DE GRUYTER
OPEN

JAISCR, 2013, Vol. 3, No. 1, pp. 27 – 40
DOI 10.2478/jaiscr-2014-0003

AGENT-BASED DISPATCHING ENABLES AUTONOMOUS GROUPAGE TRAFFIC[1]

Max Gath, Stefan Edelkamp and Otthein Herzog

Institute for Artificial Intelligence, Center for Computing and Communication Technologies,
Am Fallturm 1, 28359 Bremen, Germany

Abstract

The complexity and dynamics in groupage traffic require flexible, efficient, and adaptive planning and control processes. The general problem of allocating orders to vehicles can be mapped into the Vehicle Routing Problem (VRP). However, in practical applications additional requirements complicate the dispatching processes and require a proactive and reactive system behavior. To enable automated dispatching processes, this article presents a multiagent system where the decision making is shifted to autonomous, interacting, intelligent agents. Beside the communication protocols and the agent architecture, the focus is on the individual decision making of the agents which meets the specific requirements in groupage traffic. To evaluate the approach we apply multiagent-based simulation and model several scenarios of real world infrastructures with orders provided by our industrial partner. Moreover, a case study is conducted which covers the autonomous groupage traffic in the current processes of our industrial parter. The results reveal that agent-based dispatching meets the sophisticated requirements of groupage traffic. Furthermore, the decision making supports the combination of pickup and delivery tours efficiently while satisfying logistic request priorities, time windows, and capacity constraints.

1 Introduction

The complexity and dynamics in logistic processes have been increased due to shorter product life cycles, the rising number of product variants, and the growing number of transnational links and dependencies of the production processes between companies. As a result, the requirements of transport processes are increasingly complex through shorter transit times, the individual qualities of shipments, and higher amounts of small-sized orders. In addition, the rising traffic density on transport infrastructures and growing demands wrt. sustainable transportation encourage logistic companies to improve the efficiency of their processes.

In the last decades, numerous efficient heuristics and meta-heuristics have been developed for the transportation domain like ant systems, tabu-search, simulated annealing and genetic algorithms, just to name a few, e.g., [1, 2, 3, 4, 5, 6]. However, central planning and control in dynamic and complex logistic processes is limited due to the requirements of flexibility and adaptability to changing environmental influences.

In autonomous logistic processes, the decision making is shifted from central, hierarchical planning and control systems to decentralized, heterarchical systems [7]. Intelligent software agents represent logistic entities, e.g., containers or vehicles. Thus, they are able to plan and schedule their way through the logistic network by themselves. The agents act on behalf of represented objects and

[1] Preliminary parts of this paper were presented at the 5th International Conference on Agents and Artificial Intelligence in Barcelona, Spain, in February 2012, and at the IEEE Symposium on Computational Intelligence in Production and Logistics Systems in Singapore, in April 2013.

try to reach the objectives assigned by their owners. Consequently, relevant information is directly linked to products. For instance, an agent representing a shipment is aware of its individual weight, volume, and its designated place and time of arrival. The agents apply and share this knowledge through communication and negotiation mechanisms with other agents in order to optimize the efficiency of processes and utilization of resources. By delegating planning and control processes to decentralized entities the overall problem is split into smaller problem instances that can be solved optimally due to smaller computational complexity.

We present an autonomous dispatching system that accommodates the requirements of groupage traffic. The paper is structured as follows: Section 2 describes the logistic problem that was extracted in an initial process with a forwarding agency and specifies the problem formally. In Section 3, we present the implemented multiagent system including the interaction protocols as well as the decision making processes of the agents. Thereby, we contemplate at the effects of increasing the problem complexity by adding constraints that have to be dealt with in groupage traffic. In Section 4 we present the multiagent simulation system PlaSMA [8].

For an evaluation study, we use PlaSMA in combination with customer orders based on real-world data provided by our industrial partner. The experimental setup and the results are provided in Section 5. Finally, we conclude the results and provide future research directions.

2 Groupage Traffic

In groupage traffic, several orders with less-than-truckload (LTL) shipments are served by the same truck to decrease total cost. In pickup tours, trucks transport loads from their origin to a local depot where the shipments are consolidated to enable economical loads. Through LTL networks the load is transported to a depot in the destination area where each good is delivered to its final destination through onward carriage. The complexity of process planning is even increased by changing amounts and individual qualities of shipments like weight, volume, priority, and value. Handling the complexity in real life is aggravated by the high degree of dynamics that result from unexpected events. The exact amount and properties of shipments are not known in advance. Actual capacities are only revealed and exactly determined while serving tasks. Furthermore, undelivered loads in pre-carriage decrease truck capacities in onward carriage. To react to changing traffic conditions and delays on incoming goods departments, it is essential to adapt tours and timetables with respect to actual capacities. The main processes are illustrated in Fig. 1.

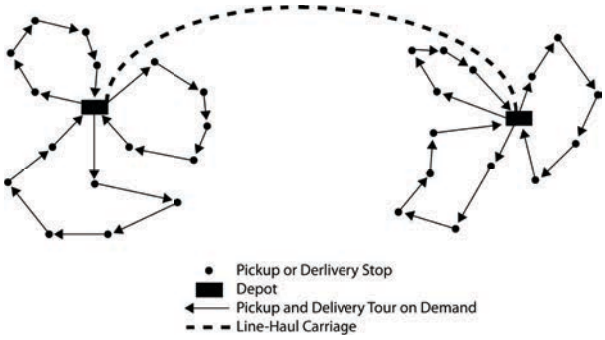
● Pickup or Derlivery Stop
▬ Depot
← Pickup and Delivery Tour on Demand
- - - Line-Haul Carriage

Figure 1. The main processes in groupage traffic.

The dynamics and complexity of planning and scheduling processes require efficient, proactive, and reactive system behavior to improve the service quality while ensuring time and cost efficient transportation. Regarding the dispatching process in pre- and onward carriage the general problem can be mapped to the well-known Vehicle Routing Problem (VRP)[9]. In general, the VRP is concerned with determining minimum costs tours for a fleet of vehicles to satisfy customer requests at different destinations. The start and end point of each tour is the depot.

