a semiconductor production facility (reference: www.fabmatics.com)

Overall, the concept and the mathematical model represent the basic features of task assignment for AGV. They allow the assignment of transport tasks to the vehicles of an AGVS, which can be parameterized according to the intralogistics domain (e.g. vehicle capacity). For application in a dynamic system, schedules need to be generated continuously by a suitable solution technique.

The concept requires evaluation to determine if schedules can be computed fast enough and with adequate quality, and if transfers lead to an improvement. Furthermore, it is important to evaluate whether individual assumptions of the concept (e.g., that transport orders can be split multiple times or the use of statically defined transfer points) are reasonable.

4 TASK ASSIGNMENT ALGORITHMS

This chapter describes two approaches to generate schedules considering transport load transfers. The first approach bases on a mathematical model which formally describes the planning problem and allows optimal calculating solutions, e.g. by a standard solver (see section 4.1). With the intention of less computational efforts compared to solving exact mathematical models, the following section examines the heuristic ALNS (see section 4.2). In addition, in section 4.3 a standard rule-based dispatching algorithm is given for reference.

4.1 Mathematical model

The model presented here bases on the work of Rais, Alvelos, and Carvalho [22]. The authors proposed a base model and additional equations to customize it to a wide range of application scenarios. We selected and adapted the equations relevant to the AGV use case. The adaptations mainly concern the objective function to minimize the operation time for schedule execution, the consideration of vehicle-specific service times for material handling operations, and a more fundamental constraint to eliminate sub-tours. In addition, our model forces vehicles that are not used for transportation tasks to leave their origin node to travel to their specified end position.

The mathematical model is stated as follows: $G(N, A)$ defines a complete weighted directed graph containing a set of nodes $i, j \in N$ and corresponding arcs $ij \in A$. Transport jobs (requests $r \in R$) are specified by a pick-up node $p(r) \in N$ and a drop-off node $d(r) \in N$. These nodes demand service in the time windows $[a_{p(r)}, b_{p(r)}]$, $[a_{d(r)}, b_{d(r)}]$. Constant a determines the earliest possible time to service a node and b the latest. Transfer points to exchange transport carriers are given by the subset $T \subseteq N$. Let K be the set of vehicles. Each of the vehicles $k \in K$ starts and ends at the corresponding nodes $o(k)$, $o'(k) \in N$. Vehicles are constraint by capacity of u_k. Let d_{kij} be the cost (measured in time) for vehicle $k \in K$ to drive from node $i \in N$ to node $j \in N$. Service times for transport load handling are given by s_{ki} as amount of time the vehicle spends at the node. Constant q_r defines the amount of capacity it needs to transport a request $r \in R$.

Several decision variables track requests and vehicles. The binary decision variable x_{kij} indicates that vehicle $k \in K$ moves on the arc $ij \in A$ ($x_{kij} = 1$, else $x_{kij} = 0$). The variable y_{krij} equals 1 in case vehicle $k \in K$ moves request $r \in R$ between the nodes $i, j \in N$. It becomes 0 otherwise. A transfer of request $r \in R$ at a node $j \in T$, between vehicles $k, l \in K$ is indicated by the binary decision variable z_{kljr}. It is 1 if a transfer takes place and 0 otherwise. The decision variables t_{ki} and t^{-}_{ki} are related to the date when a vehicle $k \in K$ enters or leaves node $i \in N$.

Objective:

$$\text{Minimize: } \sum_{k\in K}\sum_{i\in N}\sum_{j\in N}(d_{kij}+s_{kj})x_{kij} \tag{1}$$

Subject to:

$$\sum_{j\in N} x_{kij} = 1, \quad \forall k \in K, \forall i = o(k) \tag{2}$$

$$\sum_{j\in N} x_{kij} = \sum_{j\in N} x_{kjl}, \quad \forall k \in K, \forall i = o(k), \quad \forall l = o'(k) \tag{3}$$

$$\sum_{j\in N} x_{kij} - \sum_{j\in N} x_{kji} = 0, \quad \forall k \in K, \quad \forall i \in N \setminus \{o(k), o'(k)\} \tag{4}$$

$$\sum_{k\in K}\sum_{j\in N} y_{krij} = 1, \quad \forall i \in p(r), \forall r \in R \tag{5}$$

$$\sum_{k\in K}\sum_{j\in N} y_{krji} = 1, \quad \forall i \in d(r), \forall r \in R \tag{6}$$

$$\sum_{k\in K}\sum_{j\in N} y_{krij} - \sum_{k\in K}\sum_{j\in N} y_{krij} = 0, \quad \forall i \in T, \forall r \in R \tag{7}$$

$$\sum_{j\in N} y_{krij} - \sum_{j\in N} y_{krji} = 0, \quad \forall k \in K, \quad \forall i \in N \setminus \{T, p(r), d(r)\}, \forall r \in R \tag{8}$$

$$y_{krij} \le x_{kij}, \quad \forall k \in K, \forall i \in N, \forall j \in N, \forall r \in R \tag{9}$$

$$\sum_{r\in R} q_r y_{krij} \le u_k x_{kij}, \quad \forall k \in K, \forall i \in N, \forall j \in N \tag{10}$$

$$t^{-}_{ki} + d_{kij} - t_{kj} \le M(1 - x_{kij}), \quad \forall k \in K, \forall i \in N, \quad \forall j \in N \tag{11}$$

$$t_{ki} + s_{ki} \le t^{-}_{ki}, \quad \forall k \in K, \quad \forall i \in N \tag{12}$$

$$a_{p(r)} \le t_{kp(r)}, \quad t^{-}_{kp(r)} \le b_{p(r)}, \quad \forall k \in K, \forall r \in R \tag{13}$$

$$a_{d(r)} \le t_{kd(r)}, \quad t^{-}_{kd(r)} \le b_{d(r)}, \quad \forall k \in K, \forall r \in R \tag{14}$$

$$t_{ki} = 0, \quad \forall k \in K, \forall i = o(k) \tag{15}$$

$$\sum_{j\in N} y_{krji} + \sum_{j\in N} y_{lrij} \le z_{klir} + 1, \quad \forall k \in K, \quad \forall l \in K, k \neq l, \forall i \in T, \forall r \in R \tag{16}$$

$$t^{-}_{ki} - t^{-}_{li} \le M(1 - z_{kljr}), \quad \forall k \in K, \forall l \in K, k \neq l, \quad \forall i \in T, \forall r \in R \tag{17}$$

$$\sum_{i \in S} \sum_{j \in S} x_{kij} <= |S| - 1, \ \ \forall k \in K, \ \ \forall S \subseteq N, \ \ |S| >= 2 \tag{18}$$

$$x_{kij} \in \{0,1\}, \ \ \forall k \in K, \forall i \in N, \forall j \in N \tag{19}$$

$$y_{krij} \in \{0,1\}, \ \ \forall k \in K, \forall i \in N, \forall j \in N, \forall r \in R \tag{20}$$

$$z_{kljr} \in \{0,1\}, \ \ \forall k \in K, \forall l \in K, \forall i \in N, j \in N, \ \ \forall r \in R \tag{21}$$

$$t_{ki} \in \mathbb{R}, \ \ \forall k \in K, \forall i \in N \tag{22}$$

$$t^{-}_{ki} \in \mathbb{R}, \ \ \forall k \in K, \forall i \in N \tag{23}$$

The objective (equation 1) minimizes the costs (measured in time) for vehicle movement between nodes (d) and corresponding service for transport load handling (s). It is subject to vehicle and request-specific constraints. In detail, constraint 2 ensures the vehicle k leaves its node of origin o, whereas constraint 3 ensures that each vehicle end the tour at the designated final destination o'. A flow conservation constraint supplements these two constraints (see constraint 4). Constraints 5 and 6 make sure that each request will be picked up and dropped off at the corresponding nodes (p and d). Flow conservation is controlled by the following two constraints, whereby constraint 7 allows the request to be transported by several vehicles at transfer nodes whereas constraint 8 enforces the request to be transported just by one vehicle at other nodes. Synchronization between y_{krij} and x_{kij} is ensured by constraint 9. Constraint 10 ensures that the vehicle's capacity is not exceeded. Constraints 11, 12 and 15 generates timestamps when a vehicle enters and leaves the nodes. Constraints 13 and 14 keep track of complying with time windows for pick-up and drop-off requests. Constraint 16 and 17 ensure the precedence of the vehicles for load transfer. The vehicle dropping off a load must complete service at the transfer point before a pick-up vehicle starts service. Constraint 18 ensures sub-tour elimination.

