Agent-Based Dispatching in Groupage Traffic

Max Gath, Stefan Edelkamp, and Otthein Herzog

Center for Computing and Communication Technologies (TZI),
Institute for Artificial Intelligence,
University of Bremen, Germany
Email: {mgath,edelkamp,herzog}@tzi.de

***Abstract*—The complexity and dynamics in groupage traffic requires flexible, efficient, and adaptive planning and controlling processes. While the general problem refers to the Vehicle Routing Problem (VRP), additional requirements have to be fulfilled in application. Individual properties and priorities of orders, a heterogeneous fleet of vehicles, dynamically incoming orders, unexpected events etc. require a proactive and reactive system behavior. To enable automated dispatching processes, we have implemented a multiagent system where the decision making is shifted from a central system to autonomous, interacting, intelligent agents. To evaluate the approach we used multiagent-based simulation and modeled several scenarios on real world infrastructures with orders provided by our industrial partner. The results reveal that agent-based dispatching meets the increasing requirements in groupage traffic while supporting the combination of pickup and delivery tours and accommodating request priorities, time-windows, as well as capacity constraints.**

I. Introduction

In recent years, the complexity and dynamics in logistic processes have been increased due to shorter product life cycles, the rising number of product variants, as well as the growing number of transnational links and dependencies within the production processes between companies. As a result, the requirements of transport processes are increasing 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 metaheuristics 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]–[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 controlling systems to decentralized, heterarchical systems [7]. Autonomously acting software agents represent logistic objects, e.g., shipments, trucks, and containers. They have the ability to interact with other agents by the use of negotiation and communication mechanisms. By delegating planning and control processes to decentralized entities, e.g., agents that represent vehicles, the overall problem is split into smaller problem instances that can be solved optimally.

We present an autonomous dispatching system that accommodates the requirements in groupage traffic. The paper is structured as follows. Section II gives insights to the logistic problem that was extracted in a foregoing process with a forwarding agency and specifies the problem formally. In Section III, we present the implemented multiagent system including the interaction protocols as well as the decision making processes of the agents. Thereby, we look at the effects of increasing the complexity of the problem by adding constraints that have to be considered in groupage traffic. In Section IV we present the multiagent simulation system PlaSMA [8].

For evaluation, 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 V. Finally, we conclude and provide future research directions.

II. Groupage Traffic

In groupage traffic, several orders to different destinations 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 build 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. Consequently, the general problem of a participating forwarding agency refers to the Vehicle Routing Problem (VRP) [9] with Simultaneous Delivery and Pickup (VRPSDP) [10].

In general, the VRP is concerned with determining tours with minimum costs for a fleet of vehicles to satisfy customer requests at different destinations while the start and end point of the tour is the depot.

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

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

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

$$\sum_{v\in V}\sum_{i\in S} x^v_{i,j} = 1 \text{ for all } j \in S \tag{3}$$

$$\sum_{v\in V}\sum_{j\in S} x^v_{i,j} = 1 \text{ for all } i \in S \tag{4}$$

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

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

Moreover, time window constraints as well as time consumption at the warehouse/customer have to be considered. 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 $time^v_{i,j}$ vehicle v's time for driving from i to j

$$x^v_{i,j} = 1 \Rightarrow l_i \geq r_i + t_i + time_{i,j} \tag{7}$$

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

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

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 to the truck's load.

The process planning complexity is even increased by individual qualities of shipments like weight, volume, priority, and value. Handling the complexity is aggregated 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 changing conditions and incoming orders, it is essential to adapt tours and timetables in respect to actual capacities.

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 fulfill 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 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 penalty if they are not fulfilling the agreed commitments.

Definition 2 (Premium Stop): Pickup or delivery stops being premium services are defined by the boolean function

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

On the other hand, *conventional* orders can be hold up to two more days. As a result, the objective function of the VRP includes not only to find a solution with minimum costs, but tours that maximizes 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:

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

In order to increase the service quality through short transit times and reliable deliveries it is mandatory to handle the high degree of dynamics and complexity of logistic processes with adaptive, reactive system behavior.

III. A Mulitagent System for 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 [11]. Therefore, agent systems are especially applied in 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 [12], [13]. Team formation and interaction protocols have been designed for efficient resource allocation [14] as well as for concurrent negotiations between logistic service providers and service consumers [15]. Agent-based systems have optimized planning and control processes within dynamic environments [16], [17]. Other ranges of application have been provided for industrial logistic processes [18]. A comprehensive survey in research on autonomous logistic processes is provided by [19] and [14].