Definition 1 (Vehicle Routing Problem) *Let V denote a set of vehicles and S a set of service requests. Given the costs $c^v_{i,j}$ for a vehicle $v \in V$ for traveling from $i \in S$ to $j \in S$ and choosing indicator variables*

$$x^v_{i,j} = \begin{cases} 1, & \textit{if } (i,j) \textit{ is part of the vehicle } v\textit{'s tour} \\ 0, & \textit{otherwise} \end{cases} \tag{1}$$

the general objective function of VRP is

$$\min \sum_{v \in V} \sum_{j \in S} \sum_{i \in S} c^v_{i,j} \cdot x^v_{i,j} \tag{2}$$

with subject to

$$\sum_{v \in V} \sum_{i \in S} x_{i,j}^{v} = 1 \, for \; all \; j \in S \tag{3}$$

$$\sum_{v \in V} \sum_{j \in S} x_{i,j}^{v} = 1 \, for \; all \; i \in S \tag{4}$$

$$\sum_{v \in V} x_{i,j}^{v} = \{0,1\} \; for \; all \; i,j \in S \tag{5}$$

$$\sum_{v \in V} \sum_{j \in S} \sum_{i \in S} x_{i,j}^{v} \leq |Y| - 1 \, for \; all \; Y \subseteq S. \tag{6}$$

Moreover, further constraints, e.g., time windows, capacities, as well as time consumption at the warehouse/customer, must be taken into account. If l_s denotes the latest delivery time, t_s the time consumption of the boarding or deboarding process, r_s the release time at $s \in S$ and $time_{i,j}^{v}$ vehicle v's time for driving from i to j, then

$$x_{i,j}^{v} = 1 \Rightarrow l_j \geq r_i + t_i + time_{i,j} \tag{7}$$

has to be satisfied. In addition, we have to ensure that the maximum capacity of a vehicle is not exceeded at any time. Let CC_s^v denote the current capacity of vehicle v at stop $s \in S$ and MC_v the maximum capacity of vehicle v, then we require

$$CC_s^v \leq MC_v \text{ for all } s \in S, v \in V. \tag{8}$$

In VRPs containing exclusively pickup or delivery orders the current capacity is decreasing or increasing monotonously. The combination of pickup and delivery tours leads to an increasing complexity due to fluctuating capacities. Consequently, the sequence of a tour has a significant impact on a truck's load. Moreover, the complexity is aggravated by the high degree of dynamics that result also from unexpected events, such as an exact amount and properties of incoming orders are not known in advance. Actual capacities are only known while serving tasks. To react to these changing conditions and incoming orders, it is essential to adapt tours and timetables in respect to actual capacities in almost real-time.

In general, it is not possible to transport all orders that are available on a certain day. However, the quality of service is an important factor to satisfy the economic objectives. The transportation of so-called premium services must be guaranteed with respect to their time windows while considering other hard constraints, e.g., the capacity of vehicles. Premium services have to be delivered on the date of receipt until 8am, 10am, 12am, or not later than 5pm. Within a logistic transport network the participating forwarding agencies have to pay high amounts of penalties if they do not adhere the agreed upon commitments.

Definition 2 (Premium Stop) *Pickup or delivery stops on which premium services have to be picked up or delivered are defined by the boolean function*

$$p_i = \begin{cases} true, & if \; i \; is \; a \; premium \; stop \\ false, & otherwise. \end{cases} \tag{9}$$

On the other hand, *conventional* orders can be delivered up to two days after their arrival without effects to guaranteed service times. There is no need to order an external freight carrier for them. Moreover, shipments which have to be delivered until 5 pm can be processed by own vehicles, e.g., by shifting the transport of another conventional order to the next day. As a result, the objective function of the VRP includes not only to find a solution with minimum costs, but tours that maximize the number of premium services with highest priority:

$$max \sum_{i \in S} \sum_{j \in S} p_i \cdot x_{i,j} \tag{10}$$

and conventional orders with second highest priority:

$$max \sum_{i \in S} \sum_{j \in S} \neg p_i \cdot x_{i,j}. \tag{11}$$

To illustrate the range of the search space, one can model the problem by an urn model including sampling with replacement and consideration of the sequence. For instance, for each shipment $s \in S$ exactly one $v \in V$ is selected from the urn. Afterwards, it is put back and the procedure continues with the next shipment. As a result, there are $|S|^{|V|}$ possible allocations of shipments to vehicles, if $|V|$ denotes the number of vehicles and $|S|$ the number of shipments (stops).

The complexity is even increased by the high degree of dynamics. Tours and routes have to be adapted throughout the entire operation, to include new orders in existing plans and to deal with unexpected events. Consequently, the dispatcher requires solid decision support systems that are capable to cover the high requirements of groupage traffic.

3 A Mulitagent System for Planning and Control in Groupage Traffic

The advantages of applying multiagent systems are high flexibility, adaptability, scalability, and robustness of decentralized systems through problem decomposition and proactive, reactive, and adaptive behavior of intelligent agents [9]. Therefore, mulit agent systems are especially applied to open, unpredictable, dynamic, and complex environments. There are many examples of multiagent systems within logistic processes for resource allocation, scheduling, optimization, and controlling. Agent-based commercial systems are used within the planning and control processes of containerized fright [10, 11]. Team formation and interaction protocols have been designed for efficient resource allocation [12] as well as for concurrent negotiations between logistic service providers and service consumers [13]. Agent-based systems have optimized planning and control processes within dynamic environments [14, 15]. Other application ranges have been provided for industrial logistic processes [16]. A comprehensive survey in research on autonomous logistic processes is provided in [17] and [12].

3.1 Agent Interaction and Execution

Similar to Schuldt [12], who developed an agent-based system for the optimization of warehouse capacities in containerized freight, agents represent transport vehicles and orders. While the general architecture of each *vehicle agent* is identical, the agents differ in their individual properties. For instance, represented vehicles vary in their capacities, work schedules, and speed limits. Likewise, each order agent considers the unique characteristics of its represented shipment such as the pickup and delivery location, weight, value, time windows, and premium service constraints. The goal of *order agents* is to find a proper transport service provider for carrying the shipment from the depot to the destination or from its origin to the depot while satisfying given time window constraints. *vehicle agents* negotiate with *order agents* to maximize the number of carried shipments while satisfying all relevant constraints and premium service priorities.