The model restricts the vehicles to visit a transfer node once only. Redundant transfer nodes must be defined at the same location to allow multiple visits. As there are no capacity constraints for the transfer points, it is assumed that there is sufficient capacity to handle all necessary transfers. The proposed model takes both cases into account: heterogeneous or homogeneous vehicle fleets. Since independent constants characterize each vehicle e.g. in terms of capacity and costs (for driving and handling).

4.2 Adaptive Large Neighborhood Search

This section describes an ALNS heuristic to generate solutions for the optimization problem presented in section 4.1. The ALNS was proposed for the PDP by Ropke and Pisinger [24], an extension for parallel computing is described in Ropke [23]. Fundamentals regarding the integration of transfers to the ALNS are discussed in Masson, Lehuédé, and Péton [15] and Sampaio et al. [25].

The optimization process applying the ALNS for considering transfers in two phases is shown in figure 3. Phase one generates an optimized PDP solution without any transfers. In the second phase, transfers are considered to solve the PDP-T. The idea is to start evaluating transfers from a good schedule and improve it. Since considering transfers is computationally intensive, this can help to make optimized planning decisions in dynamic environments (in short time). The disadvantage is that opportunities for transfers may be missed because the search is initially limited. Both phases of the optimization process are terminated by a time limit. It is part of the time budget of Phase 1 to generate the initial schedule. Phase 1 is skipped if it is already reached after the Greedy Insert Procedure. Furthermore, when the time limit for Phase 2 is 0 s, this step is also skipped. Consequently, no transfers will be evaluated. In the following, we use ALNS-wt when transfers are considered in the optimization process and ALNS-wot if they are not.

Algorithm 1 describes the ALNS. It requires an initial schedule sc. The heuristic is terminated by time ($time - limit$, see line 3). A Simulated Annealing acceptance criteria is applied to accept schedules ($accept$() function, see line 19) with higher costs for further investigation and allows higher diversity in exploring the search space. As optimization progresses, worse schedules are less likely to be accepted. Until the stop criterion is not met, the currently accepted solution is modified by sub-heuristics to remove a number of transport jobs q from the schedule to reinsert them iteratively (see lines 8 to 13).

Each parallel thread starts with a local copy of the currently accepted schedule and selects the number of transport jobs and the combination of sub-heuristics independently of the other threads. Sub-heuristics are selected by their corresponding weights and a roulette wheel selection. Each thread generates a schedule and appends it to the set of all schedules SC' of the current iteration. When all threads are finished, these schedules are evaluated to accept for the next iteration. For cost evaluation (*evaluation*, line 17), time window constraints are relaxed. A factor will penalize violations. After a specified number of iterations ($weight-update-interval$, see line 22), the performance of the sub-heuristics is evaluated. That leads to higher weights for sub-heuristics that could find new solutions or the overall best solutions.

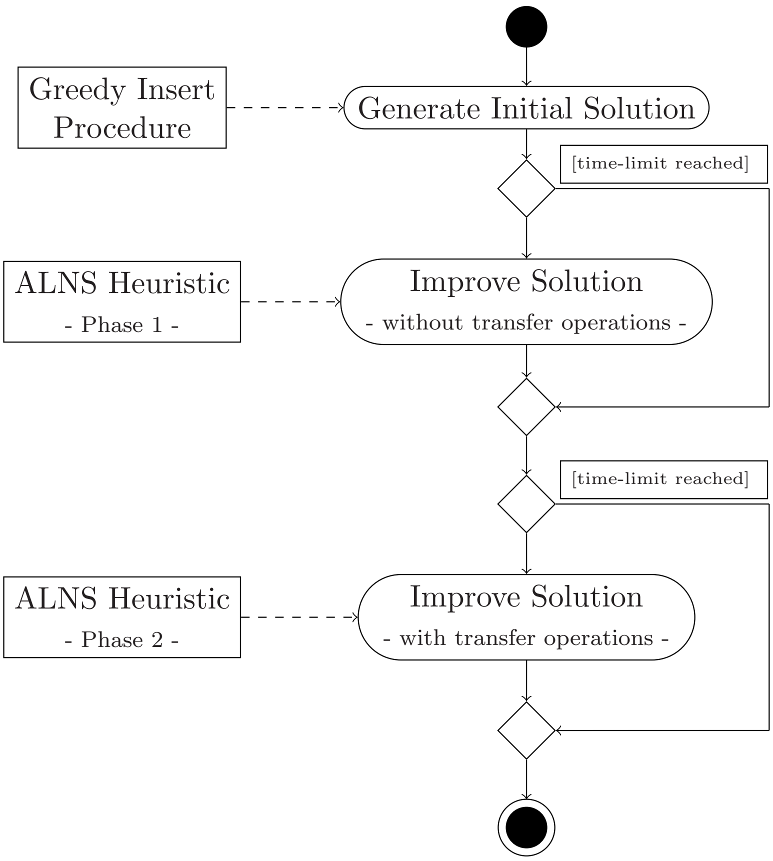

Figure 3: Optimization process: creating an initial schedule and applying the ALNS in two phases. The second phase is skipped if transfers are not considered (ALNS-wot).

Algorithm 1 ALNS heuristic

```
1:  function ALNS(initial solution sc)
2:      solution : sc_best = sc
3:      while time − limit not met do
4:          number running threads tr = 0
5:          empty set of solutions SC′
6:          while thread − limit not met do
7:              Start parallel thread:
8:                  sc′ = sc
9:                  select amount of transport jobs to remove q
10:                 remove q transport jobs from sc′
11:                 determine if transfers should be considererd by pt
12:                 insert removed transport jobs into sc′
13:                 add sc′ to SC′
14:             tr = tr + 1
15:         wait until parallel threads are finished
16:         for sc′ ∈ SC′ do
17:             if evaluation(sc′) < evaluation(sc_best) then
18:                 sc_best = sc′
19:             if accept(sc′, sc) == true then
20:                 sc = sc′
21:         evaluation of sub-heuristics
22:         if weight − update − interval met then
23:             update sub-heuristic weights
```

The key element of an ALNS are the sub-heuristics applied for neighborhood generation. They can be differentiated between removal sub-heuristics that remove transport jobs from the schedule and insertion sub-heuristics that reinsert the removed transport jobs. Transport job removal is carried out in one iteration based on the evaluation of the schedule. Transport job reinsertion is applied iteratively. The following sub-heuristics are implemented to remove transport jobs:

- random: selects transport jobs to remove randomly
- longest transport time: selects transport jobs with the highest time difference between the pick-up and drop-off
- longest waiting time: selects transport jobs with the highest time until pick-up
- longest delivery time: selects transport jobs with the highest time until drop-off
- complete route: selects all transport jobs from a randomly chosen vehicle

These sub-heuristics are fast to compute since transport time, waiting time and delivery time are maintained due to schedule evaluation and the random selection of transport jobs or a complete route does not take much effort. The selection of sub-heuristics follows the idea that transport jobs with long waiting times and long transport or delivery times benefit especially from a new insert position. Transport job insertion is implemented by the following sub-heuristics:

- random: selects the next transport job randomly
- best cost position: selects the transport job that will least increase the cost of the schedule
- low cost position: selects one of the transport jobs that will least increase the cost of the schedule
- regret: selects the transport job with the highest cost increase when it is not inserted at the best cost position (e.g. in comparison to the second-best cost position)

To select one of the pending transport jobs for insertion, the cost of all possible insert positions in the current schedule is calculated for each transport job. The sub-heuristic application bases on this set of evaluated schedules. After a transport job is selected for insertion, it will be inserted at its best position regarding cost.

In the second phase of the optimization process (only relevant for ALNS-wt, compare figure 3), the heuristic randomly decides whether transfers should be considered or not for each thread by *pt* (probability of transfer consideration, see line 11). This decision applies to all transport jobs to be inserted. If a transfer needs to be considered, the transport jobs are split into two parts – a pick-up tour ending at the transfer point and a delivery tour starting from the transfer point. The transfer location is selected randomly from all transfer locations, preferring those resulting in short detours compared to the direct delivery. Transport jobs that are split due to transfers can be split again in further ALNS iterations.

4.3 Standard dispatching approach

A basic algorithm was selected following Ho and Chien [10] to compare the results of the algorithmic approaches under consideration of transport load transfers to common AGV task assignment in the intralogistics domain. Algorithm 2 describes the procedure.

