

Agent-based transportation planning compared with scheduling heuristics

Martijn Mes*, Matthieu van der Heijden, Aart van Harten

18 May 2004

Abstract

Here we consider the problem of dynamically assigning vehicles to transportation orders that have different time windows and should be handled in real time. We introduce a new agent-based system for the planning and scheduling of these transportation networks. Intelligent vehicle agents schedule their own routes. They interact with job agents, who strive for minimum transportation costs, using a Vickrey auction for each incoming order. We use simulation to compare the on-time delivery percentage and the vehicle utilization of an agent-based planning system to a traditional system based on OR heuristics (look-ahead rules, serial scheduling). Numerical experiments show that a properly designed multi-agent system may perform as good as or even better than traditional methods.

Keywords: Transportation; Multi-agent systems; Auctions/bidding

1 Introduction

Transportation networks are nowadays moving towards more flexible and open systems in order to cope with stochastic and real-time demand that has to be satisfied fast within small time-windows. This has consequences for the logistical planning and control of such networks. Traditionally, these planning and control systems are based on operations research (OR) techniques (heuristics and optimization methods). Because of the system dynamics and fast response requirements, one may wonder whether these techniques are still most suitable.

*Corresponding author. Address: Department of Operational Methods for Production and Logistics, Faculty of Business, Public Administration and Technology, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands; phone +31-53-489-4062; fax: +31-53-489-2159; e-mail: m.r.k.mes@utwente.nl.

Firstly, most optimization algorithms require a lot of information, often in advance. Such algorithms are sensitive to incompleteness of required data. Furthermore because the relevant data are frequently altered, we need extensive and reliable data exchange which leads to an extensive and possibly vulnerable planning system. Secondly, the time required for the algorithm may not permit timely response to unexpected events such as equipment failure and the arrival of rush orders. Thirdly, flexible transportation networks may consist of multiple independent organizational units that are working in an autonomous, self-interested and not necessarily cooperative way. Therefore, these individual players may not be willing to share all their information, so that traditional centralized or hierarchical approaches are not applicable anymore. So one may wonder whether traditional techniques are most suitable under heavy system dynamics and fast response requirements.

An alternative that has been proposed within the computer science literature is the multi-agent system (MAS). Such a system consists of independent intelligent control units, which correspond to physical and functional entities. It has been presented as a promising solution for controlling complex networks, providing more flexibility, reliability, adaptability and reconfigurability. Agents act autonomously by pursuing their own interest and interact with other agents, for example using information exchange and negotiation mechanisms. In a transportation network, each order (job) and each resource can have its own goal-directed agent. For example, a job agent may focus on on-time delivery against the lowest possible costs, and a resource agent may strive for utilization and/or profit maximization. A proper pricing and trade mechanism is needed to optimize the system wide performance, such as the minimization of the total lateness at acceptable costs.

The principle of multi-agent systems is elegant and has clear advantages from an ICT point of view. However, it is unclear whether the system-wide performance will be similar to or even better than the performance of more centralized or hierarchically organized planning systems. It is even not guaranteed whether and when a multi-agent system will show a stable behavior. That is, will all orders be transported, will resources properly be utilized and will prices remain within reasonable bounds in the absence of a coordination mechanism?

Although many papers have appeared on multi-agents systems, also applied to logistics, literature on the performance comparison between traditional OR-based systems and multi-agents systems is scarce. In this paper, we aim to make such a comparison for a transportation network where orders (full truck loads) with various soft time windows arrive continuously and should be handled in real time. From the wide range of decisions to be taken, we focus on the assignment

of jobs, characterized by an origin, a destination, a release time, a due time and penalty costs for tardiness, to vehicles that are dispersed over the network. New orders can be given to the system asynchronously. Because a fast response is required, we use more traditional local dispatch rules and serial scheduling as benchmarks, see (Heijden, Ebben, Gademane, and van Harten 2002; Ebben, van der Heijden, and van Harten 2004). For the multi-agent system, we develop an auction mechanism with several pricing variants. To compare agent-based and more traditional approaches, we have developed a discrete event simulation model that we use to perform a large range of numerical experiments. As performance criteria we focus on the average in-time delivery percentage, variation in the in-time delivery percentage over smaller time periods as a measure of robustness and the empty mile percentage.

The remainder of this paper is structured as follows. In the next section, we give an overview of related literature and we explain the contribution of our paper. In Section 3, we present our model and in Section 4 we discuss our choice for a particular agent based planning concept. Next, we discuss several options for agent bidding and job assignments using auctioning in Section 5. In Section 6, we briefly present two more traditional planning approaches that we use as benchmarks in a simulation study. We describe the experimental settings in Section 7 and provide the numerical results from this study in Section 8. We end up with conclusions, remarks on generalizations and directions for further research (Section 9).

2 Related literature

2.1 Transport planning

Our problem of assigning jobs to vehicles in a transportation network is well-known in the area of vehicle routing problems (VRP) as a multi-vehicle pickup and delivery problem with time windows, also indicated as a dial-a-ride problem. Dial-a-ride problems arise in several practical applications such as the transportation of elderly and/or disabled persons, shared taxi services, certain courier services and so on. We consider a variant with full truckloads, stochastic arrival of orders and stochastic handling- and travel times, where even the probability distributions are not known beforehand.

The VRP and its variants have been studied extensively; see (Laporte 1992; Toth and Vigo 2002) for a survey. It is well-known that most variants of the VRP problem are NP-hard, so that an optimal solution given a very small reaction time is virtually impossible. Most work focuses on

static and deterministic problems in which all information is known at the time of the planning of the routes, see for example (Desrosiers, Dumans, Solomon, and Soumis 1995; Fischer 1995). In dynamic and stochastic vehicle routing problems (also known as real-time routing and dispatching problems) the input data (travel times, demands) are stochastic and depend on time. Therefore, the output of a dynamic VRP (DVRP) is not a set of routes, but rather a policy that prescribes how the routes should evolve as a function of those inputs that evolve in real-time (Psaraftis 1988). This allows for situations where some relevant information for the planning of the routes is not known on beforehand, such as the demand process and order handling times. Also, information can change after the initial routes have been constructed.

Routing and scheduling in a dynamic environment has been studied by a number of authors, see for example (Psaraftis 1988; Gendreau and Potvin 1998). The most common approach to handle these problems is to solve a model using the data that are known at a certain point in time, and to re-optimize as new data become available. Because a fast response is required in a real-time environment, a solution is usually achieved by using relatively simple heuristics or by parallel computation methods, see (Giani, Guerriero, Laporte, and Musmanno 2003) for an overview of approaches.

The dynamic assignment problems as discussed in a number of papers by Powell (Powell 1996; Powell and Carvalho 1998; Godfrey and Powell 2002) also show similarities. These problems consist of dynamically assigning resources to tasks. These papers differ from our work, because (1) they consider only one known job per vehicle to be scheduled (2) the price of a job is given externally and not subject to negotiation. The rest of the schedule remains open until arrival of that load at the destination. Here we allow complete schedules comprising series of jobs for each vehicle. In (Powell and Carvalho 1998), Powell presented a novel formulation (called Logistics Queues Network (LQN) there), in which a big and complex problem is being replaced by a series of very small problems. This concept provides the ability to consider more real world details which cannot be modeled in traditional approaches. However this concept is still based on a hierarchical control structure. In this paper we will apply completely decentralized agent technology to solve these dynamic assignments problems.

2.2 Agent technology

According to Wooldridge and Jennings (1995), an agent is a hardware or software based computer system with key properties autonomy, social ability, reactivity and pro-activeness. Auton-

omy means that agents can operate, to some degree, without external invocation or intervention. Therefore autonomous agents have individual states and goals, and based on their own knowledge they are trying to achieve these goals on behalf of their owners.

When we have a group of agents that are in some way connected to each other, we speak of a multi-agent system (MAS). In (Wooldridge and Jennings 1995) MAS is defined as a "loosely coupled network of problem solvers that work together to solve problem that are beyond the individual capabilities or knowledge of each problem solver". For a more detailed discussion on these topics we refer to (Parunak 1998). An overview of different multi-agent applications can be found in (Jennings, Sycara, and Wooldridge 1998).

Despite these clear characteristics, many different viewpoints on the multi-agent concept itself exist. A major source of inspiration for our research comes from free-market economics where systems are being controlled by pricing mechanisms. Especially the allocation of scarce resources by groups of self-optimizing individuals, which is a central concept to microeconomics, is directly relevant to the development of multi-agent systems where agents for example may represent companies, customers and suppliers. For a detailed overview on the early work on these systems we refer to (Clearwater 1996) which contains applications of market-based systems in a number of diverse areas. In this paper we will also use a market-based control mechanism for the allocation of vehicles to transportation orders.