Firstly, each *order agent* sends a cluster request to a cluster-agent which collects all requests until the operational processes of the trucks start in the morning. Then, the cluster-agent starts the rough planning by applying a K-Means clustering [18]. The goal is to assign each shipment to one of k available vehicles. An initial allocation is computed by an office-specific mapping of postal codes to trucks that is provided by the forwarding agency. Next, the cluster algorithm considers the coordinates of the pickup or delivery location of the shipment. Therefore, the algorithm consolidates shipments that have pickup or delivery locations in nearby districts in the same cluster. Further constraints are neglected, but considered in the detailed planning processes. A limitation of conventional information systems, which support the dispatchers of forwarding agencies, is that the rough planning is currently merely based on a static office-specific mapping and does not consider the effective order situation. Contrary, the agent-based clustering solution covers seasonal fluctuations and the daily varying order volume. Furthermore, it computes a uniform distribution of orders to trucks with respect to the location of the orders.

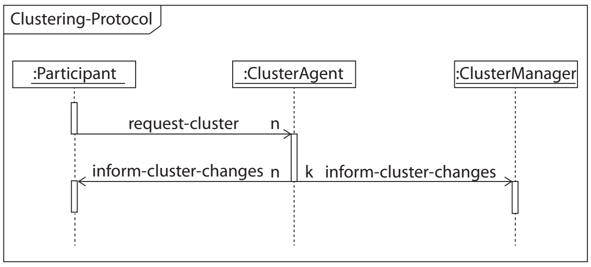

Figure 2. The interaction protocol for rough planning that is applied by the cluster-agent.

The implemented interaction protocol is shown in Figure 2. If n denotes the number of shipments, the communication effort of the protocol is $O(n^2)$. It is a stable interaction protocol that prevents manipulation of the outcome by a participant. Moreover, it ensures that confidential data of any shipment is only sent to agents with appropriate access rights, e. g., truck agents receive only information about assigned shipments.

To reduce the computational effort after the initial allocations are computed, shipments at the same pickup or delivery location constitute an order. The orders' properties are defined by the shipments and an order contains either pickup or delivery requests. Afterwards, each *vehicle agent* starts a detailed planning process. On the one hand, the *vehicle agent* considers the truck's capacity, the driving times which are depended on the type of the street and the respective speed limits, as well as the individual capacities of the shipments such as the weight, priority, time windows, handling times, and obviously the pickup or delivery location. On the other hand, the agent optimizes the objective functions to reduce costs and determine efficient solutions. Therefore, with highest priority it maximizes the amount of transported premium services. With second highest priority it maximizes the processing amount of conventional orders and with third highest priority it identifies the shortest path for visiting all stops. As a result, the problem refers to a generalization of the NP-Hard Traveling Salesman Problem (TSP) [19] which is a single vehicle variant of the VRP [20]. The solver which meets the special requirements in groupage traffic is described in Section 3.2.

After the detailed planning step of each *vehicle agent*, several orders may not be serviced by a vehicle. As the decision making process of *vehicle agents* prioritize premium services, this affects conventional orders more frequently. Nevertheless, conventional orders may be transported by another truck. Thus, the responsible agent acts in the same way like agents representing dynamically incoming orders: They start a contract-net protocol [21] negotiation with the truck agents. The agent sends a *call for proposal* message to available trucks. Next, the *vehicle agents* compute proposals by determining their additional cost for transporting the shipment. In order to schedule new orders also while transporting other shipments, the *vehicle agent* has to consider all relevant changes in the environment as well as its internal state, e.g., the position of the truck and current capacity restrictions. For instance, picked-up shipments reduce available capacities and the position of the vehicle affects the determination of shortest ways and optimal routes. Consequently, the planning and decision making processes of the agents are linked directly to their execution behaviors. Computed cost are sent back to the *order agent* that chooses the transport provider with the least cost. If it is not possible to satisfy the orders' requirements, a *refuse*-message is sent by the *vehicle agent*.

To transport a premium service instead of conventional orders or another premium service with less cost, already accepted orders (that have not been boarded yet) may not be included in the new plan and have to be rescheduled. Afterwards, affected *order agents* negotiate with other transport service providers again. Potentially, this results in a series of computation and communication intensive negotiations between agents to achieve small improvements. To reduce this effect (especially if several shipments are processed consecutively within a short time window and the global allocation changes significantly) the agent waits for a certain period of time before it starts the negotiation procedure. Delays and not delivered shipments may also affect current plans of vehicle agents. Therefore, the *vehicle agent* validates its plans consecutively and adapts the route if necessary. New plans may effect the executing actions of the trucks. Therefore, the truck agent checks during driving, if the next stop has changed and if necessary it adapts the tour. In real processes as well as in the simulation the handling processes (boarding and deboarding of shipments) must not be interrupted. This requirement is satisfied by not adopting plans that interfere with the running handling processes.

3.2 Agent Decision Making

Within the negotiation, *vehicle agents* have to compute proposals and decide which service request has to be satisfied. These cost are based on the additional distance that has to be driven by the vehicle. To calculate the distance, agents must solve a generalization of the *NP-Hard* Traveling Salesman Problem (TSP) [19] that is a single vehicle variant of the VRP defined in Section 2. More formally, there are n different stops i with $i \in \{1,2,\ldots,n\}$ and all distances between two stops i and j are specified by $c_{i,j} \in^{+}$ and $c_{i,i} = 0$ for $1 \leq i,j \leq n$. Feasible solutions are permutations of $(1,2,\ldots,n)$ with the additional constraint that the first (and thus the last stop to be visited) is the depot. Real transport infrastructures are commonly represented by directed graphs. In this case the problem is an asynchronous TSP for which we search an optimal tour. In order to meet the special requirements in groupage traffic, the problem changed into a maximizing-minimizing problem that can be described as follows.