Algorithm 2 Standard dispatching approach

1: **function** DISPATCH(set of transport jobs R)
2:     empty set of running transport jobs R_{run}
3:     empty schedule sc
4:     **for** $k \in K$ **do**
5:         append start location $o(k)$ to sc at route of k
6:     **while** $|R| > 0$ **do**
7:         select next available vehicle k
8:         **if** k has free capacity **then**
9:             select transport job r with shortest distance from k to pick-up location
10:             append pick-up operation of r to sc in route of k
11:             remove r from R and insert r to R_{run}
12:         **else**
13:             select transport job r with shortest distance to drop-off location
14:             append drop-off operation of r to sc in route of k
15:             remove r from R_{run}
16:     **while** $|R_{run}| > 0$ **do**
17:         select a vehicle k that has unfininshed transports
18:         select transport job r with shortest distance from k to drop-off location
19:         append drop-off operation of r to sc in route of k
20:         remove r from R_{run}
21:     **for** $k \in K$ **do**
22:         append end location $o'(k)$ to sc at route of k

The idea is to prefer near-located pick-up tasks as long as there is free capacity on the vehicle and new transport jobs are available [see 10]. Starting from an empty schedule, sc for each vehicle $k \in K$, an empty route containing the start position $o(k)$ is generated (see lines 4-5). Next, the algorithm iteratively assigns pick-up and delivery tasks to the schedule (see lines 6 to 15). The vehicle with the shortest planning horizon (assuming this is the next available vehicle) is selected in each iteration. In cases where several vehicles are available simultaneously, one of these vehicles is randomly selected. If the vehicle has a remaining capacity to pick-up one of the pending transport jobs (request $r \in R$), the one with the closest pick-up location in respect to the last scheduled vehicle location will be selected. If there is no free capacity, the vehicle will be sent to the nearest location to drop-off. When all transport jobs are scheduled, the vehicle end locations $o'(k)$ are appended (see lines 21 to 22). The costs for all vehicle movements and service times are calculated for evaluation. For simplification, time window constraints are neglected.

With this dispatching approach, transparent control decisions for large vehicle fleets can be calculated fast (compare section 2.1). The solution quality is limited due to the short planning horizon, as only the next step for a vehicle is considered and time windows are neglected. The algorithm prefers pick-up operations before drop-off. This leads to a high use of vehicle capacity and inefficiencies because effective drop-off opportunities might be missed. However, since benchmarks for the application area are missing, this algorithm serves for comparison of the ALNS with and without transfers (wt and wot) to a common task assignment technique for AGVS.

4.4 Implementation

The ALNS (wt and wot) and the standard dispatching approach are implemented in C++. The mathematical model was built with the help of the Python package Pyomo, while the solver CPLEX 12.9.0 was used for solving. All experiments are executed on a desktop PC using an Intel i7-4770 CPU with 3.40 GHz on 8 logical kernels.

5 ALGORITHM EVALUATION BY TEST INSTANCES

This chapter investigates test instances to assess the task assignment algorithms and discusses the effect of possible transport load transfers on AGVS' performance. Against this background, we evaluate generic parameters focusing on the influence of the systems' size and hence the size of the problem.

The chapter starts with an introductory example of a planning problem that was solved considering a transport load transfer operation (see section 5.1). It illustrates the setting of the experiments and explains the procedure of a transfer operation. In order to get confident results from a statistical point of view, random problem instances are needed. So, section 5.2 introduces an appropriate generator. Section 5.3 evaluates and compares the algorithmic approaches introduced in chapter 4. We demonstrate the influence of relevant AGV system parameters on the benefit of transport load transfers by the results of a full factorial experiment (see section 5.4).

5.1 Basic example

A test instance and the corresponding schedule are presented in figure 4. The vehicles can move directly between these points (euclidian distance). The concrete parameter values are listed in the appendix in table 6. The underlying scheduling problem is represented by a set of two vehicles (k_0 and k_1), two transport jobs (r_0 and r_1) and four transfer locations (t_0 to t_3). The vehicle's routes start and end at the corresponding locations S and E. The transport jobs start and end at the job-specific pick-up points P and end at the drop-off points D.

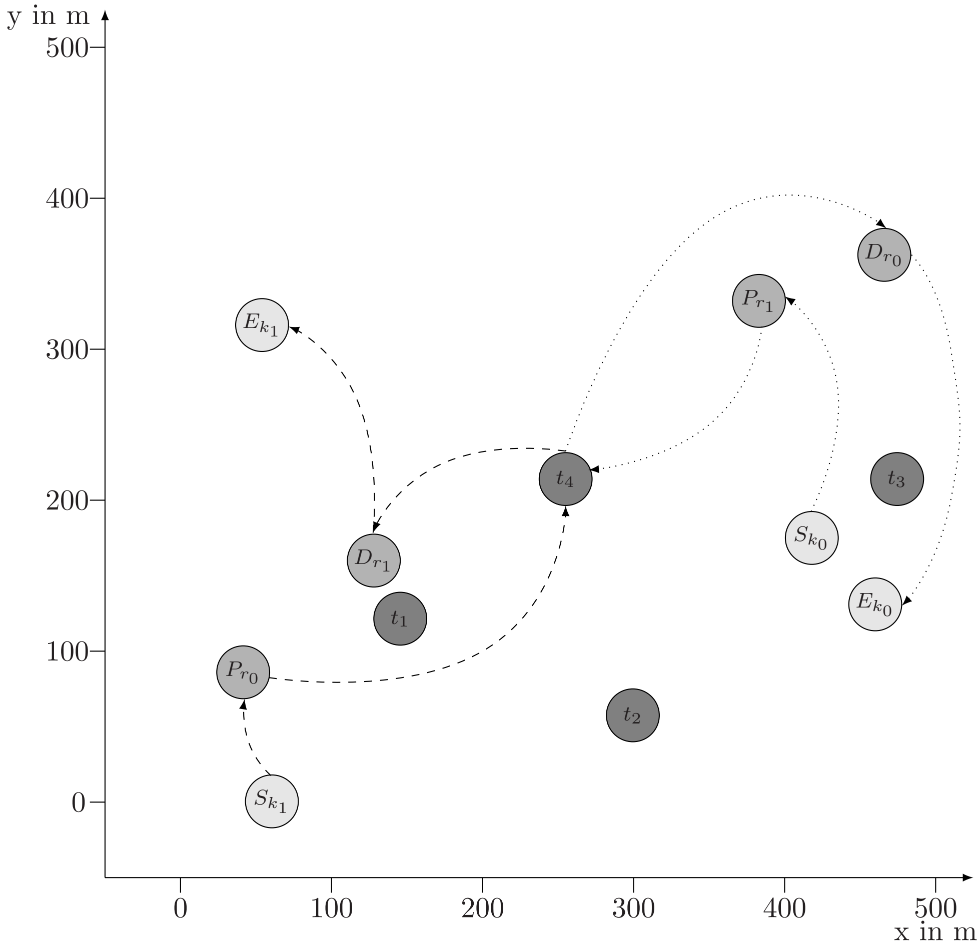

Figure 4: Solved problem instance considering a transport load transfer (note: arcs are just for visualization purpose)

In detail for the example shown in figure 4: vehicle k_1 picks up transport r_0 at $P_{r_0,}$ deposit it at t_4, picks r_1 from t_4, to drop it at location D_{r_1} and returns to the selected end location. Vehicle k_0 does the remaining transport sub-tasks. It starts at location $S_{k_0,}$ picks up r_1 at $P_{r_1,}$ deposit the load at t_4 and picks up r_0 to deliver the job at D_{r_0} and ends at $E_{k_0.}$ It considers that the delivery sub-tasks are started after the drop-off of the corresponding transport load at the transfer point.

Considering the case without transfers (k_0 transports r_1 and vehicle k_1 transports r_0), the cost function/objective (driving distance and handling/service times, see equation 1) reduces by 18 %. Even though the sum of delivery times increases by 11 %. The result is proven optimal by the exact solution approach by a standard solver. This example shows that transport load transfers can improve the objective.

A transport load transfer execution leads to additional effort for detours to reach the transfer location and material handling. In contrast, transfers allow to reduce vehicle tours and meet time windows. As already stated, we assume that the benefit of transfers relies on systems' characteristics. Which of them are of particular importance is the subject of section 5.4.

5.2 Generation of test instances

To generate the test instances, we vary a set of parameters characterizing planning problems for the task assignment of AGV fleets in a generic way. They refer to the characteristic of vehicles, transport tasks and transfer points. In detail:

- transport characteristic
  - number of transport jobs (ntj)
  - min length of transport jobs (mltj)
  - factor time windows (ftw)
- transfer characteristic
  - number of transfer points (ntp)
  - position of transfer point (ptp)
- vehicle characteristic
  - velocity (v)
  - handling time (ht)
  - capacity (c)
  - number vehicles (nv)
  - fleet characteristic (fc)

The number of transport jobs (ntj) determines how many transport tasks the scheduling algorithm needs to consider. Parameter (mltj) defines each transport task's minimum length based on the shortest distance between the start and end location. For simplification, we assume that all transport tasks are ready to be picked up at the beginning of the scheduling process and that there is the latest drop-off timestamp that is the same for all transport tasks. We generate this timestamp by summing up the time the slowest vehicle needs to go from start to end for each transport task. This value we multiply by a time window factor (ftw). If this factor is less than 1, the time window is reduced.