2.3 Agent-based logistic planning

Agents have been used for a vast range of applications, ranging from e-mail assistants to air traffic controllers, see (Jennings, Sycara, and Wooldridge 1998). In the last years, research on multi-agent systems also has boosted in the logistics and operations research community. However, it is sometimes hard to distinguish some agent-based scheduling methods from more traditional local search schemes, other than that they facilitate parallel computations because of their distributed nature.

Quite some papers deal with manufacturing scheduling and control. For example, (Cardon, Galinho, and Vacher 2000) uses genetic algorithms to solve job-shop scheduling problems. The authors represent genetic entities by agents and obtain new schedules by agent negotiations. Other applications in job shop scheduling include (Saad, Biswas, and Kawamura 1996; Gu, Balasubramanian, and Norrie 1997; Dewan and Joshi 2002). There are also some applications in material handling and inventory management (Kim, Heragu, Graves, and Onge 2002) and supply chain

management (Ertogral and Wu 2000; Qinghe, Kumar, and Shuang 2001). Only Dewan and Joshi (Dewan and Joshi 2002) compare their agent approach with an exact solution found by CPLEX. They conclude that centralized models are an unattractive choice compared to decentralized models because of computational inefficiency and degradation in the quality of solution with increasing problem size.

Within the area of physical distribution there are several publications on agent-based transport planning and scheduling. An interesting contribution comes from the artificial intelligence community, where (Fischer, Muller, and Pischel 1996) developed a simulation testbed for multi-agent transport planning, called MARS. They describe the information architecture and decision structure for a quite generic transport planning system and test their model on the traditional vehicle routing problem with time-windows where all orders are known in advance. No results are given for dynamic planning, stochastic processes (e.g. transport times) or multi-actor planning. In this paper, we cover dynamic, real-time planning under stochastic order handling times.

In (Hoen, Bragt, and Poutré 2002) a multi-agent system is presented for real-time vehicle routing problems with consolidation in a multi-company setting, where cargo is assigned to vehicles using a Vickrey auction. They show the advantage of truck decommitment, which is the option to break an agreement in favour of a better deal if another truck from the same company can handle the cargo. This transportation model differs from ours in the sense that vehicles can carry multiple loads and all loads have to be transported before the end of the next day. They use a simple bidding strategy, i.e. the vehicle bid equals the revenue of an order that is delivered minus the additional pickup, transportation and delivery costs. They do not consider time windows within a day and assume that . Also, they do not benchmark their results by comparison to a traditional (OR-based) approach.

Another interesting contribution comes from (Figliozzi, Mahmassani, and Jaillet 2003), who present a framework for the study of carriers' strategies in an auction marketplace for dynamic, full truckload vehicle routing with time windows. They use a Vickrey auction and a simple heuristic for generating bids, namely the additional costs of serving a shipment by appending it to the shipment queue of the truck. The focus in their results is on profit allocation rather than the efficiency of assignment decisions (no comparison to the performance of other planning techniques is given). Our research is different regarding the following aspects: (1) we will analyze the impact of various levels of intelligence when generating bids (2) we consider stochastic order handling times (3) we compare the overall planning performance to traditional OR planning heuristics.

In the area of railroad scheduling, (Böcker, Lind, and Zirkler 2001) present a multi-agent approach for real-time coupling and sharing of train wagons. In (Zhu, Ludema, and van der Heijden 2000) a multi-agent solution for air cargo assignment is considered. Although this paper contains an interesting agent-based application it does not provide detailed information on the design of a multi-agent system itself in terms of goals, behavior, pricing strategies etc.

2.4 Contribution to the literature

From our discussion above, we conclude that some first results on multi-agent planning and scheduling are available, also in the area of transportation. However, there is little known about the performance of agent-based transportation control compared with more traditional control methods and to which extent intelligence contributes to the overall system performance. Our contribution focuses on the following new issues:

- construction of a multi-agent planning framework for real-time planning of full truckload transport with soft time windows and incomplete information (demand, order handling times); we consider contracts for already accepted orders that allow for a high schedule flexibility if new orders are considered; further, agents have to learn the travel time and order handling time characteristics from historical data
- construction of several bid strategies with various levels of intelligence for vehicle agents and evaluate the impact on the overall system performance using discrete event simulation
- comparison of our multi-agent system to more traditional approaches for real time transport planning based on fast look-ahead rules and OR algorithms (serial scheduling)
- next to common performance characteristics such as the long run on-time delivery percentage, we also consider performance robustness, measured by standard deviation of the daily service levels; further, we analyze prices and profits in the agent framework, for example the impact of order characteristics (such as tightness of the time window) on the price.

3 Model, assumptions, terminology and notation

The essence of online vehicle control in transportation networks is the matching of open capacity with incoming orders, no matter whether it is done through a market place with agents or more classical control authorities. A general impression of the situation is given in Figure 1.

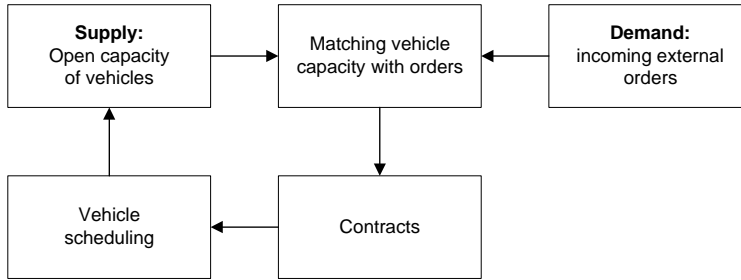


Figure 1: Demand driven online order assignment

The system dynamics is driven by the incoming orders that are not known beforehand. The matching of open vehicle capacity with the incoming orders can be done with planning rules by or agent based systems. Anyway, each matching leads to a contract between a fleet owner and a shipper. Execution of these contracts requires scheduling of the vehicles while taking the contract terms into account. Vehicle scheduling has its impact on the future availability of open capacity of vehicles and on the system dynamics. Further, we want to look at the system in a goal-oriented way. In this respect, the relation between the structure underlying matching and scheduling and several performance indicators is interesting. In the next subsections we shall discuss the elements as mentioned in more detail. Here we mention that we will mainly consider the case of one fleet owner and a homogeneous fleet of vehicles for sake of simplicity. We will discuss generalizations in Section 9.

3.1 Transportation network and demand

We consider a transportation network consisting of a set of nodes and a set of arcs connecting these nodes. Without loss of generality, we assume that a shipper operates from one node. However, at one node multiple shippers might operate. Orders to transport unit loads (full truckloads) between these nodes arrive one-by-one according to some unknown stochastic arrival process. We define an order by the following characteristics:

- the announcement time a ;
- its origin: node i ;
- its destination: node j ;
- the expected time $r \geq a$ at which the load can be picked up at the origin (earliest departure time, release time of the load at its origin);

- the due time $d > r$, i.e. the latest time at which the load should be delivered at its destination;
- the type of contract: in the contract the fleet owner agrees with the earliest departure time as a hard restriction and the due time as soft restriction with a penalty $c^p(T)$ costs charged to the transporter in case of tardiness T ; as for scheduling the transporter has full flexibility to set the planned pickup time t_1 and delivery time t_2 with the obligation to notify the shipper initially and later on in case of changes before the actual departure.

In the sequel we shall refer to a specific load with an index l . All order parameters, including the penalty function, are order dependent. Once an order is considered for execution by a transporter, it will often be referred to as a job being a logical terminology for a transporter. Note that other types of contracts with less flexibility for the transporter (for example with t_1 and t_2 fixed once they are set initially) can be imagined. Here we restrict ourselves to one type of flexible contract as mentioned. Generalizations will be discussed in Section 9.

The time τ_{ij}^f required to handle an external order from node i to node j driving full is a random variable. Variation in handling times may arise from traffic congestion, variation in loading and unloading times and waiting times at the nodes. The time to drive empty from node i to node j is a random variable τ_{ij}^e . The order processing times, handling times and empty travel times are unknown and should be learned before use in planning procedures.

The orders should be handled by a homogeneous set of vehicles \mathcal{V} . An idle vehicle that is not needed for a while can be parked at any node. Here we assume that parking capacity is sufficient. Some of the nodes may be used for parking purposes only, i.e. there are no orders from or to that node. Such a node is useful as a pre-positioning location, if it is close to several nodes where sufficient profitable orders are expected in the near future.