Definition 3 (Optimal Tour with Premium Stops) *The optimal tour of the asymmetric TSP must be feasible and fulfill the following requirements ordered by their priorities for the variables*

$$x_{i,j} = \begin{cases} 1, & \textit{if } (i,j) \textit{ is part of the tour} \\ 0, & \textit{else} \end{cases} \tag{12}$$

1. *Maximize the number of transported premium services:* $max \sum_{i=1}^{n} \sum_{j=1}^{n} p_i \cdot x_{i,j}$
2. *Maximize the number of visited stops:* $max \sum_{i=1}^{n} \sum_{j=1}^{n} x_{i,j}$
3. *Minimize the total cost of the path:* $min \sum_{j=1}^{n} \sum_{i=1}^{n} c_{i,j} \cdot x_{i,j}$ *subject to*

(a) $\sum_{i=1}^{n} x_{i,j} = 1$ *for all* $j \in \{1,\dots,n\}$

(b) $\sum_{j=1}^{n} x_{i,j} = 1$ *for all* $i \in \{1,\dots,n\}$

(c) $x_{i,j} = \{0,1\}$ *for all* $j,i \in \{1,\dots,n\}$

(d) $\sum_{j \in S} \sum_{i \in S} x_{i,j} \leq |S| - 1$ *for all* $S \leq \{1,\dots,n\}$

Branch-and-bound (BnB) is an Operations Research (OR) programming paradigm used to solve hard combinatorial optimization problems. To apply *Branch-and-bound* (BnB), we extend depth-first search (DFS) by upper and lower bounds.

An initial upper bound can be obtained by constructing any solution, e.g., established by a greedy approach. Unfortunately, for lager TSPs the branching process consumes a lot of time to determine a greedy solution. As with standard DFS, the first solution obtained might not be optimal. With *depth-first BnB* (DFBnB), however, the solution quality improves over time together with the global upper bound U until eventually the lower bound $L(u)$ at some node u is equal to U. In this case an optimal solution has been found, and the search terminates. It is ensured that the algorithm terminates when a fixed number of expansions is exceeded. As a result, we have an anytime algorithm that finds better solutions the more time it keeps running. It returns a valid solution if it is interrupted. If no further improvement is possible, the optimal solution is found.

```
Constraint-TSP-DFBnB($n, depot, X$)
  Initialize upper bound $U$
  $maxExp \leftarrow X$
  $exp \leftarrow tour \leftarrow best \leftarrow \emptyset$
  call $DFS(n, depot, 0, U)$
  return $bestPath$

DFS($n, u, g, U$)
Input: Node $u$, path cost $g$, upper bound $U$
Side effects: Update of threshold $U$, solution path $bestPath$

  $d \leftarrow depth(u)$
  $tour_d \leftarrow u$
  if $(cost(best) < cost(tour))$
    $best \leftarrow tour$
  if $(d = n-1)$
    if $(g + c_{u,depot} < U)$
      $best \leftarrow tour$
      $U \leftarrow g + c_{u,depot}$
  else
    $\{v_0, \dots, v_n\} \leftarrow nextcities(u)$
    for each $j$ in $\{1, \dots, n\}$
      $exp{+}{+}$
      if $(Constraint(v_j) \wedge maxExp \geq exp)$
        if $(g + h(v_j) < U)$
          call $DFS(n, v_j, g + c_{u,v_j}, U)$
```

DFBnB Algorithm for the Constraint TSP.

The pseudo-code implementation is shown in Algorithm 3.2. At the beginning of the search, the procedure is invoked with the start node and with the upper bound U set to some reasonable estimate (it could have been obtained using some heuristics; the lower it is, the more can be pruned of the search tree, but in case no upper bound is known, it is safe to set it to a maximum value). The tour and the number of expansions are maintained globally. Another global variable *best* keeps track of the current best solution path. If a tour with lower cost is found this tour is saved as the best found result. The cost function has to consider also the priorities of premium services and conventional orders. It is obvious that an increasing depth leads to a rising number of included orders. If all orders are included $(d = n-1)$, the current cost are saved as upper bound and further pruning rules can be applied to accelerate the search. If the algorithm terminates before the maximum number of expansions is reached, the optimal solution with the maximum number of shipments as well as the shortest path is returned.

At each node the visited cities, the current time, premium service information, and the current capacity of the tour are saved as a computer word. All bit-vector operations (setting, clearing of bits, check for subsumption) run in $O(1)$ which is is the standard assumption in the RAM model. This enables constraint checks in $O(1)$ at each node, because each check is done by bit-vector comparison[2]. For instance, the constraint check, if stop j has already been visited is shown in Figure 3. Similarly, the other constraints such as time windows are checked.

```
(((used >> j) & 1L) > 0)
```

Figure 3. An example for the implementation of a constraint check in the procedure *Constraint*(v_j) in Algorithm 3.2.

In pure pickup or delivery problems checking the capacity is done by comparing the sum of all transported shipments to the maximum capacity of the truck. However, this holds not for the mixture of both problems as capacities are varying. Delivered shipments release capacities for picking up more freight. Therefore, two more variables for the capacity of all deliveries and the maximum capacity have to be maintained at each node.

Theorem 1 *Checking the capacity constraints of the truck with simultaneous pickup and deliveries is done in constant time and space.*

Proof Saving at each node the maximum capacity χ_M that the truck has reached on the tour, the current capacity χ_C of the truck, χ_D the sum of all delivery shipments that have to be loaded at the depot, and let ω denote the weight of the order at stop s, on each node χ_M is updated with

$$\chi_M = \begin{cases} \max(\chi_C, \chi_M + \omega), & \text{if } s \text{ is a delivery stop} \\ \max(\chi_C, \chi_M), & \text{otherwise.} \end{cases} \tag{13}$$

If τ denote the maximum capacity of the truck, the capacity constraints for adding a new order are satisfied by checking

$$\tau \geq \chi_M. \tag{14}$$

Consequently, all operations can be implemented by a single bit-vector comparison. No backtracking is necessary to avoid an overcharge of trucks on predecessor nodes by adding new delivery stops to the tour. □

Table 1. An example for checking the capacity constraints of a truck in the tree structure.

depth	χ_C	χ_M	is pickup stop	plan is valid (Equation 14)
0	0	0	-	-
1	1	1	yes	true
2	2	2	yes	true
3	2	3	no	true
4	2	4	no	true
5	2	5	no	false
5	3	4	yes	true
6	4	4	yes	true

Table 1 gives an example which enables constraint checks in constant time and space. In this example, we assume that the maximum capacity of the truck is 4 and the weight of each shipment is 1. To avoid backtracking, we save χ_C and χ_M at each node. If the truck picks up a shipment, the current capacity is increasing. Adding a delivery stop in the plan does not effect the current capacity because loading the shipment was not considered up to this point. Nevertheless the truck has to load the shipment at the depot before starting the tour. Therefore, χ_M is increasing. Consequently, it is not possible to add a delivery stop in depth 5 although other pickup stops are included afterwards.