We vary the characteristic of transfer locations by selecting a number of transfer points (ntp) and by determining a strategy for transfer location placement (ptp) (randomly or centrally defined by a square from the middle of the layout with a side length of 25 % of the layout x-axis length).

Vehicles are characterized by capacity (c), velocity (v) and handling time (ht). We determine the vehicle fleet size by the parameter number of vehicles (nv). If a fleet is homogenous (identical vehicle parameters) or heterogeneous (varying vehicle parameters) is controlled by the parameter fleet characteristic (fc).

We create a spatial representation of the problem instance based on these parameters. Therefore, locations for vehicle start and end stations, transport job start and end stations and transfer stations are defined. They are randomly placed in an area with a 500 m * 500 m dimension. The vehicles pass the euclidian distance to drive from one location to another.

5.3 Discussion of the algorithmic approaches

This section evaluates the essential characteristics of the implemented ALNS heuristic. We focus on the configuration with transfers (ALNS-wt) and compare it to the setting without transfers (ALNS-wot). In addition, since no standard benchmark problem allows evaluating the algorithm, we compare its capabilities to an exact solution by a standard solver (see section 4.1) and a standard AGV dispatching approach (see section 4.3). The evaluation take test instances into account described in the previous sections 5.1 and 5.2. Therefore the basic parameters number of transport jobs (ntj) and number of vehicles (nv) are varied in the specified limits as shown in table 1. According to the literature, the model's exact solution generation is highly sensitive to problem size. We created test instances (100 each) ranging from ntj = 4 and nv = 2 up to ntj = 16 and nv = 8.

Table 1: Set of test instances as reference for algorithm evaluation

ntj	nv	ntp	# of instances
4	2	4	100
8	4	4	100
12	6	4	100
16	8	4	100

5.3.1 Configuration of the ALNS

We want to evaluate the real-time decision-making capabilities of the ALNS-wt heuristic to ensure its applicability for AGVS. Accordingly, the following parameters need to be configured (compare section 4.2 and algorithm 1):

- general for ALNS-wot and ALNS-wt
  - *time − limit* phase 1
  - threshold *accept*() function (Simulated Annealing)
  - percentage of jobs to remove q for neighborhood generation
  - *weight − update − interval* for sub-heuristic evaluation
  - *thread − limit* for parallel processing

- specific for ALNS-wt
  - *time − limit* phase 2
  - probability of transfer consideration pt

From systematic tests, we achieved proper results (regarding the algorithm's capabilities to create meaningful transfers) by the following configurations. We set the threshold level for the integrated Simulated Annealing acceptance criteria *accept*() to 75 %. Therefore, schedules are only investigated as a basis for the next iteration if their objective is slightly worse than the actual best-known solution. This limits the diversification of the search procedure to generate good results in short time. 50 % of transport jobs can be removed (q) from the schedule to create new neighborhoods to enable the heuristic to change a substantial part of the schedule. Therefore, we can test scenarios in which the simultaneous transfer of several transport loads is necessary to achieve an improvement. We set the number of iterations between sub-heuristics performance evaluation and weight update (*weight−update−interval*) to 100 and the number of parallel threads (*thread − limit*) to 4.

We specify a *time − limit* of 30 s for solution calculation, spending 1/3 of the calculation effort for phase 1 and 2/3 for phase 2 (see figure 3). Starting from an initial solution, phase 1 creates an optimized schedule neglecting transfers. Applying the algorithm of phase 1 is crucial as it turned out as a basis for finding beneficial transfer options in phase 2 of the ALNS-wt. For phase 2, we configured the algorithm's probability to investigate transfers for transport job insertion (pt) to 50 %. With these parameter settings, a schedule with 8 transport jobs allows the removal of up to 4 transport jobs and on average half of them are investigated for transfers.

A test instance is solved by ALNS-wt and by the ALNS-wot. A transfer is only accepted for evaluation in the following if the objective improves the solution.

5.3.2 *Comparison between ALNS-wt and ALNS-wot*

First, we investigate whether the ALNS-wt heuristic can identify solutions that potentially lead to benefits for the overall system. Therefore, the schedule should contain a transfer operation and the objective should be lower compared to the schedule generated by the ALNS-wot.

The experiment with the ALNS-wt resulted in the identification of 31 instances containing at least one transfer operation in 400 defined test instances (see table 1). From these, for 20 instances, the objective value of the ALNS without transfers could be improved. The average improvement in cost (for driving and handling) was 1.0 % and 7.3 % in maximum. ALNS-wt performed 0.7 % worse on average than ALNS-wot, resulting from a resource conflict: identifying transfers is computationally expensive. Obviously, it is more efficient to find schedules without transfers in the present case and its problem-solving time restrictions. Easy to imagine, the effect becomes more relevant with rising problem sizes, as a higher number of vehicles,

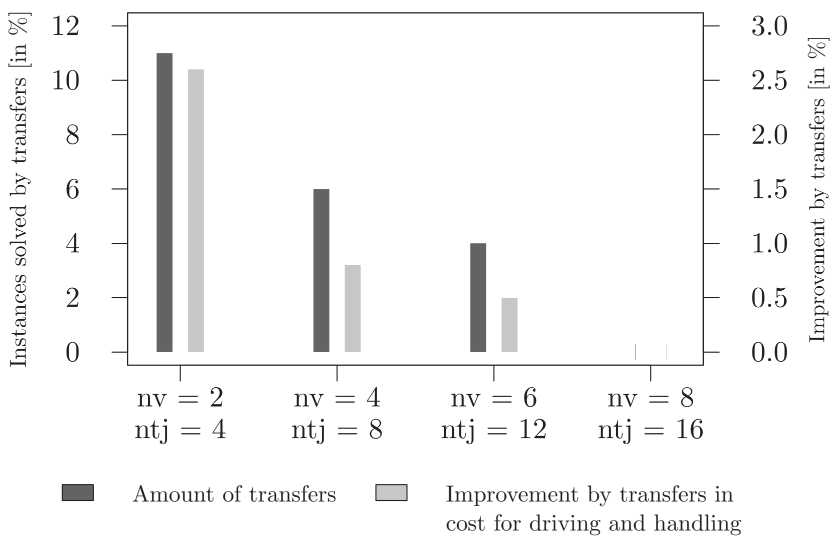

Figure 5: Percentage of test instances solved with at least one transfer operation and average improvement compared to the solution neglecting transfers (from ALNS-wot) concerning problem size for a time-limit of 30 s

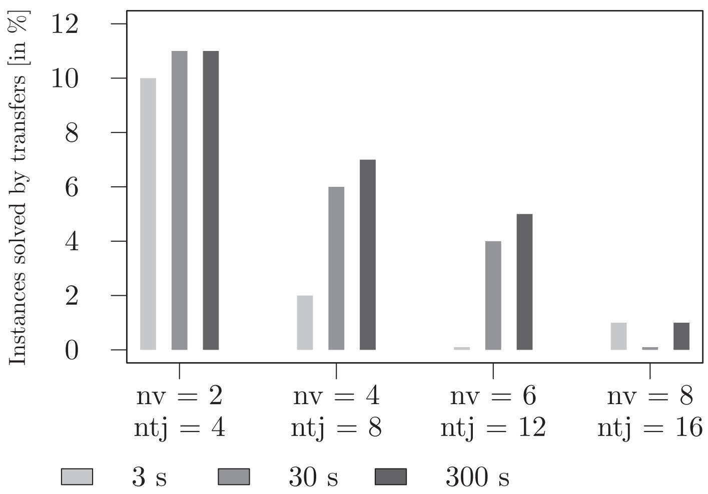

Figure 6: Percentage of test instances solved with at least one transfer operation concerning problem size and calculation time-limit

transport jobs and transfer locations lead to more options to build schedules.

We found that many transfers can be identified for small problem instances (ntj = 4, nv = 2 and ntj = 8, nv = 4), as demonstrated in figure 5. With ntj = 16 and nv = 8, it was not even possible to identify a transfer. To demonstrate the influence of calculation time the experiment was also carried out, with a time-limit of 3 s and 300 s. Figure 6 summarizes the results: for small problem instances, the results are independent of calculation time. With an increasing problem size, better results can be achieved with additional calculation time.