3.2 Vehicles, scheduling and system dynamics

To start with, we assume that an external order in process cannot be interrupted (no preemption). That is, a vehicle may not temporarily drop a load in order to handle a more profitable load and return later. However, due to the flexibility in the contracts, orders that have been accepted but not started yet may be rescheduled. Hence we assume that at each moment in time a vehicle has a list of external orders and a schedule to execute these external orders. Formally, we define a schedule for a vehicle as a sequence of “actions” of the following types: (i) move full along arc (i, j) (ii) move empty along arc (i, j) (iii) wait at node j until time t . The latter two options will be called internal orders. By definition, the first action in a schedule is in execution. Also, by

definition a schedule will always end with option (iii) at some location (for example a parking) with $t = \infty$. Note that given a list of K external orders there are $K!$ sequencing possibilities in the schedule and several possibilities for internal orders in between.

As for the lists of external orders and the schedules per vehicle, updating takes place at discrete moments in time through (a) completion of the first action in a schedule (b) matching a new external load with open vehicle capacity (c) another control action redefining the schedule(s) and / or the assignment of external orders to vehicles. In a discrete event simulation, the system dynamics consists then of following the evolution over time of the state consisting of the lists of external orders and the schedules per vehicle, as well as the lists per node of still open external orders. This underlying structure of the dynamics is the same for agent-based control or more classical control heuristics.

As for the dynamics and control, we assume that the system is stable in the long run, so that all orders can be handled. As for the control structure, we require that it works in such a way that it will also deal with orders that can only be delivered late. Furthermore, the vehicles are location aware and fleet owners are aware of the next node to be visited by their vehicles. As for communication, we assume that at any moment in time communication between shippers, vehicles and fleet owners is possible. Due to non-preemption of external orders in execution a full move can never be interrupted while move empty actions can always be interrupted.

3.3 Performance criteria, information, goal oriented design.

Several parties are involved in transportation networks: transporters, shippers and possibly, governmental agencies (public / private partnerships at main ports or terminals). As a consequence, evaluation of the system performance is typically a multi-criteria situation. Relevant criteria have to do with costs, service levels and sustainability. Here we use as key performance measures:

- Costs and profits per vehicle and costs and profits per order type.
- Fill rate, i.e. the fraction of orders that is delivered before the due time.
- Stability of the service level, measured by the standard deviation of the fill rates per simulation run.
- Percentage of kilometers driving loaded, being an indicator for energy waste and loss of vehicle capacity.

In our models, we assume that a vehicle $v \in \mathcal{V}$ faces the following transportation costs: (i) variable transportation costs c^t per time unit, both for full or empty driving (ii) variable waiting costs c^w per time unit. The fleet owner can set these cost parameters for his homogeneous fleet of vehicles and keep record of historical data as a check. Therefore waiting- and travel cost per time unit are equal for all vehicles within the same fleet. We assume that fixed costs are identical for all vehicles, so that we can ignore fixed costs when constructing schedules. Some remarks on generalization to different vehicle types and / or different fleet owners, different rates for empty and full driving, are made in Section 9.

An important aspect in practice is that not all information on performance criteria is necessarily open information to all parties. We assume that costs information such as c^t and c^w are in principle not open to the shippers, but private to the fleet owner. On the other hand, in other situations with multiple fleet owners, shippers can also have private information on costs per order type charged by different fleet owners.

Our main task in paper is to compare the “efficiency” of traditional control heuristics and agent-based control systems. For this question, one should keep in mind that for a fair comparison one has to know with what goal orientation in mind the system was designed. Goal orientation in this sense means a set of weights for the criteria as introduced. One can choose to put emphasis on the service levels (fill rates) as we did in previous work, cf. (Heijden, Ebben, Gademan, and van Harten 2002) or put more emphasis on transportation costs, as we will do in this paper in our agent-based approach. Of course, one should expect that this difference in design emphasis reflects itself in the results on performance as measured in simulations. But, even so the results are pretty surprising, as we will show.

4 Agent-based planning concepts

In our agent-based planning concept, we assign vehicles to jobs using a market-like negotiation protocol that implicitly coordinates the agents’ decisions. The definition of such an agent-based planning concept depends on three key choices: (i) which agents to distinguish with their tasks and goals, (ii) which products (services) to trade, and (iii) which market mechanism (auction) to define. We will address these three issues below. To make the concept operational, the goal-directed behavior of each agent has to be defined. In Section 5 we introduce several variants of agent behavior.

4.1 Agent types

Because we are creating a market mechanism between buyers and sellers, it is logical to introduce at least one type of agent for the resources (vehicles) and one for the orders to be processed.

An elementary structure is to define one agent per vehicle and one agent per order. However, it can also be useful to introduce agents at a higher level in view of information and / or coordination actions concerning multiple vehicles of a fleet owner or multiple orders. This leads us to the structure in Figure 2:

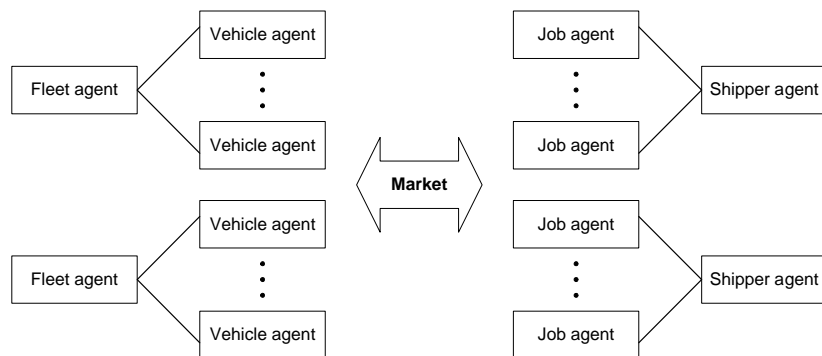


Figure 2: Agent structure for transportation networks

A vehicle agent has the task to deploy the vehicle capacity in order to maximize its profit. A job agent has the task to arrange transportation of the corresponding load before the due time at minimal costs. All vehicle agents and job agents may meet on the marketplace. This structure has the highest degree of decentralization. The vehicle- and job agents themselves will do the basic assignment decisions of vehicles to orders. The vehicle agents themselves maintain their own local lists of contracts and schedules as introduced in the previous modeling section. Hence the solution to the global scheduling problem emerges from the local decision-making of the agents. Thus, one complex overall plan is replaced by many smaller and simpler plans.

The introduction of hierarchy may improve the coordination between agents. We can define hierarchy both at the job level and at the resource level. At the job level, a shipper agent can be responsible for a set of orders. A possible task is to reallocate the transport capacity that has been acquired such that their orders are handled before the due times at the lowest costs. For example, they may exchange an order that has been scheduled already, but for which transportation did not yet start, for a rush order that comes in. To this end, they have full information on all orders under their control and all transport capacity that has been acquired for these orders. At the

resource level, a fleet agent can be responsible for a subset of vehicles. They have full information on the position and local schedules of all vehicles under their control. Therefore, they can reassign vehicles to the jobs that they have acquired such that the profit of the fleet is maximized.

Although a hierarchical concept is interesting, we choose in our basic variants for a fully decentralized concept. The main reason is to keep it simple and to examine to which extent such a simple agent-based concept can already meet the performance of traditional OR based planning methods. However, in order to provide information to the vehicle agents and job agents, we will use respectively a fleet agent and shipper agent. The fleet agent keeps track of the travel times of his vehicles and the handling times of the orders in the transportation network. To this end, the fleet agent uses a simple exponential smoothing procedure for the mean and the mean absolute deviation (to estimate the standard deviation) of both the travel time and the handling time for each origin-destination pair, cf. (Silver, Pyke, and Peterson 1998). The shipper agent keeps track of all the bids his job agents receive, that is, he calculates the expected cost for routes starting from his node and estimate the handling times for these routes. The latter is necessary in order to estimate the latest departure for jobs using Dynamic Threshold (see Section 5.2). In another extension called Trade (see Section 5.1) we give a simple coordination task to the fleet manager to get a first impression of the effect of such coordination intelligence.

4.2 Products traded

To create a marketplace, a product definition is necessary. Here we will use the flexible product definition as introduced in Section 3.1 (the contract type in the order definition). As a logical consequence, the job agent will start the auction.