Figure 4 illustrates the search tree of the branch and bound algorithm and shows the bit-vector implementations of the visited cities, the current time, premium service information, and the current capacity of the tour at each node.

[2] If n denotes the number of stops and w the length of the hardware dependent computer word, we assume that the complexity is of $O(\frac{n}{w}) = O(1)$.

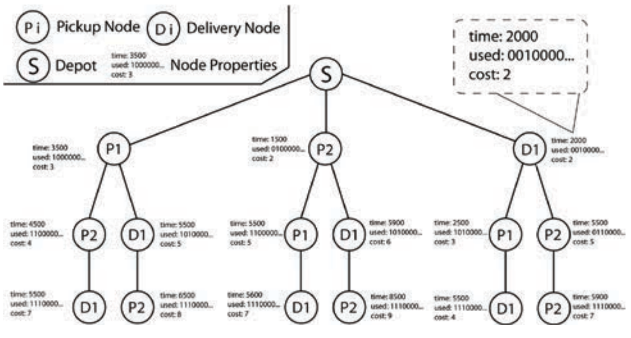

Figure 4. An example of the branch and bound search tree for solving constraint TSPs with pickups and deliveries.

Serving pickup and deliveries simultaneously does not effect the optimality of the decision making process, if the algorithm is not interrupted by exceeding a maximum number of expansions.

Theorem 2 *Setting the number of allowed expansions to ∞ the solver is optimal for admissible lower bounds, the above pruning rules, and the objective functions specified in Equation 10, 11, and 2.*

Proof If no pruning was taking place, every possible solution would be generated, so that the optimal solution would eventually be found. Pruning rules that satisfy capacity and time window constraints, cut off infeasible branches from the search tree so that the solution will be optimal. In addition, the search tree is only pruned by the upper bound U, if the maximum depth is reached and all cities are still visited (this satisfies Equation 10 and 11). If the tree is pruned by finding a better lower bound, as for admissible weight functions exploring the subtree cannot lead to better solutions than the one stored with U. □

4 Agent-Based Simulation

Changing logistic processes often requires hardware investments, negotiations and communication with involved persons, and implies risks for the company, e.g., the benefit could be lower than expected. Applying multiagent-based simulation (MABS) to procure well-founded assessments of the impact of potential changes is an accurate cost and time reducing method before the deployment of multiagent systems. This holds especially for scenarios with run-time agent interactions that cannot be predicted in advance [22].

PlaSMA [8] (see: http://plasma.informatik.uni-bremen.de/) is an agent-based event driven simulation platform that has been designed for modeling, simulation, evaluation, and optimization of planning and control processes in logistics. It extends the FIPA-compliant Java Agent DEvelopment Framework (JADE) [23] for agent communication and coordination and provides discrete time simulation that ensures correct synchronization while satisfying time model adequacy, causality, and reproducibility [24].

The transport infrastructure within the simulation environment is modeled as directed graph. Nodes represent traffic junctions or logistic sources and sinks, while edges represent different types of ways, e.g., roads, motorways, trails, and waterways. They have additional parameters that determine the maximum allowed velocity and the distance of an edge. Therefore the simulation system enables fine-grained modeling of road sections whose maximum allowed velocity is changing.

In order to simulate industrial and transport processes reliably, it allows the simulation of real-world infrastructures and supports their import from OpenStreetMap. This enables the integration of highly detailed graphs with up to 300,000 edges and 150,000 nodes. After clipping, a user defined map section and choosing relevant types of edges (e.g., roads, waterways, highways and railways) several preprocessing procedures are started to reduce the complexity of the overall graph without effects on the granularity of the infrastructure model. For example, redundant nodes as well as nodes that are only important to mark the course of the roads are deleted. The result is a directed graph which includes information about the real worlds speed limits, the distance as well as the type of an edge.

Particularly, shortest-path searches on real infrastructures are cost-intensive operations. However, computing the distance matrix between cities is essential for solving the TSP on a shortest path reduced graph (see Section 3.2). Consequently, we implemented Dijkstra's single-source shortest paths search [25] that is realized by a memory-efficient joint representation of graph and radix heap nodes [26]. Therefore, the algorithm is optimal and has linear time complexity as long as the distances between cities are bounded by a small constant[3]. Computing a distance matrix requires a lot of shortest-path searches with a fixed starting node and is a time critical procedure. Hence, we adapted the search procedure and saved the last visited nodes within a hash map as well as in the radix heap structure as long as the start node has not changed. While processing new search requests we check in constant time, if the shortest path to the node was already found and retrieve the corresponding node from the radix heap. Otherwise, the shortest path search is continuing at the last expanded node. In addition, we extended the search algorithm with a cache. Moreover, PlaSMA is capable of linking process data of cooperating companies and partners, e.g., customer orders or service requests, directly into the simulation platform to induce plausible, pertinent, and precise results that permit conclusions and analyses of real logistic processes with low costs. Batch-runs, process visualization (see Fig.5), as well as automated measurements of individually defined performance indicators allow fast and valid process evaluations.

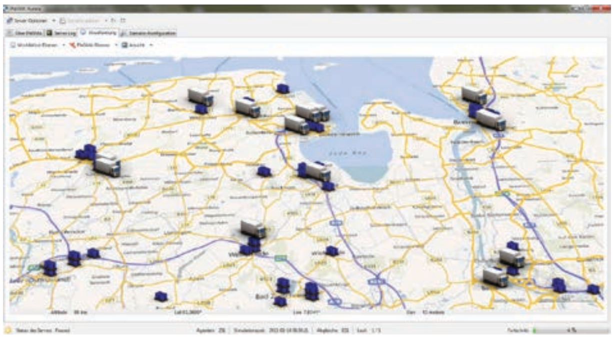

Figure 5. The PlaSMA simulation platform.