As mentioned before regarding the control of AGVS, the solutions of the ALNS-wt should be carefully examined. A transfer should only be accepted if it improves the schedule of the ALNS-wot. Overall, the results indicate that it is possible to identify beneficial transfer operations by the ALNS-wt. However, the application is limited to small vehicle systems.

5.3.3 Comparison to an exact solution approach

Here we will evaluate how great the difference in solution quality is between the exact method and the heuristic. Thus it becomes clear how much potential for improvement remains through further development of the heuristic.

We compared the results of the ALNS-wt with an exact solution of the model described in section 4.1. Pursuing an exact approach to solving the problem is not suitable for real-time applications. None of the randomly generated problem instances (see table 1) could be solved within 30 s. Even with the extended time-limit of 3000 s, only 48 instances could be solved optimally by the standard solver, whereas 6 instances contain transport load transfers. 3 of them are identified by the ALNS-wt heuristic too.

So, not surprisingly, the solver allows higher solution quality than our heuristic with the price of high computational efforts. The ALNS-wt, however, provides ”good” solutions generated in a fraction of time but lacks always find beneficial transfers. All in all, the ALNS-wt's average objective deviation is 0.9 %.

5.3.4 Comparison to a standard dispatching approach

In the absence of an established benchmark, the comparison to a standard dispatching approach (see section 4.3) will serve as a reference in the following. Furthermore, the results will show the potential of the heuristics in comparison to a conventionally controlled AGV system.

As already described, it is not clear in advance whether ALNS-wt or ALNS-wot leads to a better result. Therefore, it is shown in table 2 which solution quality can be achieved compared to using ALNS-wt/ALNS-wot individually and combining both methods.

Compared to the standard dispatching approach, the ALNS-wot and the ALNS-wt show significant improvements (see table 2). On average, ALNS-wt performs around 27.9 % and ALNS-wot 28.3 % better in terms of cost (measured in time for driving and handling). As shown in table 2, the improvement rises by calculation time. As discussed before, ALNS-wot performs slightly better overall than ALNS-wt. A combination of both approaches, shown in column [&], has the best results (average cost improvement of 28.5 %). Here always, the minimum result was taken for evaluation.

Table 2: Improvements by the ALNS compared to a standard dispatching approach ([&]: minimum result of ALNS-wt and ALNS-wot)

		Time-limit Approach								
		3 s			30 s			300 s		
ntj	nv	wt	wot	[&]	wt	wot	[&]	wt	wot	[&]
4	2	18.3	18.1	18.4	18.3	18.2	18.4	18.5	18.4	18.5
8	4	28.2	28.8	29.1	29.2	29.3	29.6	29.8	29.7	29.9
12	6	29.5	31.0	31.1	32.6	33.2	33.5	33.5	33.8	34.1
16	8	29.3	30.2	30.4	32.8	33.7	33.9	34.5	35.2	35.3

5.4 Influence of system characteristics

The paragraphs above illustrate how transport load transfers can improve system performance concerning AGV-fleet size (nv) and number of jobs (ntj). Besides these high-level AGV system parameters, we assume an impact of different other parameters on the beneficiary of transfers. So, based on the parameters specified in section 5.2 we conducted a full factorial experiment study. For each parameter, 10 different random problem instances were generated and solved by the ALNS heuristic with and without transport load transfers (wt/wot) – all in all 58.320 test instances.

Overall, 8 % of the test instances are solved considering at least one transport load transfer. In about 1 % of these test instances, a transport job was split into more than two sub-transports. Multiple splits lead to a better result, but the overall effect is limited.

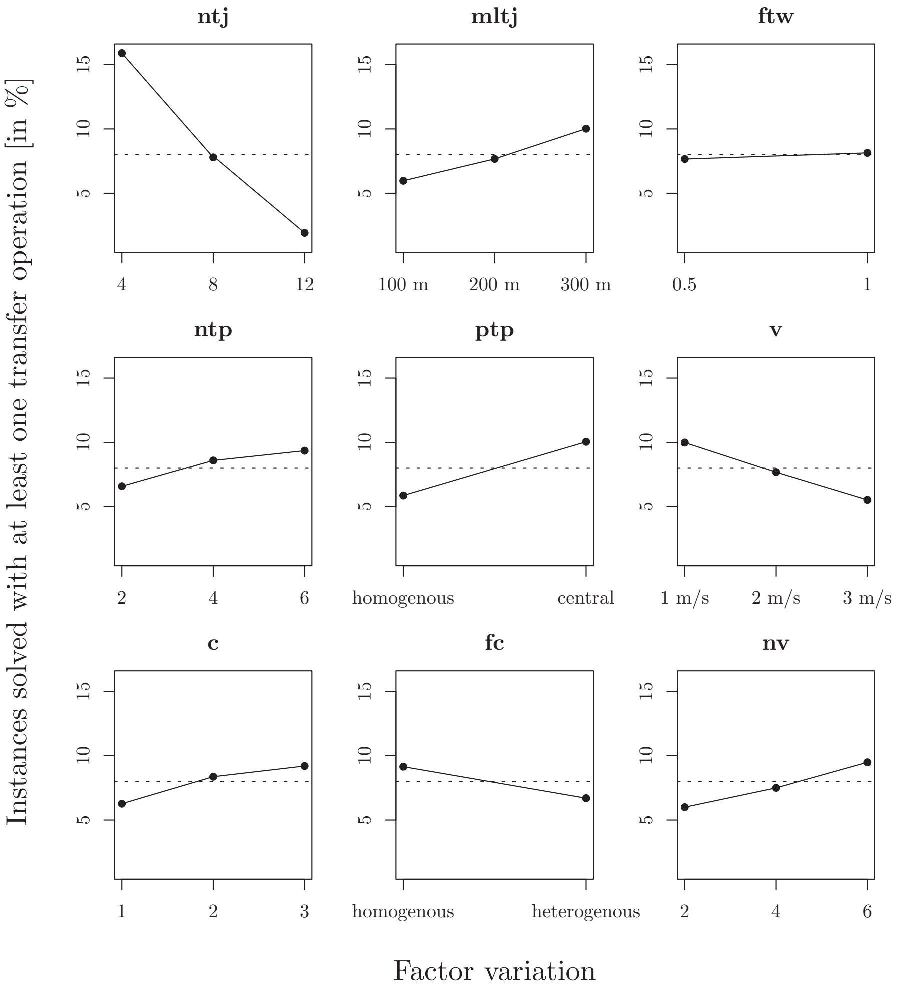

Figure 7: The influence of basic system characteristics to the percentage of solutions solved by transport load transfers

Compared to the identical scenario neglecting transfers, the average improvement was 2.2 % and 26.4 % on maximum. The possibility that transfers are considered for schedule generation is highly dependent on the problem characteristic. Figure 7 provides an overview of the parameter effects. In each box, the average value for all experiments is described by a dotted line. In the following, we describe the influence for each parameter. A detailed overview is given in table 8 (see appendix).

5.4.1 Characteristic of transport jobs

As a result, for the parameter ntj (number of transport jobs), we observe the highest influence on the probability that an solution contains at least one transport load transfer. Starting from an average value of 15.9 % for ntj = 4, the value decreases to a value of 1.9 % for ntj = 12. Following the discussion in section 5.3, the effect is related to the algorithmic approach. Alternative algorithms could result in more transfers, even for large problem instances.

For the parameter mltj (min length of transport jobs), a rising percentage of transport load transfers is shown by increased job distance. Long transport jobs could result in large detours for the executing vehicles concerning the other jobs carried by the vehicle or the vehicle end position. Thus, even considering the additional effort for a transfer execution, transfers can lead to an overall improvement through these detours. From 6.0 % for a mltj = 100 m, the value increases to 10.0 % for mltj = 300 m.

By the parameter ftw (factor time windows), we investigated by shortening end-time windows for transport load drop-off different time-limits to execute the transport tasks. A factor ftw = 0.5 results in a smaller amount of time. Here only 60 % of the problem instances could be solved by respecting the defined constraints. A percentage of 96 % was solved for ftw = 1. The results represent a comparable relevance of transfers for booth variants regarding the share of transport load transfers. However, we found 22 instances that could only be solved by respecting the time window constraints considering transfer operations.