One should realize that other product- and order definitions are possible. This might then have repercussions for the initiation of auctions. For example, we may consider as product definition the transport capacity of a unit load that is available at node A at time t_1 and that may be used during a time period T . For such a product, a vehicle agent would be the logical initiator for an auction. Such a product may be useful for a job agent, because it provides the flexibility to reserve capacity for future jobs with some arbitrary destination. However, bidding is harder for vehicle agents who do not know the vehicle location at time $t_1 + T$. Also, we may trade a transport capacity of size N vehicles that can be used in some time interval $[t_1, t_2]$. Such a bulk trade may be advantageous for fleet management as a whole, but it is not suitable for a fully decentralized agent-based planning concept. As another option, we may trade transport capacity with size one

unit load from node A to node B that is picked up at time t_1 and delivered at time t_2 . Although this definition fits well with the order definition, it hampers flexibility for dynamic reallocation of capacity when additional (rush) orders arrive. All alternative options also lack an intrinsic way to deal with uncertainty, which is coped with by the tardiness penalty introduced in the product definition that we use.

4.3 Auctioning mechanism

Most implementations of distributed scheduling systems have used auctions to create schedules (Clearwater 1996; Wellman and Walsh 2001). Auctions are also the most efficient means of selling the object when the seller does not know the value of the product. However, there is a large number of different auction mechanism to choose from. Common auction types are:

- Bargaining, this is a one-on-one negotiation protocol where all trading partners contact each other individually; therefore, it is a communication intensive and time-consuming approach that is less suitable for real-time application as in the problem that we consider.
- Open outcry auctions consist of multiple bidding rounds where all bids are known to each bidder. We can distinguish two types of open outcry auctions, the English and the Dutch (or falling click) auction. In an English auction, bidders sequentially raise their bids until nobody is willing to bid higher; the object is sold at that final price. In a Dutch auction the price starts at a high level and is reduced until one of the bidders declares that he is willing to pay that price. Obviously, these mechanisms are information intensive and time consuming as well.
- Sealed-bid auctions can be used where every bidder submits his bid only once and the best bid is selected; special cases are the first-price sealed-bid auction where exactly the price offered is paid, and the Vickrey auction in which the bidder receives the price of the one but best offer (second-price sealed-bid). A nice property of the Vickrey auction is the following: it can be shown that under mild conditions the optimal bid is the net cost price of the bidder, who will make profit from the margin between the two best bids, cf. (Vickrey 1961). Therefore, it provides a natural mechanism for acceptable profits.
- Double-sided auctions are used to connect multiple buyers and multiple sellers; examples are Call Markets and Continuous Double Auctions. In Call Markets a central impartial auctioneer collects bids from all buyers and offers from all sellers and matches them. In a

Continuous Double Auction, a group of buyers and a group of sellers simultaneously and asynchronously announce bids and offers: at any time the sellers are free to accept any buyer's bid and the buyers are being free to accept any seller's offer.

In this paper we choose for sealed bid auctions because they are easy to implement avoiding bid iteration over time. We select the Vickrey auction as mechanism in our paper. Using a Vickrey auction enables us to concentrate on the transportation control variables themselves rather than on learning and rationality issues of the agents.

We implement the market mechanism as follows. Each time an order l arrives, a corresponding job agent is created that starts an auction by issuing an announcement to all vehicles to bid for a contract of the type as specified in Section 3.1. In response, each vehicle agent $v \in \mathcal{V}$ creates one bid b for this order l , thus confirming agreement with the contract terms. The bid comprises an initial price, an expected departure time and an expected arrival time. Note that due to the contract flexibility, changes in these initial price, expected departure time and expected arrival time are allowed. If this occurs, the original price will be reduced by the increase in the penalty costs.

Note that in our paper we choose for a design where a vehicle may issue only one bid, which is the best bid he can offer taking scheduling into account. Another design possibility is that a vehicle agents sends a bid for every different way to schedule this new job. In that case the vehicle agents would not require any knowledge of the penalty cost function. However the amount of information exchange would increase dramatically.

To determine the winner of the auction, the job agent evaluates all bids and sends a message to all the vehicle agents. The winner of this auction receives a grant message and all the other bidders a reject message. As a variant we shall also consider job agent designs where all bids can be declined by the job agent. The rationale is that a better bid may arise by re-auctioning in the future, because new (yet unknown) orders may arrive that prevent some empty travelling.

Bid calculation by vehicle agents and bid evaluation by job agents can be done in a lot of different ways. We will discuss several variants for this agent behavior in Section 5.

5 Variants for agent behavior

5.1 Vehicle agents

Vehicle agents place bids to transport jobs and therefore have to calculate prices. We define several variants for the vehicle agent behavior with various levels of intelligence. We will test the impact of additional intelligence on the system performance in a numerical experiment (Section 8). In this section, we first discuss the generic cost calculation of a schedule by vehicle agents. Next, we present three variants for scheduling jobs by vehicle agents in a completely decentralized framework. Finally, we present a level of coordination over vehicles by a fleet owner agent.

To define the cost structure, we first introduce the notation S_v^* for the current schedule of vehicle v . We indicate each alternative new schedule n for vehicle v by S_v^n . Under the Vickrey auction mechanism, the vehicle agent will issue a bid that is exactly equal to the expected additional costs for doing the new job. Therefore, the bid price for a vehicle v will be equal to the minimum cost for all alternative schedules n . As mentioned in Section 3.2, the costs depend on the additional time needed to move the load (linearly), possible extra waiting time (linearly) and the total time by which the latest departure times are exceeded (general function):

$$\hat{P}_{v,l} = \min_n \left(c_v^t \Delta T_{v,l,n} + c_v^w \Delta W_{v,l,n} + \sum_{\forall o \in S_v^n} c_o^d (\Delta D_{v,o,n}) \right)$$

Here $\Delta T_{v,l,n}$ is the expected additional travel- and handling time required for vehicle v in schedule alternative n to transport the order l . By $\Delta W_{v,l,n}$ we denote the increase in the expected waiting time for vehicle v in schedule alternative n until the last order in the schedule has been served. Obviously, this value may be negative if the new job can be inserted nicely in a gap in the local vehicle schedule. By $\Delta D_{v,o,n}$ we denote the change in the total tardiness for order o in schedule alternative n of vehicle v . Penalty cost are being calculated when the actual arrival time is later than the due time. Still, it may be worthwhile to accept extra penalty costs, if the new job fits very nice in the schedule because of this, so that a cheap bid can be constructed. To compensate for late delivery of other jobs, the bid of vehicle v should cover the possible penalty costs for all jobs that are already in his schedule S_v^* and the new job.

In general, the creation of bids depends on the internal order scheduling of the vehicle agent. All our vehicle agents will use a few simple basic scheduling rules: (i) if a last external order is in a schedule is finished at node i and no external order is available yet, the vehicle is sent to a

predefined parking node $p(i)$ (pre-positioning) (ii) if an external order is finished at node i and the next external order in the schedule occurs at j , then the vehicle move empty to j and wait at this destination if necessary. Additionally, we distinguish three ways of scheduling jobs.

The simplest method is to schedule a new order always at the end of the already existing schedule. We will refer to this behavior by the AgentEnd method. In this method, $\Delta D_{v,l}$ always equals the tardiness for order l because a new job does not influence the expected arrival times of jobs already in his schedule. Additional waiting time $\Delta W_{v,l}$ will only occur at the origin of the new job l . The additional travel time $\Delta T_{v,l}$ equals the handling time of order l plus possibly the time needed to reposition the vehicle from the end location of schedule S_v^* to the start location of job l .

A second option is to allow a vehicle agent to insert a new job at any position in the existing schedule S_v^* without altering the order of execution for the other jobs. We will refer to this option as the AgentInsert method. Now the bid calculation is somewhat different, because the scheduling of this new job can affect the expected arrival times of previous jobs.

Finally, the AgentTSP method can also change the order of jobs in the joblist by solving Traveling Salesman Problem (TSP). Obviously, an exact solution is only realistic if the job list of a single agent is relatively short; otherwise we have to rely upon well-known fast heuristics for the TSP, such as tabu search, cf. (Gendreau, Hertz, and Laporte 1994). In our experiments we use a depth-first, branch and bound algorithm, where we used an upper bound found with AgentInsert to test the lower bound for the remaining branch. Given the limited number of jobs in the list of a vehicle (less than ten) and the quality of the upper bound, an optimal solution of the TSP is possible within a reasonable amount of time.