5 Evaluation

To verify the system's performance and show its applicability, we simulated real-world scenarios based on orders provided by our industrial partner as well as on transport infrastructures imported from OpenStreetMap databases. In Section 5.2, we evaluate the overall performance of implementing autonomous groupage traffic and focus on the decision making process by investigating at the impact of interrupting the solver if a fixed number of expansions is exceeded. To verify the systems performance quantitatively and show its applicability, Section 5.3 provides the result of a case study conducted in one of the offices of our industrial partner.

5.1 Experimental Setup

In our investigation we integrate the road network of Northern Germany. The whole modeled transport infrastructure contains $156,722$ nodes and $365,609$ edges. It includes all relevant highways, motorways, and inner city roads of the OpenStreetMap database. In order to prevent deadlocks caused by inaccurate data, nodes that cannot be reached from or to the depot of the transport service provider are removed.

Table 2. The amount and capacities of modeled vehicles within the experiments.

# Trucks	7.5 tons	12.5 tons	32 tons
80	30	20	30
60	10	20	30
40	0	10	30

The dispatched orders are provided by our industrial partner. We start a reverse geocoding process to map the address information to coordinates and determined the nearest neighbor node in the map, to link the addresses with graph nodes. The real weight, premium service constraints, latest delivery times, as well as the incoming dates of deliveries are attached to the order. Since exact incoming dates with timestamps of pickup orders are not available, only the date is considered during evaluation. Thus, we modeled the dynamics by setting the incoming date of every 10th pickup order to a random time of the day during operation. In real transport processes, vehicles with interchangeable units are sent to stops that have to be visited on a daily basis by fixed schedules. Consequently, we do not consider these orders. In each experiment $7,575$ orders are distributed within a whole week.

The reproducibility of results with the same input data is guaranteed by the simulation platform (see Section 4). All simulation runs are computed in a few hours on a laptop computer (equipped with an Intel Quad-Core i7 processor). Consequently, the system satisfies the runtime requirements for an application in real planning and control processes.

[3] For instance, for double precision floating data on a 64 bit machine it is bounded by $log(Doulbe_{M}axValue) = 1024$.

5.2 Evaluation of Autonomous Groupage Traffic

In the overall evaluation, we simulate a heterogeneous fleet of vehicles with varying capacities of 7.5 and 12.5 tons as well as trucks with swap bodies that have a maximum capacity of 32 tons. In addition, we assume shift-work between 5 am and 7 pm and set the maximum velocity of trucks to 120 km/h. Note that the maximum possible velocity is reduced by the corresponding speed limit of the road sectors. The handling and waiting periods at incoming goods departments is set to 10 minutes for each order. Table 2 depicts the vehicles and their capacities modeled in the experiments. Therefore, we investigate the strategy to raise the number of small-sized trucks to increase the transport volume.

Figure 6 depicts the amount of transported delivery and pickup orders as well as the number of service requests that cannot be satisfied.

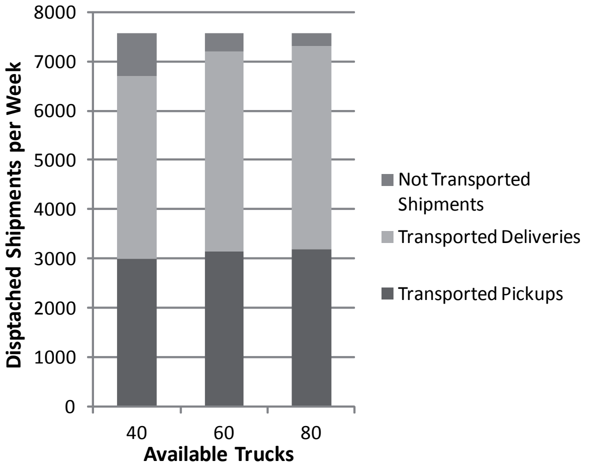

Figure 6. The delivered, picked up, and not transported shipments after the simulation of three scenarios of a whole week with a varying number of trucks.

It shows that the agent system is well suited to its application in groupage traffic. Dynamically incoming orders, the heterogeneous fleet, as well as individual properties of shipments are considered in the dispatching processes. Pickup and delivery orders are combined in valid tours without exceeding the maximum capacity of any truck.

It is obvious, that the number of transported shipments is increasing with the amount of available trucks. Nevertheless, the significant reduction of available trucks has only small effects on the efficiency of the whole system. This is caused by our strategy to remove small-sized trucks at first.

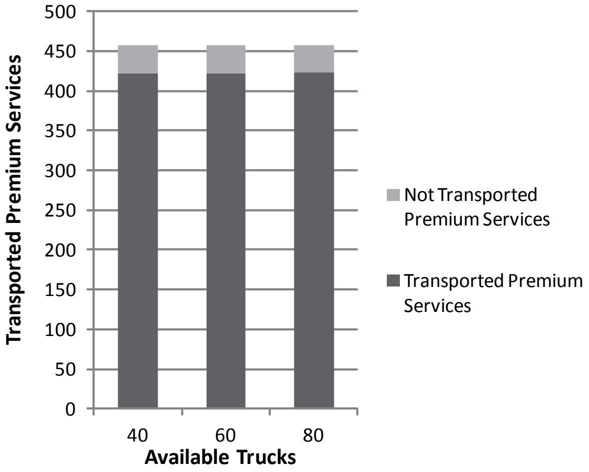

Figure 7. Transported and not transported premium services in the corresponding experiment.

Moreover, the results shown in Figure 7 pinpoint that the agent system considers all premium services with higher priorities than conventional orders. While the percentage of transported orders is reduced by nearly 10% if the number of available trucks is decreased, the amount of not transported premium services remains constant.

If all trucks have left the depot, new incoming delivery requests cannot be accommodated on this day (assuming that each truck is driving a single tour per day). Consequently, even if enough trucks are available about 259 service requests are not processed in the whole week.

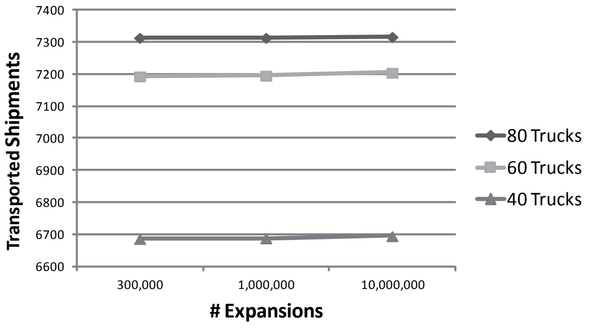

Figure 8. Expansions of each TSP within the agent's decision making process in correlation with transported shipments in scenarios with 40, 60, and 80 trucks.