5.4.2 Characteristic of transfer points

The influence of the characteristic of transfer points was tested in this experiment in two ways. On the one hand, we varied the number of transfer points (parameter ntp) and on the other hand, we compared two approaches for transfer point positioning (parameter ptp). A rising amount of transfer locations results in a higher amount of transport load transfers. The value increases by 2.8 % from ntp = 2 to ntp = 6. If there are many locations to transfer, the chance is rising that a transfer point can be visited with short detour by the vehicles to exchange a transport load. The positioning strategies for transfer points result in more transport load transfers for problem instances with central transfer point positioning (5.9 % for ptp = heterogeneous and 10.1 % for ptp = central). A central positioning raises the chance of visiting transfer locations with short detours. For example, by heterogenous positioning, there are often transfer locations near the borders of the system their vehicles drive with a low chance. Hence, these positions can mainly be visited at a high cost (for driving) for the vehicles.

5.4.3 Characteristic of vehicle fleet

A rising vehicle velocity (parameter v) lowers the percentage of transfers found. It is decreasing from 10.0 % for v = 1 m/s to 5.5 % for v = 3 m/s. The main reason is that the cost for detours is related to the vehicle velocity. It is more unlikely to decompensate the additional costs for material handling at the transfer point to exchange a transport load.

An increase of parameter c (vehicle capacity) leads to more transfers. Higher capacity results in higher flexibility to generate schedules. This way, synergy effects can be achieved by combining transport loads with similar destinations. However, with a capacity of only 1 transport load, it is also possible to benefit from transport load transfers. A possible reason here are lower detours regarding the vehicle end locations.

Homogenous vehicle fleets (see parameter fc) tend to benefit more from transfer operations. A detailed analysis has shown that fast vehicles are preferred to execute the entire transport for heterogeneous fleets. The difference of fc = homogenous and fc = heterogeneous vehicles is 2.5 %.

For a rising vehicle fleet size (parameter nv), there is also a positive effect on the share of transport load transfers. For test instances, with nv = 2, 6.0 % of the schedules contain a transport load transfer. With nv = 6, in 9.5 % of instances, transfers are observed. Since we consider vehicles with defined start and end locations, a rinsing number of vehicles raises the possibility that a part of transport can be realized with a short detour.

In summary, experiments by test instances revealed that transport load transfers lead to improved performance. Single instances are improved by more than 20 % in driving and handling costs. The potential is essentially dependent on the characteristic of the planning scenario. While the relevance of transfers is marginal over all experiments, they lead to significant improvements for specific parameter configurations.

6 CASE STUDY

We verify by a material flow simulation study the advantageousness of transport load transfers and the approach's applicability to control AGVS in the intralogistics domain. The experiments are related to an exemplary layout shown in figure 8 that could serve in a warehouse or a job shop production system [see 2, 14]. The layout is characterized by a unidirectional path topology with 36 positions to start/end transports and 9 transfer locations. Each transfer location holds an individual position for each vehicle to avoid blocking. Transfer locations can be used to buffer transport loads without capacity restrictions.

The following experiments consider a fleet of 4 vehicles with a velocity of 0.5 m/s, a capacity of 2 transport loads and a handling time of 10 s. 3 central transfer locations (t_4, t_5 and t_6) are activated. We set a calculation time-limit for ALNS-wt and ALNS-wot of 3 s to test real-time capabilities. A new schedule is generated when the system status is changed by a material handling or a new transport job. Transport jobs are generated in advance. These parameters are assumed to be reasonable for intralogistics applications from our experience.

Following the results from section 5.4, the identification of meaningful transport load transfers is highly dependent on the number of transport jobs to schedule. We varied the number of transport jobs for the case study by assuming different utilization levels by applying different interim arrival times (ia) of new transport jobs. Based on preliminary tests, we have determined that executing 25 repetitions with different seeds per factor combination is sufficient to achieve reliable results.

6.1 Basic example

An illustrative example scenario was selected to test the applicability to a dynamic system. It consists of three transport jobs that are transmitted to the system simultaneously. There is a time-limit for executing the jobs of 480 s from when the transport job is submitted. In the beginning, all the vehicles are in the assigned parking positions. The transport jobs are selected in a way that the exchange of transport loads provides a cost advantage. These three transport jobs are executed repeatedly to test the algorithm. There is sufficient time between iterations for the vehicles to reach their parking position again.

- transport jobs
  - r_0: start at location 25; end at location 10
  - r_1: start at location 4; end at location 25
  - r_2: start at location 16; end at location 13
- vehicles
  - k_0: start at location 11; end at location 11
  - k_1: start at location 26; end at location 26
  - k_2: start at location 8; end at location 8
  - k_3: start at location 29; end at location 29

The best-known schedule neglecting transport load transfers yields an objective function value (operation time for driving and handling) of 1472 s and 3 vehicles are used. Each of the vehicles operate one transport job on its own. There are several solutions considering transfers that allow better objective function values. Two of them are described in the following.

The best known schedule involves the transfer of transport job r_0 and r_1 between vehicles k_1 and k_2 at t_5. Transport job r_2 is transported by k_3. Hence, also three vehicles are involved. An objective function value of 1229 s is achieved and associated with a cost reduction of 16.5 %.

In another schedule variant, all transport jobs are executed by k_0 and k_1. Transport job r_0 is transferred

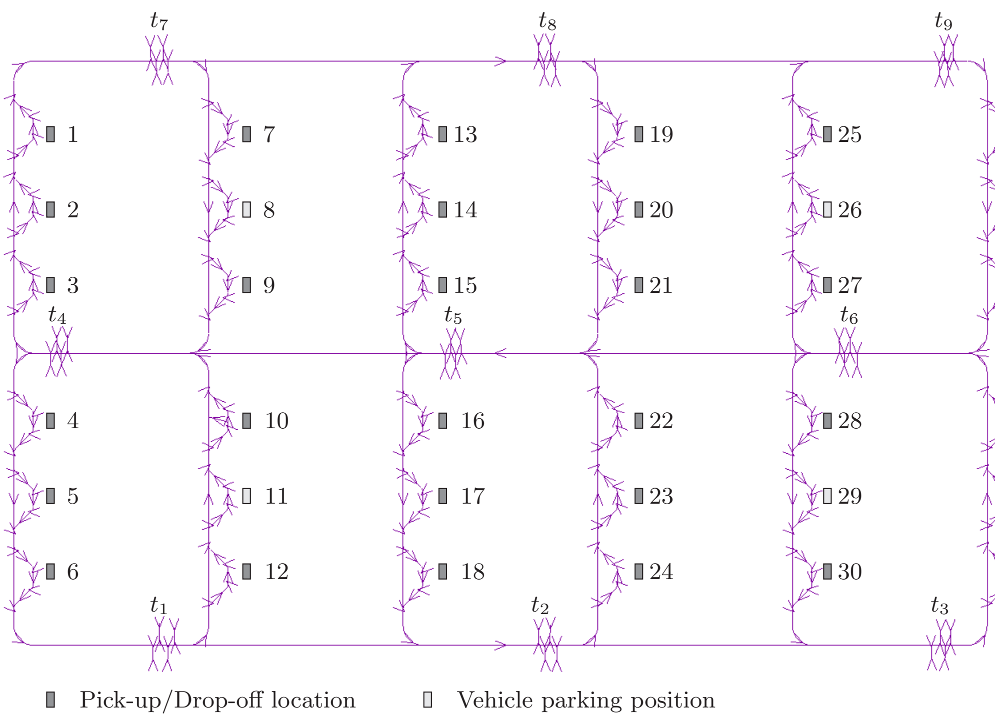

Figure 8: Layout of the simulation model with a dimension of 50 m in width and 30 m in heigth

Table 3: Material flow simulation results from a basic example

Indicator	ALNS-wt	ALNS-wot	Effect [in %]
Costs driving [in s]	369028	427380	-13.7 %
Costs handling [in s]	27100	20020	+35.4 %
Sum [in s]	369128	447400	-11.5 %
Average delivery time [in s]	391	282	+38.7 %
Average transport time [in s]	312	205	+52.2 %
Average waiting time [in s]	79	77	+2.6 %

from k_1 to k_0 at t_5. An objective function value of 1295 s is achieved, resulting in a cost reduction of 12 %. Compared to the other option, the disadvantage is a slightly higher cost. However, one less vehicle is needed.

Assuming that all orders are known in advance (e.g. in an offline scheduling scenario), one of these variants could be chosen and applied repeatedly. This would allow choosing between minimization of average vehicle utilization and vehicle fleet size.

The evaluation of the entire simulation run with an ad-hoc calculation of schedules (see table 3) shows that by taking transfers into account, approximately 11.5 % of the costs for driving and handling could be saved. Due to the additional handlings and waiting times at the transfer location, the average delivery time increases by 38.5 %. The average waiting time until the first pick-up of the transport loads is comparable. Thus, the increase in the delivery time results mainly from increased transport times.