Because of the dynamic environment where new orders arrive, travel times are stochastic and the vehicle agents will stick to contracts made in the past, the assignment of jobs to vehicles is not guaranteed to be pareto-optimal (Wellman 1992). Therefore we introduce, additionally to the three vehicle agenttypes, some simple coordination by a fleetowner agent as an option, called Trade (TR). The fleet owner agent provides the vehicle agents the possibility to exchange jobs with each other. We assume that this can only take place when a vehicle, after unloading at a certain terminal, has to travel empty to another terminal. Whenever this occurs, the fleet agent searches for the agent in question for another vehicle agent, which has a job from this terminal to the same next terminal that is already released. Whenever savings occur, the job will be exchanged. We assume that this exchange results in pure profit for the fleet agent and not for the job agent in

question.

5.2 Job agents

The job agents have as main task to evaluate all bids and to decide which bid to accept. We consider two variants for the job agent behavior. The first variant is simply to accept the best bid received by all vehicle agents. Because we included the penalty cost in the bid prices, this evaluation is only based on the price.

A second variant is to refuse all bids if the lowest bid is still above a certain threshold value. In that case, the job-agent restarts the auction at a later time, hoping for a better bid. The motivation for this behavior is that prices in different auction rounds will fluctuate due to changes in the available overall transportation capacity and vehicle schedules. When a certain job agent has a lot of time remaining until departure, he will wait until he receives an attractive price, i.e. a price higher than his reserve price. As the deadline for dispatch comes nearer and an acceptable bid still has not been encountered, the job agent may increase the threshold price in order to get transportation. We call this variant DynamicThreshold (DT). The decision of the job agent is (1) to set an initial threshold price for the first auction round (2) to determine the timing and threshold price for the next round if all first round bids are above the initial threshold price.

In order to use this extension, the job agents must have insights in the cost and handling times for their routes. This information comes from the shipper agent. As mentioned in Section 4.1, they keep track of the travel times and prices that are being paid for delivering their jobs. Job agents can use this information in their bid acceptance strategies to establish threshold prices for each auction round.

The bid acceptance under the DynamicThreshold variant works as follows. For the timing between successive auctions for the same job, we take a fixed period R . It is logical to relate the threshold price to the maximum number of auction rounds N before the job has to be transported with $N = \lfloor (d - t - a)/R \rfloor$ with d the due date, a the first announcement time of the order and t the expected handling time as obtained from the shipper agent. In the last round, any bid is accepted in order to force the job to be served. Without loss of generality, we assume that R is such that always $N \geq 2$ (if not, the DynamicThreshold variant coincides with the first variant discussed in this section in which the lowest bid is always accepted).

The threshold price p_N for the last auction round is always infinite, i.e. any offer is accepted. The first threshold price p_1 will be equal to a certain minimum price P_{min} and the threshold price

for the second last auction round p_{N-1} equals a maximum price P_{max} . These values P_{min} and P_{max} can be based on historical data provided by the shipper agent. We will use two different pricing strategies: linear and quadratic. For the linear strategy, the threshold price p_r in the round r is given by:

$$p_r = P_{min} + \left(\frac{P_{max} - P_{min}}{N - 1} \right) (r - 1) \quad \text{for } r = 1, \dots, N$$

For the quadratic pricing strategy we define:

$$p_r = P_{min} + \left(\frac{P_{max} - P_{min}}{(N - 1)^2} \right) (r - 1)^2 \quad \text{for } r = 1, \dots, N$$

We expect that it this DynamicThreshold policy will provide more flexibility for doing rush orders. However, the probability of a delay will increase by restarting an auction later in time. A better pricing strategy would be to calculate the probability of receiving a better bid in the future and the expected value of such a bid. For now we only consider the linear and quadratic pricing strategies.

6 Traditional OR based heuristics as benchmark

Some traditional operations research methods have been developed for real-time job scheduling in transport networks. We will use two of the methods from (Heijden, Ebben, Gademan, and van Harten 2002) as benchmark for our agent system, because the focus in that paper is on a similar problem as we consider here.

Both methods that we will consider are hierarchical methods. At the top level, vehicles are distributed amongst nodes based on actual and expected orders, without detailed job assignment. At the node level, vehicles are assigned to jobs, where only the vehicles can be used that are assigned to that node by the top level. The advantage of such an approach is that a complex schedule is decomposed into two simpler decisions. One of these decisions, assignment of vehicles to jobs, should be done in real time. The other decision, distribution of vehicles amongst nodes, should be done frequently, but not necessarily real time, because it is a higher-level decision without immediate consequences. We will use two methods that fit within the hierarchical framework, namely hierarchical control and integral planning.

Under hierarchical control, the top level distributes vehicles using a simple priority rule, based on a central order list and a central overview of all vehicle positions and current activities. First,

we calculate the latest departure time for each order as the due time minus an offset for the expected handling time (loading, transportation, unloading) and the variation in the handling time. Next, we sort the order list orders in increasing order of latest departure times. We process the list sequentially. To each order, we assign the vehicle that can be available at the earliest point in time. If a vehicle is waiting at or driving to a different node, the top level issues an empty vehicle repositioning orders with corresponding latest dispatch time to that node.

At the node level, we have a list of orders to be dispatched (with latest departure time) and a list of empty vehicle dispatch orders (with latest dispatch time). Every time a vehicle becomes available at the node, we choose the highest priority order from both lists. For efficiency reasons, we try to combine empty dispatch orders with load dispatch orders if possible. For example, if it is most urgent to dispatch a job from node A to node B, we look in the order list of node A whether there is a (lower priority) load dispatch order from A to B, and if so, the vehicle takes this load on its trip. Hence the node level operates independently of the top level, but within the conditions set by the top level. We refer to (Heijden, Ebben, Gademan, and van Harten 2002) for more details.

In integral planning approach, we construct a better planning to distribute vehicles over nodes. To this end, we use serial scheduling (Ebben, van der Heijden, and van Harten 2004), where different priority rules are being used to create a sequence of jobs, which are virtually assigned to vehicles. At the node level, we still decide on the assignment of jobs to vehicles. However, to maintain the structure of the vehicle distribution planning from the top level, the node level has to handle all orders in a sequence that has been prescribed by the top level. In that sense, we move responsibility from the node level to the top level, hoping to receive a better performance in return in terms of fill rate and distance traveled empty.

The aim of a hierarchical control concept as described above is to construct a more flexible and fast schedule compared to a fully centralized concept. The difference between centralized, hierarchical and heterarchical (agent based) control structures is illustrated in Figure 3.

Of course, a hierarchical control concept has some advantages compared to purely central control. It requires less data exchange and is capable of reacting quicker to unexpected events because of the allocation of tasks and responsibilities to two hierarchical levels. However, this hierarchical decomposition of control does not take into account the different roles of various independent stakeholders that negotiate on their mutual services and corresponding prices. Besides, a key difference with the agent approach is that under the hierarchical planning all order and

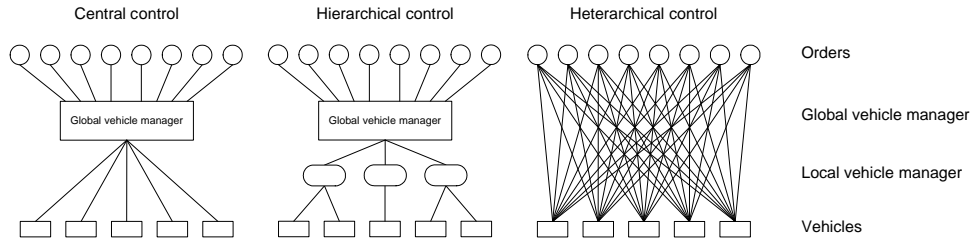


Figure 3: Control structures

vehicle information should be centrally available and that a central vehicle distribution plan is constructed.

7 Experimental setting

In this section we discuss the experimental design. First, we describe the network characteristics (7.1). Then we give the parameter settings that we fix over all experiments (7.2) and the factors that we vary over the experiments and their setting in (7.3).

7.1 Network characteristics

To test the proposed multi-agent concepts and to compare them with other control methods, we use a setting inspired by a case study on a proposed underground transportation system (OLS) near Amsterdam Airport Schiphol, the Netherlands, as inspiration (Heijden, Ebben, Gademan, and van Harten 2002). In this system, Automated Guided Vehicles (AGVs) carry cargo between terminals. We use an artificial network layout based on this OLS-case consisting of five terminals, see Figure 4, where the distance between terminals is measured in meters.