By delegating the decision making process to decentralized entities the overall VRP is split into smaller TSPs that can be solved efficiently by each agent with Algorithm 3.2. Figure 8 shows the maximum number of expansions of Algorithm 3.2 and the impact to successfully transported shipments. The TSP solver already finds adequate solutions after expanding 300.000 nodes. It should be noted, that expanding more nodes in each TSP is negligible for the solution quality of the overall VRP. As a result, the applied pruning rules reduce the problem space significantly, if we evaluate the solver with real world orders.

5.3 Utility, Advantages and a Case Study

To verify the system performance in real world application, we constitute a consistent scenario based on the processes and orders of our cooperation partner. Subsequently, the results of simulated scenarios are compared to performance indicators which have been measured in current processes. In this section, we conclude with the benefit and the advantages observed in the case study.

In the case study, we simulated a whole month including all effectively transported orders (more than 1000 per day) and the real fleet operated within the simulated time window. In addition, exact properties of the orders are provided by the our industrial partner (see Section 5.1). The modeled fleet reflects real speed limits as well as capacities. Note that also in this investigation the maximum possible velocity is reduced by the corresponding speed limit of the road sectors. In addition, we simulate that each cargo handling operation of shipments (up to 300 kg) consumes 15 minutes. Process disturbances, e.g., if the delivery is not possible, are identified in the real process data and simulated respectively. Measured performance indicators were specified in cooperation with the controlling department of our industrial partner.

While in current processes routes are determined by each freight carrier manually, the system increases the efficiency by providing optimal and factual proposals at the start of a shift. It checks hard constraints automatically and accelerates the decision-making of freight carriers. As a result, each freight carrier saves about 20 minutes time each day, because route proposals are computed automatically and have to be checked only. Moreover, the continuous process monitoring improves the transparency. The current positions of each shipment are visualized and additional information about each load is provided, e.g., the effective weight, the estimated time of arrival, and its volume. This information can also be applied to further optimize and synchronize the supply chain. For instance, the estimated time of arrival can be sent automatically to the incoming goods departments of customers (e.g., via apps), who start preparing the receipt of goods or proactively send a message indicating that a delay is expected. Consequently, the freight carriers can react on the changing situation in advance and adapt tours and routes if necessary. In addition, the dispatching system increases the customer service level by reliable pickups and deliveries. At each step of the process the system checks time windows as well as the premium service constraints.

The results reveal an increasing efficiency and a significant reduction of cost by applying the agent-based dispatching system in groupage traffic. Therefore, the number of stops is reduced by an average of 29%. The agent system enables an efficient grouping of packages at a certain pickup or delivery location. Consequently, loads at the same location are transported by a single vehicle (if it is possible). Indirectly, this is also achieved by shifting conventional orders to following days. Thus, the probability of bundling shipments in increasing.

Moreover, the number of shipments which have to be transported by an external transport provider is reduced by an average of 82%. This is due to prioritizing the transport of premium services in the planning process and to shifting the delivery of conventional orders to next days if a premium service can be delivered instead. In addition, the agent system improves the reactions on the daily changing order situation and unexpected events. Also during operation the main goal is to maximize the amount of premium services.

Finally, we analized all tours and routes of an arbitrary day in cooperation with the dispatcher of our industrial partner in detail. Therefore, we prepared a visualization of all tours and routes in Google Earth including stops and additional information such as the time of arrival, the weight of the freight, the number pickup or delivery units at a stop, the maximum capacity of the truck, the working time of the freight carriers, the total delivery weight of a tour, and the total pickup weight of the tour. The discussion focused on several important aspects such as the length of routes, the working times of the freight carriers, the capacities of the trucks, guaranteed time windows of orders, vehicle restrictions with respect to the transport infrastructure, as well as special situations in which the knowledge of the dispatcher is essential. In conclusion, this analysis proves the road capability as well as the suitability for the daily use of automatically computed tours and routes as tours and routes were considered to be adequate and realizable. While the agent-based dispatching system considers more constraints and options within the planning processes to increase the potential of optimization, the human dispatcher is capable to further improve proposals selectively with his/her expert knowledge which is not modeled in the current system.

6 Conclusion and Outlook

To face the high complexity in groupage traffic and the dynamics of consecutively incoming orders, we provided a reactive and proactive multiagent system for the planning and control processes of a forwarding agency. Agents link the planning and scheduling processes directly to the actions of their represented vehicles and shipments. Therefore, both internal and external changes are considered during runtime and induce a reactive behavior. The focus is on the planning and decision making processes of the agents to develop an efficient Traveling Salesman Problem (TSP) solver which is crucial for negotiations between agents. The solver supports the combination of pickup and delivery tours without exceeding the maximum capacity of the vehicles and considers time windows, handling times, and request priorities. Applying bitvector operations allows for constraint checks in $O(1)$ time and space. It is shown that the *anytime* behavior of the TSP algorithm accelerates the search without significant impact to the solutions' quality.

To evaluate the dispatching system, we simulated several scenarios using the PlaSMA simulation platform with real orders provided by our industrial partner. The results reveal that applying the agent system is adequate in dynamic scenarios with daily varying amounts of orders, unknown requests, and heterogeneous properties. The system is designed to meet the special requirements in groupage traffic. It supports the combination of pickup and delivery tours and considers all relevant constraints. Moreover, the system maximizes the number of transported premium services as well as the processing amount of conventional orders. It computes shortest routes for each vehicle. The agent-based system supports dispatchers and contracted freight carriers during operations. It automatically controls the processes and adapts plans on the detection of delays or a changing order situation.

In a further application, forwarding agencies are able to integrate historical and anticipated orders into the simulation platform for the evaluation of different transport strategies. For instance, managers may investigate the effects of engaging more or less trucks.

In our further research, we will investigate the integration of a state-of-the-art Vehicle Routing Problem (VRP) solver to improve the rough planning processes. Afterwards, the *vehicle agents* may improve the route proposals and check problem specific requirements which are not covered by solver in general. Subsequently, the agent system will improve the overall performance with its flexible behavior in dynamic environments.