6.2 Sensitivity analysis

For further evaluation and to reveal that the effects of transport system characteristics described in section 5.4 also apply to dynamic systems, we conduct a sensitivity analysis. It starts from an experiment with an interim arrival time of transport jobs (ia) of 200 s. Time windows are configured so that transport jobs must be executed within 600 s.

The experiment with ia = 200 s is characterized by low average utilization of the vehicles (about 35 %). The costs (sum of operation time for driving and handling) are improved by 1.34 %. Average vehicle utilization was reduced by a comparable amount.

The results are summarized in table 4. On average, 5.3 % more handling operations due to transport job transfers are carried out. Transfer-related delays due to additional buffering at the transfer location increase the average delivery time by 0.75 %. Further analysis revealed that the increase results mainly from longer transport times (3.4 %) while the waiting time until

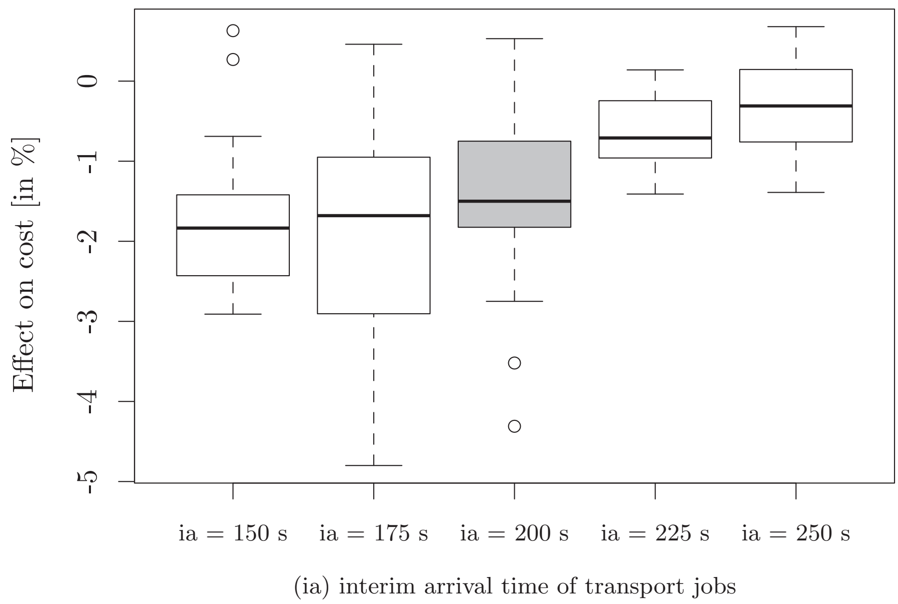

Figure 9: Effect on different throughput levels on the cost of transport execution; whiskers limits are set to 1.5 times the interquartile range from lower to upper quartile

Table 4: Reference experiment for sensitivity analysis

Experiment	Cost [in %]	Delivery time [in %]	# handling [in %]
ia = 200 s	-1.34	+0.75	+5.3

the first pick-up of the load carrier can be improved by 1.8 %.

Figure 9 shows the results based on a variation of the throughput, represented by different interim transport job arrival times (ia). The parameter varies between ia = 150 s (corresponds to 24 transports per hour) and ia = 250 s (corresponds to 14.4 transport jobs per hour). Shorter ia leads to an overload of the system since problem size becomes to large to be processed by the heuristic (see section 5.3). Longer ia is followed by very low utilization, with fewer options to create synergy effects by other transport jobs. The number of material handling operations increases for ia = 250 s only by 3.1 %. Thus, transfers are less relevant here. Nevertheless, costs are reduced by 0.3 %.

With higher throughput (ia = 150 s), about 8.5 % of additional handling operations were performed due to transfer operations. This is associated with a reduction in cost of 1.7 %. The highest cost improvement (1.9 %) was measured at ia = 175 s. Here, improvements of up to 5 % were possible in single simulation runs. For this scenario, a reduction in delivery time (0.7 % on average) was observed.

We tested transport job, transfer location, and vehicle-specific parameters in the following. An overview of the results can be found in table 5. All in all, the results are comparable to those in section 5.4. In contrast, no effects concerning the length of the transport jobs (mltj) can be shown here. The layout limits the possibilities for varying the length in this experiment. Also, a higher number of transfer locations (ntp) is not followed by higher improvements. A reason could be that the additional transfer locations are in a less central regions and they are therefore less visited by the vehicles.

Regarding the characteristics of the vehicles, we conclude that a higher velocity (v) of the vehicles or higher material handling time (ht) make transfers less attractive since detours and additional handlings are less likely compensated by synergy effects. Also, for a capacity (c) of one transport load, we were able to show improvements by transfer operations. However, the benefits of transfers decrease with less capacity or even higher capacity. As discussed before, in a heterogeneous vehicle fleet, transfers are less relevant.

7 CONCLUSION AND OUTLOOK

This paper presents a meta-heuristic (Adaptive Large Neighborhood Search) for task assignment in AGVS in the intralogistics domain, considering transport load transfers. The intention is to improve AGV system performance by reducing vehicle utilization due to

Table 5: Sensitivity analysis (reference experiment ia = 200 s)

Experiment	Cost [in %]	Delivery time [in %]	# handling [in %]
ia = 150 s	-1.74	+1.23	+7.95
ia = 175 s	-1.90	-0.72	+6.55
ia = 200 s	-1.34	+0.75	+5.30
ia = 225 s	-0.61	+0.91	+3.80
ia = 250 s	-0.32	+0.45	+3.05
mltj = 40 m	-1.21	+0.87	+6.35
mltj = 60 m	-1.42	+1.18	+8.45
mltj = 80 m	-1.30	+1.74	+11.55
ntp = 9	-0.84	+1.07	+5.15
v = 0.75 m/s	-0.05	-0.03	+1.30
v = 1.0 m/s	+0.03	-0.39	+0.40
ht = 5 s	-1.23	+1.00	+5.35
ht = 20 s	-0.53	+0.48	+4.45
ht = 30 s	-0.42	+1.40	+5.30
c = 1	-0.94	+0.47	+3.85
c = 4	-0.86	+1.15	+5.00
fc = heterogenous	-0.24	-0.04	+1.15

more efficient transport execution to yield higher throughput. To execute a transfer: a vehicle buffers a transport load at a predefined transfer location; later, a vehicle continues the transport. Transfer operations are planned ad-hoc considering the current system state.

The algorithm was evaluated using static test instances and a material flow simulation study to address system dynamics and stochasticity. The test instances represent scenarios with different characteristics, e.g. varying vehicle velocity or a minimum distance of transport jobs, resulting in an average cost improvement for driving and material handling of 2.2 % when transfers are considered. Whereas, for individual scenarios, the application of transport load transfers resulted in cost improvements of more than 25 %. The results show that the effects are sensitive to parameters like transportation distance or handling time.

Applying a dynamic simulation study, we found that the effect of transport load transfers highly depends on the characteristics of the task assignment scenario. For a simplified example with a recurring sequence of transports, savings in cost for transport execution of over 10 % are demonstrated. For randomly selected transport jobs with appropriate parameter configurations, vehicle driving and handling costs are reduced by 1.9 % on average and a maximum of 5 %.

The proposed concept and the results obtained are a first step towards evaluating the benefits of transport load transfers in AGVS. In the studies presented here, the investigation has been limited to transport tasks that can be subdivided into any number of sub-transports that are done at static transfer points. The work was focused on reducing the costs of transport execution to improve vehicle utilization. Generally, other options are possible and reasonable for these aspects. It might also be beneficial to study alternative algorithms which promise to scale to a greater extent with problem size. This publication focuses on investigating the profound principles of abstract models. Further studies based on real-life examples are necessary to validate these findings.

However, our investigation of an AGV system with a few vehicles and task assignments by detailed schedules revealed that transport load transfers could achieve benefits. We will focus on systems with a higher number of vehicles for further research. On the one hand, we will test other configurations of our heuristic, like large parallelization and extended calculation time. On the other, we will customize a standard dispatching approach for task assignment in AGVS that is proven for large-scale applications to consider transport load transfer operations.

ACKNOWLEDGMENT

The work was carried out as a part of the IGF research project 21269 BR / 1 ('iDynaTrans') and supported by 'Bundesvereinigung Logistik (BVL) e. V.'

Supported by:
Federal Ministry for Economic Affairs and Energy
on the basis of a decision by the German Bundestag

IGF Industrielle Gemeinschaftsforschung

REFERENCES

[1] John W. Baugh, Gopala Krishna Reddy Kakivaya, and John R. Stone. "Intractability of the Dial-A-Ride Problem and a Multiobjective Solution Using Simulated Annealing". In: *Engineering Optimization* 30.2 (1998), pp. 91–123.