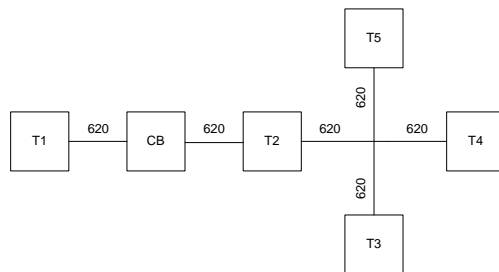


Figure 4: Network structure

Each terminal has an internal track structure and consists of a number of docks where AGVs can be loaded or unloaded. As performance indicators, we will use:

- the service level SL, defined as the overall percentage of jobs delivered before their due time
- the standard deviation of the service level for simulation runs as a measure of robustness of the control method,
- the time that vehicles are driving loaded as percentage of the total driving time (excluding waiting time)
- the profit per vehicle and the job costs (for agent-based planning).

The route between a given origin-destination pair is fixed. When a vehicle is not needed for a while, it may be parked in a terminal (local parking) or on a central parking area (CP) that can be considered as a network node. Orders arrive according to a (non)stationary Poisson process (cf. Section 7.3). Travel times between terminal entrances are deterministic and known in advance because they only depend on the distance and speed of vehicles. Although the distances are deterministic, the handling times show variation due to the following causes:

- Variations in loading and unloading times.
- Waiting times at the terminals.
- Dock-dependent distances on terminals, where AGVs can only drive with reduced speed for sake of safety.

Therefore, we treat the handling times as random variables. The mean and standard deviation of the handling times are dynamically being updated using a standard exponential smoothing procedure, see (Silver, Pyke, and Peterson 1998). The fleet manager/ agent keeps track of all handling times and the corresponding estimates are available to all vehicles under its control.

7.2 Fixed parameter settings

Of course, our performance will be sensitive to the parameter settings in our pricing and evaluation functions in the agent-based approach. The vehicles have a speed of 6 m/s outside the terminals. Given the distances between nodes as shown in 4 (meters) and handling and delay on terminals, we have expected handling times ranging from 7 minutes (T3 to T4) to 13 minutes (T3 to T1) with a standard deviation ranging from 35 seconds (T1 to T2) to 1 minute (T5-T3). Note that the stochasticity in the handling times is significant, but not very large.

We set the cost per time unit for traveling for all vehicles to 1. It is reasonable in our setting that the waiting cost per time unit are equal to the historical average profit per time unit. This information is estimated by the fleet agent. The cost per time unit for tardiness is chosen such, that the job agent will almost always prefer a job without delay. We establish these cost at 10. We set the parameters of DynamicThreshold as follows. P_{min} is equal to the mean price for a specific route, P_{max} to the maximum price paid so far for this route and the fixed time interval R between the auction rounds is set to 5 minutes.

7.3 Experimental factors

In this section, we discuss the factors that we will vary in our simulation experiments and we give the parameter settings. An overview of the general experimental factors can be found in Table 1. As already mentioned, two hierarchical control methods already have been developed for this OLS-case. By using simulation, we will compare our multi-agent model with these methods. We will explain the settings of the factors demand structure and fleet size below.

Factor	Range
Control method	LocalControl, SerialScheduling, AgentControl
Agent scheduling	AgentEnd, AgentInsert, AgentTSP
Demand structure	Stable, Dynamic, Highly Dynamic
Mean arrival rate	Low (0.5), High (3.0) (products per minute)
Fleet size	Large, Small
Agent extension	Trade, DynamicThreshold
Threshold pricing options	Linear, Quadratic

Table 1: Experimental factors

The experimental factor *demand structure* concerns the distribution of transportation flows. We distinguish three cases: Stable, Dynamic, and Highly Dynamic. In the stable demand structure, (i) the order arrival rates are identical for all origin destination pairs, (ii) the order arrival rates are constant over the day, and (iii) all orders have the same time window of 45 minutes (no rush-orders). In the dynamic demand structure, (i) the order arrival rates are still identical for all origin destination pairs, (ii) the order arrival rates vary over hours of the day within a band with of 25% around the mean arrival rate as given in Table 1, and (iii) the time windows are drawn from a discrete probability distribution as given in Table 2.

The highly dynamic demand structure is similar to the dynamic demand situation, except that the order arrival rates are not identical for all origin destination pairs. We have used the probabilities of Table 3 to establish an origin for a job. Once an origin is found, we draw a

Percentage	Time-window
50	90 min.
30	60 min.
20	30 min.

Table 2: Time-windows

destination out of the remaining nodes with equal probability.

Origin	Probability
T1	0.20
T2	0.25
T3	0.35
T4	0.15
T5	0.05

Table 3: Order arrival rates

We further assume that the announcement times a for jobs are all equal to the earliest departure time r . Therefore the time-windows can be defined as the time between the first auction for a job and the due time d .

Regarding the fleet size, we have to adjust the fleet size to the demand volume in order to achieve a high service level, let us say around 98%. In case of low mean arrival rate (0.5), we use 7 vehicles, whereas we either use 32 vehicles (small fleet size) or 33 vehicles (large fleet size) in case of high mean arrival rate (3.0). Hence we have three combined settings for the experimental factors *Mean arrival rate* and *Fleet size*. Given this experimental design, we have 9 different network settings (combinations of *Demand structure*, *Mean arrival rate* and *Fleet size*). We use these network settings to compare the three variants of agent scheduling to the two basic control methods (9x5=45 experiments). Because the highest impact of agent intelligence is found for the highly demand situation, we test the agent extensions and threshold pricing options (see Section 5.2) for the highly dynamic demand structure only. For the factors *agent extension* and *threshold pricing option*, we use four additional settings: trade, dynamic threshold with linear pricing, dynamic threshold with quadratic pricing and the combination of trade with dynamic threshold, linear pricing. Hence we have 3x3x4=36 additional experiments and the total number of simulation experiments equals 81.

We use a replication / deletion approach for our simulations (cf. (Law and Kelton 2000)), where each experiment consists of five runs of six days, each including a one-day warm-up period. The simulation batch size is small in order to cope with the long simulation times, especially for the instances with AgentTSP. However, the variances in our experiments show that there is

enough significance to compare the different control mechanisms.

8 Numerical results

In this section, we present the results of our simulation experiments. First, we compare decentralized agent-based planning with a single auction round (no dynamic threshold) to the two traditional heuristics (8.1). Next, we examine the impact of two extensions, namely coordination between vehicles by the fleet agent (Trade) and the dynamic threshold variant for the job agents 8.2. Finally, we analyze the relation between the job prices and the job characteristics as well the distribution of the vehicle profits 8.3.

8.1 Comparison of agent-based concept to traditional OR heuristics

When comparing our agent-based concept to the OR heuristics of Section 6, we use the service level (SL) and percentage driving loaded (DL) as key performance indicators, see Table 4.

AGVs	Scenario	Stable		Dynamic		Highly dynamic	
		DL	SL	DL	SL	DL	SL
7	Local	63	99.61	65	98.83	63	96.28
	Serial	65	99.99	67	99.64	63	97.93
	AgentEnd	73	99.66	72	94.52	70	92.59
	AgentInsert	78	99.92	78	99.48	73	97.82
	AgentTSP	77	100	77	99.76	73	98.96
32	Local	81	100	83	100	-	-
	Serial	87	100	88	100	-	-
	AgentEnd	90	99.86	90	95.64	81	72.07
	AgentInsert	94	99.97	93	99.76	82	91.57
	AgentTSP	94	99.99	93	99.88	82	91.81
33	Local	80	100	81	100	76	83.74
	Serial	86	100	87	100	77	90.52
	AgentEnd	90	99.84	90	97.59	81	90.21
	AgentInsert	94	99.96	93	99.86	81	97.05
	AgentTSP	94	99.99	93	99.89	81	98.01

Table 4: Simulation results: comparing control methods

The hierarchical control methods with 32 vehicles and highly dynamic demand are not able to handle all orders that arrive per day and therefore the service levels will go to zero. The results for other simulation settings are all stable in the long run.

We see that the service levels for our agent approach and the hierarchical control methods (Local, Serial) are very close to each other. In other words, the service levels of our best agent approach are not significantly worse than the service levels for the two hierarchical control methods.

The service levels (but also the percentages driving loaded) are especially worse for the hierarchical control methods if the demand is (highly) dynamic and has a high volume. When we look at the different demand volumes, we see that the hierarchical heuristics only result in a slightly higher service level for the stable cases with a high number of vehicles. The more vehicles there are within the network, the more the hierarchical heuristics can benefit from their hierarchical structure and local flexibility.