Profit sharing methods for freight carriers might be considered to promote further cooperation also between companies (e.g., [27]). As a result, these methods potentially increase the efficiency of forwarding agencies and reduce the amount of orders that must be transported by cost-intensive external operators.

Future investigations will include different optimization criteria such as the reduction of CO_2 emissions. For instance, longer tours can be CO_2-efficient given that a smaller load leads to less fuel consumption.

Acknowledgment

The presented research was partially funded by the German Research Foundation (DFG) within the Collaborative Research Center 637 Autonomous Cooperating Logistic Processes (SFB 637) and the project Autonomous Courier and Express Services

(HE 989/14-1) at the University of Bremen, Germany.

References

[1] M. Gendreau, "Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms," *Transportation Science*, vol. 39, no. 1, pp. 104–118, 2005.

[2] M. Gendreau and O. Brysy, "Vehicle Routing Problem with Time Windows, Part II: Metaheuristics," *Transportation Science*, vol. 39, pp. 119–139, 2005.

[3] O. Brysy, W. Dullaert, and M. Gendreau, "Evolutionary Algorithms for the Vehicle Routing Problem with Time Windows," *J. Heuristics*, vol. 10, pp. 587–611, 2004.

[4] J. Homberger and H. Gehring, "A Two-Phase Hybrid Metaheuristic for the Vehicle Routing Problem with Time Windows," *European Journal of Operational Research*, vol. 162, pp. 220–238, 2005.

[5] B. Golden, S. Raghavan, and E. Wasil, *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer Verlag, 2008.

[6] M. Dorigo and L. Gambardella, "Ant colonies for the traveling salesman problem," 1996.

[7] B. Scholz-Reiter, K. Windt, J. Kolditz, F. Böse, T. Hildebrandt, T. Philipp, and H. Höhns, "New Concepts of Modelling and Evaluating Autonomous Logistic Processes," in *IFAC Manufacturing, Modelling, Management and Control*, G. Chryssolouris and M. D, Eds. Athens, Greece: Elsevier, 2004.

[8] T. Warden, R. Porzel, J. D. Gehrke, O. Herzog, H. Langer, and R. Malaka, "Towards Ontology-based Multiagent Simulations: The PlaSMA Approach," in *European Conference on Modelling and Simulation (ECMS)*, A. Bargiela, S. Azam Ali, D. Crowley, and E. J. Kerckhoffs, Eds., 2010, pp. 50 – 56.

[9] M. Wooldridge, "Intelligent Agents," in *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*, G. Weiss, Ed. MIT Press, 1999, pp. 27 – 77.

[10] K. Dorer and M. Calisti, "An Adaptive Solution to Dynamic Transport Optimization," in *International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, ACM. ACM Press, 2005, pp. 45–51.

[11] J. Himoff, G. Rzevski, and P. Skobelev, "Multi-Agent Logistics i-Scheduler for Road Transportation," in *International Conference on Autonomous Agents and Multi Agent Systems (AAMAS)*, 2006, pp. 1514–1521.

[12] A. Schuldt, *Multiagent Coordination Enabling Autonomous Logistics*. Springer Verlag, 2011.

[13] J. O. Berndt and O. Herzog, "Distributed Learning of Equilibrium Strategies in Concurrent Iterated Many-issue Negotiations," in *Multiagent System Technologies (MATES)*. Berlin: Springer, 2012.

[14] K. Fischer, J. P. Müller, and M. Pischel, "Cooperative Transportation Scheduling: An Application Domain for DAI," *Journal of Applied Artificial Intelligence*, vol. 10, pp. 1–33, 1995.

[15] H.-J. Bürckert, K. Fischer, and G. Vierke, "Applied Artificial Intelligence : An Holonic Transport Scheduling with Teletruck," *Applied Artificial Intelligence*, vol. 14, no. 7, pp. 697 – 725, 2000.

[16] P. Skobelev, "Multi-agent systems for real time resource allocation, scheduling, optimization and controlling: Industrial applications," in *Holonic and Multi-Agent Systems for Manufacturing (HOLOMAS)*, 2011, pp. 1–14.

[17] M. Hülsmann, B. Scholz-Reiter, and K. Windt, Eds., *Autonomous Cooperation and Control in Logistics: Contributions and Limitations - Theoretical and Practical Perspectives*. Berlin: Springer-Verlag, 2011.

[18] J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *5th Berkeley Symp. Math. Statist. Prob.*, vol. 1, pp. 281 – 297, 1967.

[19] N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem," Graduate School of Industrial Administration, Carnegie-Mellon University, Tech. Rep. 388, 1976.

[20] R. Baldacci, A. Mingozzi, and R. Roberti, "Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints," *European Journal of Operational Research*, vol. 218, no. 1, pp. 1–6, 2012.

[21] Foundation for Intelligent Physical Agents, "FIPA Contract Net Interaction Protocol Specification," Geneva, Switzerland, Standard No. SC00029H, 2002.

[22] N. R. Jenninges, "An Agent-Based Approach for Building Complex Software Systems," *Communication of the ACM*, vol. 44, no. 4, pp. 35 – 41, 2001.

[23] F. Bellifemine, G. Caire, and D. Greenwood, *Developing Multi-Agent Systems with JADE*. Chichester, UK: John Wiley & Sons, 2007.

[24] J. D. Gehrke, A. Schuldt, and S. Werner, “Quality Criteria for Multiagent-Based Simulations with Conservative Synchronisation,” in *13th ASIM Dedicated Conference on Simulation in Production and Logistics*, M. Rabe, Ed., Citeseer. Stuttgart: Fraunhofer IRB Verlag, 2008, pp. 545–554.

[25] E. W. Dijkstra, “A note on two problems in connexion with graphs.” *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[26] R. K. Ahuja, K. Mehlhorn, J. Orlin, and R. Tarjan, “Faster Algorithms for the Shortest Path Problem,” *Journal of the ACM*, vol. 37, no. 2, pp. 213–223, Apr. 1990.

[27] M. a. Krajewska, H. Kopfer, G. Laporte, S. Ropke, and G. Zaccour, “Horizontal Cooperation among Freight Carriers: Request Allocation and Profit Sharing,” *Journal of the Operational Research Society*, vol. 59, no. 11, pp. 1483–1491, Sep. 2007.