[2] Patric Beinschob et al. "Semi-automated map creation for fast deployment of AGV fleets in modern logistics". In: *Robotics and Autonomous Systems* 87 (2017), pp. 281–295.

[3] Brian Coltin and Manuela Veloso. "Scheduling for Transfers in Pickup and Delivery Problems with Very Large Neighborhood Search". In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*. AAAI'14. AAAI Press, 2014, pp. 2250–2256.

[4] Jean-François Cordeau. "A Branch-and-Cut Algorithm for the Dial-a-Ride Problem". In: *Operations Research* 54.3 (2006). Publisher: INFORMS, pp. 573–586.

[5] Cristián E. Cortés, Martín Matamala, and Claudio Contardo. "The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method". In: *European Journal of Operational Research* 200.3 (2010), pp. 711–724.

[6] N. Danloup, H. Allaoui, and G. Goncalves. "A comparison of two meta-heuristics for the pickup and delivery problem with transshipment". In: *Computers & Operations Research* 100 (2018), pp. 155–171.

[7] M. De Ryck, M. Versteyhe, and F. Debrouwere. "Automated guided vehicle systems, state-of-the-art control algorithms and techniques". In: *Journal of Manufacturing Systems* 54 (2020), pp. 152–173.

[8] Pius J. Egbelu and Jose M. A. Tanchoco. "Characterization of automatic guided vehicle dispatching rules". In: *International Journal of Production Research* 22.3 (1984), pp. 359–374.

[9] Said Hammouda, Ali Lemouari, and Mokhtar Taffar. "Hybrid Heuristics for The Dial-A-Ride Problem with Transfer and Synchronization in Central Hub". In: *Statistics, Optimization & Information Computing* 10.1 (2022), pp. 263–281.

[10] Y.-C. Ho and S.-H. Chien. “A simulation study on the performance of task-determination rules and delivery-dispatching rules for multiple-load AGVs”. In: *International Journal of Production Research* 44.20 (2006), pp. 4193–4222.

[11] Byung Ho Jeong and Sabah U. Randhawa. “A multi-attribute dispatching rule for automated guided vehicle systems”. In: *International Journal of Production Research* 39.13 (2001), pp. 2817–2832.

[12] Nabin Kafle, Bo Zou, and Jane Lin. “Design and modeling of a crowdsource-enabled system for urban parcel relay and delivery”. In: *Transportation Research Part B: Methodological* 99 (2017), pp. 62–82.

[13] Tuan LeAnh. “Intelligent Control of Vehicle-Based Internal Transport Systems”. PhD Thesis. Rotterdam: Erasmus Research Institute of Management, Erasmus University Rotterdam, 2005.

[14] Guomin Li et al. “Tasks assigning and sequencing of multiple AGVs based on an improved harmony search algorithm”. In: *Journal of Ambient Intelligence and Humanized Computing* 10.11 (2019), pp. 4533–4546.

[15] Renaud Masson, Fabien Lehuédé, and Olivier Péton. “An Adaptive Large Neighborhood Search for the Pickup and Delivery Problem with Transfers”. In: *Transportation Science* 47.3 (2013), pp. 344–355.

[16] Snežana Mitrović-Minić and Gilbert Laporte. “The Pickup And Delivery Problem With Time Windows And Transshipment”. In: *INFOR: Information Systems and Operational Research* 44.3 (2006), pp. 217–227.

[17] Peter Oertel. “Routing with reloads”. PhD Thesis. Cologne: Faculty of Mathematics and Natural Sciences, University Cologne, 2000.

[18] Sophie N. Parragh, Karl F. Doerner, and Richard F. Hartl. “A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations”. In: *Journal für Betriebswirtschaft* 58.2 (2008), pp. 81–117.

[19] Hanne L. Petersen and Stefan Ropke. “The Pickup and Delivery Problem with Cross-Docking Opportunity”. In: *Computational Logistics* 6971 (2011). Ed. by Jürgen W. Böse et al., pp. 101–113.

[20] Jacopo Pierotti and J. Theresia van Essen. “MILP models for the Dial-a-ride problem with transfers”. In: *EURO Journal on Transportation and Logistics* 10 (2021), p. 100037.

[21] Yuan Qu and Jonathan F. Bard. “A GRASP with adaptive large neighborhood search for pickup and delivery problems with transshipment”. In: *Computers & Operations Research* 39.10 (2012), pp. 2439–2456.

[22] A. Rais, F. Alvelos, and M. S. Carvalho. “New mixed integer-programming model for the pickup-and-delivery problem with transshipment”. In: *European Journal of Operational Research* 235.3 (2014), pp. 530–539.

[23] Stefan Ropke. “PALNS - A software framework for parallel large neighborhood search”. In: *8th Metaheuristic International Conference CDROM*. 2009.

[24] Stefan Ropke and David Pisinger. “An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows”. In: *Transportation Science* 40.4 (2006), pp. 455–472.

[25] Afonso Sampaio et al. “Delivery systems with crowd-sourced drivers: A pickup and delivery problem with transfers”. In: *Networks* 76.2 (2020), pp. 232–255.

[26] M. W. P. Savelsbergh and M. Sol. “The General Pickup and Delivery Problem”. In: *Transportation Science* 29.1 (1995), pp. 17–29.

[27] Hiva Shiri, Morteza Rahmani, and Morteza Khakzar Bafruei. “Examining the impact of transfers in pickup and delivery systems”. In: *Uncertain Supply Chain Management* (2020), pp. 207–224.

[28] Günter Ullrich. *Automated Guided Vehicle Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015.

[29] Stefan Voigt and Heinrich Kuhn. “Crowdsourced logistics: The pickup and delivery problem with transshipments and occasional drivers”. In: *Networks* (2021).

[30] Yimeng Zhang et al. “Pickup and Delivery Problem with Transshipment for Inland Waterway Transport”. In: *Computational Logistics* 12433 (2020). Ed. by Eduardo Lalla-Ruiz, Martijn Mes, and Stefan Voß, pp. 18–35.

[31] Boden, Patrick et al. (2020). “Dispatching of Multiple Load Automated Guided Vehicles based on Adaptive Large Neighborhood Search”. In: *Operations Research Proceedings 2019*. Cham: Springer International Publishing, pp. 375–380.

APPENDIX

Table 6: Parameter configuration for the introductory example

Parameter	Configuration
ntj	2
mltj [in m]	100
ftw	1
ntp	4
ptp	homogenous
v [in m/s]	1
ht [in s]	10
c	2
nv	2
fc	homogenous

Table 7: Parameter variation for evaluation of system characteristics by test instances

Parameter	Variation
ntj	[4; 8; 12; 16]
mltj [in m]	[100; 200; 400]
ftw	[1; 2]
ntp	[4]
ptp	[homogenous; central]
v [in m/s]	[1; 2; 4]
ht [in s]	[5; 10]
c	[1; 2; 4]
nv	[2; 4; 6; 8]
fc	[homogenous; heterogenous]

Table 8: Summary of the results for the analysis of basic system characteristics

ntj	4	8	12
Share solved in %	66	82	84
Share transfers in %	15.9	7.8	1.9
Avg. improvement in %	3.5	1.8	2.1
mltj	100 m	200 m	300 m
Share solved in %	74	78	82
Share transfers in %	6.0	7.7	10.0
Avg. improvement in %	3.2	2.6	2.6
ftw	0.5	-	1
Share solved in %	60		96
Share transfers in %	7.7		8.1
Avg. improvement in %	3.6		2.3
ntp	2	4	6
Share solved in %	78	72	78
Share transfers in %	6.6	8.6	9.4
Avg. improvement in %	2.9	2.7	2.7
ptp	homogenous	-	central
Share solved in %	78		78
Share transfers in %	5.9		10.1
Avg. improvement in %	2.7		2.8
v	1 m/s	2 m/s	3 m/s
Share solved in %	89	78	65
Share transfers in %	10.0	7.7	5.5
Avg. improvement in %	2.7	2.6	3.2
c	1	2	3
Share solved in %	77	77	79
Share transfers in %	6.3	8.4	9.2
Avg. improvement in %	2.7	2.6	2.9
fc	homogenous	-	heterogenous
Share solved in %	79		77
Share transfers in %	9.2		6.7
Avg. improvement in %	2.7		2.8
nv	2	4	6
Share solved in %	54	86	93
Share transfers in %	6.0	7.5	9.5
Avg. improvement in %	4.1	2.7	2.3