With regard to the percentage of driving loaded we see that our agent approach always perform better than the hierarchical control methods. These differences are significant because the 95% confidence intervals for the hierarchical methods and our agent approach do not overlap.

Regarding the scheduling method of the vehicle agent, we observe that AgentEnd performs considerably worse than AgentInsert and AgentTSP, particularly under highly dynamic demand. However, the latter two variants performs quite similar. Because of the dynamic system behavior, solving a TSP problem exactly has apparently little added value.

Another advantage of our agent control is that it seems to be more robust than the two hierarchical control methods. From the standard deviation in service levels per day over runs of five days for the highly dynamic case (Table 5), we see that our agent control is less sensitive against fluctuations in the stochastic variables such as demand volume, loading- and unloading times.

	7	33
Local	5.7	24.2
Serial	3.2	12.4
AgentEnd	3.7	4.8
AgentInsert	2.2	3.2
AgentTSP	1.3	2.6

Table 5: Simulation results: standard deviation in service level per day

From these results, together with the results from Table 4, we conclude that our agent methods are performing very well with respect to the percentage of driving loaded and robustness, both in terms of different demand structures and for different demand volumes. Next we investigate the impact of additional intelligence.

8.2 The impact of additional intelligence

To examine the impact of additional intelligence, we show the key performance indicators DL and SL for agent systems with or without vehicle coordination (Trade) and DynamicThreshold (both

linear and quadratic) in Table 6.

AGVs	Scenario	AgentEnd		AgentInsert		AgentTSP	
		DL	SL	DL	SL	DL	SL
7	Normal	70	92.59	73	97.82	73	98.96
	Trade	71	95.1	74	98.17	73	99.14
	DP-lin	73	94.94	75	97.56	74	98.57
	DP-Qdr	74	95.52	75	97.39	74	98.55
	TR-DP-lin	73	95.59	75	97.77	74	98.73
32	Normal	81	72.07	82	91.57	82	91.81
	Trade	82	81.63	82	91.61	82	92.93
	DP-lin	81	82.02	82	90.86	82	91.96
	DP-Qdr	81	81.62	82	90.84	82	91.94
	TR-DP-lin	82	82.21	82	90.8	82	91.83
33	Normal	81	90.21	81	97.05	81	98.01
	Trade	81	91.24	82	96.84	82	98.08
	DP-lin	81	91.43	81	96.87	81	97.99
	DP-Qdr	81	91.42	81	96.46	81	97.82
	TR-DP-lin	81	91.56	82	96.71	82	97.7

Table 6: Simulation results: additional intelligence

We see that using additional intelligence always improves the performance in case of the AgentEnd scheduling method, especially in terms of service levels. With 7 vehicles the difference in service level is statistically significant with confidence level of 98% and 90% for 32 and 33 vehicles. The service level with the more intelligent scheduling methods (AgentInsert and AgentTSP) can only be improved by using Trade, although the differences are not statistically significant at confidence level 98%.

An improvement in the percentage driving loaded is only achieved in case of a small fleet size. Because the standard deviation in driving loaded over runs is always below 1%, these differences are statistically significant with a confidence level of almost 100%.

It might be surprising to see that the extensions Trade and DynamicPricing do not improve the percentage of driving loaded when we have a larger number of vehicles. The reason for this is that there is very little room for improvement. We will illustrate this with a simple calculation of the upper bound on the percentage of driving loaded. Let us relax the problem by assuming that all orders are known in advance, there are no time windows and all travel- and handling times are deterministic. Then penalty costs are not relevant, so the problem reduces to the minimization of the total empty travel time under flow conservation constraints. Therefore, we can easily calculate

the maximum percentage of driving loaded by solving the following mathematical program:

$$\begin{aligned} & \min \sum_{i,j} X_{i,j} \tau_{i,j}^e \\ & \text{subject to} \\ & \sum_k X_{k,i} + D_{k,i} = \sum_j X_{i,j} + D_{i,j} \quad \forall i \end{aligned}$$

where $\tau_{i,j}^f$ the expected empty travel time from i to j , $D_{i,j}$ the number of jobs for this route and the decision variable $X_{i,j}$ denote the minimum number of times the route from i to j has the be travelled empty. The maximum percentage of time driving loaded DL^* is now given by:

$$DL^* = \frac{\sum_{i,j} D_{i,j} \tau_{i,j}^f}{\sum_{i,j} X_{i,j} \tau_{i,j}^e}$$

When we use the average handling times and demand data resulting from our simulation experiments for the highly dynamic case as input for the mathematical program, we find an upper bound of 84% for the percentage of driving loaded. This leads us to conclude that our agent control is capable of providing high quality solutions in terms of service levels and percentage driving loaded.

8.3 Analysis of job costs and AGV profits in an agent framework

Although we focus on the performance indicators percentage of driving loaded and service level, it is also interesting to take a short look at the prices in the system in case of agent-based control. In this section, we examine two performance measures:

1. The mean profit for AGVs; because we use a Vickrey auction at vehicle level, the profit of a vehicle for a specific job equals the difference between his winning bid and the second-highest bid.
2. The mean relative cost per terminal, defined as the mean price paid for jobs starting from a certain terminal compared to the cost price for these routes. Here the cost price equals the time for loading, transportation and unloading, so excluding empty travel times.

The experiments were done for 7 and 33 AGVs in case of the highly dynamic demand structure. The mean profit for AGVs, together with the standard deviation in profit per AGV can be found in Table 7.

	Mean profit for AGVs	Std. Dev.
7 AGVs and AgentEnd	12.3	0.56
7 AGVs and AgentInsert	13.8	0.80
33 AGVs and AgentEnd	7.0	0.67
33 AGVs and AgentInsert	8.0	0.80

Table 7: Simulation results: mean profits for AGVs

From these results, we see that the mean profit for AGVs increases by using a smarter scheduling method. Also the standard deviation will be slightly higher. Increasing the number of AGVs will also increase the competition which in turn results in lower transportation cost. This will lead to lower profits for the vehicles and savings for the terminals.

Due to the network flow and geographical position the relative cost will differ per terminal. These percentages, together with the mean relative cost for all terminals can be found in Table 8.

	T1	T2	T3	T4	T5	Mean
7 AGVs and AgentEnd	29.5	41.0	62.3	47.8	48.8	45.9
7 AGVs and AgentInsert	23.5	35.3	62.5	36.5	25	36.6
33 AGVs and AgentEnd	6.3	24.8	52.3	5.3	2.3	18.2
33 AGVs and AgentInsert	9.0	19.8	59.3	-6.3	-15.8	13.2

Table 8: Simulation results: relative cost per terminal

To explain the contents of Table 8; the relative cost 62.5 for terminal T3 with 7 AGVs means that this terminal has to pay on average 62.5% more than the cost price for every route starting from T3. This surcharge is due to the time driven empty to pick up the load that could not be charged to other jobs.

Of course, all terminals will profit from an increase in competition. However some nodes profit more than others. These results can be explained using Table 3 and Figure 4. Particularly interesting are the negative percentages for terminal T4 and T5. In case of 33 AGVs, these terminals can profit from the fact that they have relatively little out going orders en relatively a lot of incoming orders. When a job agents announced an auction for an order with origin terminal T4 or T5 and a wide time-window, the probability that this job can be nicely inserted in a gap in some vehicle schedule is high. Therefore the insertion cost can be lower than the cost price for this route.

Finally, we will explore the influence of the flexibility defined as the length of the time window in more depth. Therefore we simulate a scenario with a dynamic demand structure with 10 different time-window lengths, 32 AGVs using AgentInsert with DynamicThreshold. The relation between this order flexibility and the resulting price can be found in Figure 5.

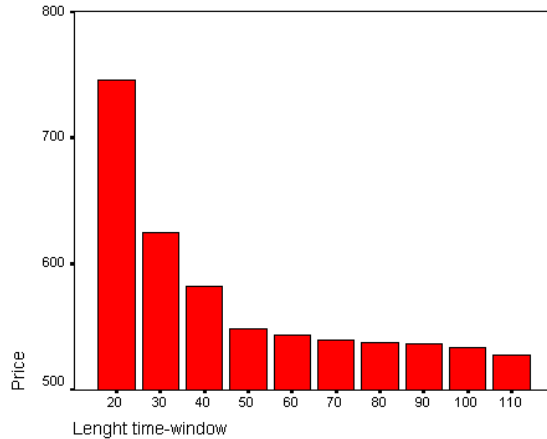


Figure 5: Mean price for route T4-T1

Because it is easier to schedule a job with a wide time window, the expected cost will be lower. Using DynamicThreshold further increases the savings with more flexibility.

9 Conclusions, generalizations and further research

In this paper, we have proposed a distributed agent-based solution to the transportation scheduling and planning problem. This agent-based control method yields a number of advantages. Firstly its robustness, it is less sensitive to fluctuations in demand or available vehicles than more traditional transportation planning heuristics of Section 6. Secondly it provides a high level of flexibility by solving local problems locally. Thirdly it provides online decision-making by using auction mechanisms. Fourthly, this multi-agent technique also provides a key for the division of cost and profits across the network. From our simulation experiments, we can also conclude that our agent approach yields a high performance in terms of vehicle utilization and service level.

Regarding further research, we will focus on model generalizations and improvements of the agent-based concept in the near future. Regarding generalizations, we can replace the single fleet owner in our simulation experiments by multiple fleet owners with different cost structures and possible different strategies. Also, we can introduce a heterogeneous fleet where vehicles have different cost structures and are possibly suitable to transport specific products. Other generalizations such as different types of contracts (for example renting vehicle capacity for a fixed time period) and different roles for the shippers and fleet owners (more hierarchical coordination) will also be investigated.

Improving the intelligence of our agents is especially concerned with their learning capabilities

and pro-active behavior. For example, the vehicle agents can use dynamic pricing. Similar to the dynamic pricing used to sell airline seats, vehicles can price their services based on the available capacity. For now, we will only use dynamic pricing at the job side. Although vehicles can schedule more jobs in advance, our model is still myopic. Vehicle agents only consider the direct cost of doing certain jobs. In the future we want our agents to base their current action not only on their previous actions but also on a prediction of the future behavior of the system. In this way, we can take opportunity losses for arriving at a terminal without a next order with low expectations for load in the near future into account. To do so, we will develop formal methods for estimating the value of certain actions, for example using approximate dynamic programming (cf. (Godfrey and Powell 2002)). Then, we expect vehicles to drive pro-actively to other nodes with higher expected future revenues, or to calculate the changes of driving empty from certain terminals and include these cost in their bid prices.

Our plans for future research on pricing mechanisms can be divided into three parts. Firstly, we will develop formal methods for dynamic pricing and dynamic thresholds. For example, vehicle agents can charge different prices depending on their available capacity and job agents use a dynamic threshold policy depending on a estimated distribution of bid prices. Secondly, we also want to explore different pricing mechanisms, for example the double-sided auction. Finally, we also want to improve the applicability of our system by looking at pilots for collaborative planning in practice. We believe that further development of these multi-agent systems will yield promising and applicable models for controlling tomorrow's transportation networks.

References

- Böcker, J., J. Lind, and B. Zirkler (2001). Using a multi-agent approach to optimise the train coupling and sharing system. *European Journal of Operational Research* 134, 242–252.
- Cardon, A., T. Galinho, and J.P. Vacher (2000). Genetic algorithms using multi-objectives in a multi-agent system. *Robotics and Autonomous Systems* 33, 179–190.
- Clearwater, S. (1996). *Market-based control: a paradigm for distributed resource allocation*. Singapore: World Scientific.
- Desrosiers, J., Y. Dumans, M.M. Solomon, and F. Soumis (1995). Time constrained routing and scheduling. In T. M. C. M. Ball, M.O. and G. Nemhauser (Eds.), *Network Routing*, pp. 35–139. North-Holland, Amsterdam.

- Dewan, P. and S. Joshi (2002). Auction-based distributed scheduling in a dynamic job shop environment. *International Journal of Production Research* 40(5), 1173–1191.
- Ebben, M.J.R., M.C. van der Heijden, and A. van Harten (2004). Dynamic resource-constrained vehicle scheduling. *European Journal of Operational Research* forthcoming.
- Ertogral, K. and S.D. Wu (2000). Auction-theoretic coordination of production planning in the supply chain. *IIE Transactions* 32, 931–940.
- Figliozzi, M.A., H.S. Mahmassani, and P. Jaillet (2003). A framework for the study of carries strategies in an auction based transportation marketplace. *Transportation Research Record* forthcoming.
- Fischer, K., J.P. Muller, and M. Pischel (1996). Cooperative transportation scheduling: an application domain for dai. *Journal of Applied Artificial Intelligence. Special issue on Intelligent Agents* 10(1), 1–33.
- Fischer, M. (1995). Vehicle routing. In T. M. C. M. Ball, M.O. and G. Nemhauser (Eds.), *Network Routing*, pp. 1–33. North-Holland, Amsterdam.
- Gendreau, M., A. Hertz, and G. Laporte (1994). A tabu search heuristic for the vehicle routing problem. *Management Science* 40, 1276–1290.
- Gendreau, M. and J.Y. Potvin (1998). Dynamic vehicle routing and dispatching. In T. Crainic and G. Laporte (Eds.), *Fleet Management and Logistics*, pp. 127–157. Kluwer, Academic Publishers.
- Giani, G., F. Guerriero, G. Laporte, and R. Musmanno (2003). Real time vehicle routing: solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research* 151, 1–11.
- Godfrey, G.A. and W.B. Powell (2002). An adaptive dynamic programming algorithm for dynamic fleet management, ii: Multiperiod travel times. *Transportation Science* 36(1), 40–54.
- Gu, P., S. Balasubramanian, and D.H. Norrie (1997). Bidding-based process planning and scheduling in a multi-agent system. *Computers and Industrial Engineering* 32(2), 477–496.
- Heijden, M.C. van der, M. Ebben, N. Gademane, and A. van Harten (2002). Scheduling vehicles in automated transportation systems. *OR Spektrum* 24, 31–58.
- Hoen, P.J. 't, D.D.B. Van Bragt, and J.A. La Poutré (2002). Bidding with decommitment in a multi-agent transportation model. Technical report, Center for Mathematics and Computer

Science (CWI), Amsterdam, The Netherlands.

- Jennings, M.J., K. Sycara, and M. Wooldridge (1998). A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems* 1(1), 7–38.
- Kim, B., S. Heragu, R.J. Graves, and A.St. Onge (2002). Intelligent agent based modeling of an industrial warehouse problem. *IIE Transactions* 34(7), 601–612.
- Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research* 59, 345–358.
- Law, A.M. and W.D. Kelton (2000). *Simulation Modeling and Analysis* (3d ed. ed.). New York: McGraw-Hill.
- Parunak, H. Van Dyke (1998). Practical and industrial applications of agent-based systems. *Environmental Research Institute of Michigan (ERIM)*.
- Powell, W.B. (1996). A stochastic formulation of the dynamic assignment problem, with an application to truckload motor carriers. *Transportation Science* 30(3), 195–219.
- Powell, W.B. and T.A. Carvalho (1998). Dynamic control of logistics queueing networks for large scale fleet management. *Transportation Science* 32(2), 90–109.
- Psaraftis, H. (1988). Dynamic vehicle routing problems. In B. G. Assad and A. (Eds.), *Vehicle Routing: Methods and Studies*, pp. 223–248. North Holland, Amsterdam.
- Qinghe, H., A. Kumar, and Z. Shuang (2001). A bidding decision model in multiagent supply chain planning. *International Journal of Production Research* 39(15), 3291–3301.
- Saad, A., G. Biswas, and K. Kawamura (1996). Performance evaluation of contract net-based heterarchical scheduling for flexible manufacturing systems. *International Journal of Automation and Soft Computing, Special Issue on Intelligent Manufacturing Planning and Shop Floor Control*.
- Silver, E.A., D.F. Pyke, and R. Peterson (1998). *Inventory management and production planning and scheduling*. New York: Wiley.
- Toth, P. and D. Vigo (2002). *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications. Philadelphia.
- Vickrey, W. (1961). Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance* 16(1), 8–37.

- Wellman, M.P. (1992). A general-equilibrium approach to distributed transportation planning. In *Proceedings of AAAI-92*, pp. 282–290.
- Wellman, M. and W. Walsh (2001). Auction protocols for decentralized scheduling. *Games and Economic Behavior* 35, 271–303.
- Wooldridge, M. and N.R. Jennings (1995). Agent theories, architectures, and languages: A survey. In *Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages*, pp. 1–39. Amsterdam, The Netherlands.
- Zhu, K., M. W. Ludema, and R.E.C.M. van der Heijden (2000). Air cargo transport by multi-agent based planning. In *Proceedings of the Thirty-third Annual Hawaii International Conference on Systems Sciences*, Maui, Hawaii.