A. Agent Interaction and Execution

In this investigation, we extended the agent system described in [20]. In our setting agents represent trucks and orders. Whenever a new request has to be acted upon, an agent is created that represents the given order. The goal of the agent 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 to the given time window constraints.

At first, the shipment agent send a cluster request to a cluster agent which collects all requests until the operational processes of the trucks start in the morning. The rough planning is done by applying a *K-Means* clustering [21]. The goal is to assign each received shipment to one of k available trucks. The clustering considers the coordinates of the pickup or delivery location of the shipment. Therefore, the algorithm consolidates shipments that have to be transported from or to nearby districts in the same cluster. In order to accelerate the clustering, other constraints are neglected, but considered in

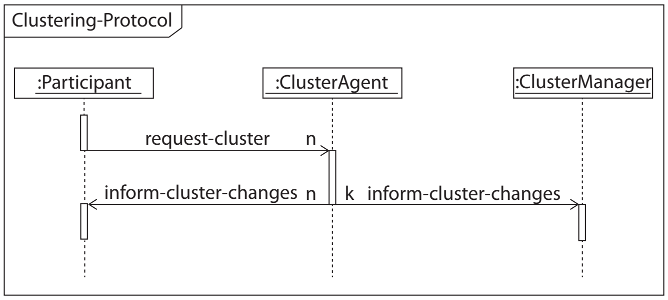


Fig. 1. The interaction protocol for rough planning.

the detailed planning processes for tour determination of each truck (see Section III-B).

The implemented interaction protocol is shown in Figure 1. 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. The truck agents start a detailed planning process (see Section III-B) with respect to their individual properties, capacities, time window, and premium service constraints. Agents representing dynamically incoming orders during operation start directly a contract-net protocol [22] negotiation with the truck agents without sending cluster requests. Thus, each shipment sends a *call for proposal* message to available trucks. Trucks compute proposals by determining their additional costs for serving the order. In transport logistics the costs of an order are commonly based on the additional distance that has to be driven by the vehicle. To compute the distance, the agent has to start a decision making process described in Section III-B. In order to schedule new orders also while transporting other shipments, the truck has to consider its current capacity constraints and position. For example, picked up shipments reduce the capacities and the position of the vehicle affects the determination of shortest ways and tours. Consequently, we link the planning and decision making processes of the agents directly with their execution behaviors and consider all relevant observed changes of the environment as well as the internal state of the agent within the decision making and tour planning.

The computed costs are sent back to the shipment agent which chooses the transport provider with least costs. If it is not possible to satisfy the shipments' requirements, a *refuse* message is send by the truck agent. To transport a premium service instead of conventional orders or another premium service by driving a shorter distance, already accepted orders may not be included in the new plan. If these orders have not been boarded the truck agent sends also *refuse* messages to the agents that act on behalf of the corresponding orders. Afterwards, the order agent negotiates 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 weaken 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.

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 he 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 manipulates the running handling processes.

B. Agent Decision Making

Within the negotiation, truck agents have to compute proposals and decide which service request has to be satisfied. These costs are based on the additional distance that has to be driven by the vehicle. To compute the distance, the agents must solve a generalization of the *NP-Hard* Traveling Salesman Problem (TSP) [23] that is a single vehicle variant of the VRP defined in Section II.

To implement an efficient decision making process and consider pickup and deliveries simultaneously, we adapted the optimal *depth-first branch-and-bound* TSP solver described in [20] that solves TSPs containing only pickup orders. Moreover, we ensure 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.

The pseudo-code implementation is shown in Algorithm 1. 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 costs 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 costs 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 and the shortest path is returned.

```
**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**$(n, u, g, U)$
  $d \leftarrow depth(u)$
  $tour_d \leftarrow u$
  **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**
    $\{v_0, \ldots, v_n\} \leftarrow nextcities(u)$
    **for each** $j$ **in** $\{1, \ldots, n\}$
      $exp{+}{+}$
      **if** $(Constraint(v_j) \wedge maxExp \geq exp)$
        **if** $(g + h(v_j) < U)$
          **call** $DFS(n, v_j, g + c_{u,v_j}, U)$
```

Algorithm 1: DFBnB Algorithm for the Constraint TSP.

At each node the visited cities, the current time, premium service information, and the current capacity of the tour are saved as bit-vector. All bit-vector operations (setting, clearing of bits, check for subsumption) run in $O(1)$. This enables constraint checks in $O(1)$ because each check is done by bit-vector comparison, e.g., 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. This holds not for the mixture of both problems. 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_T that the truck has reached on the tour, the current capacity χ_C of the truck, and let ω denote the weight of the order at stop s, on each node χ_T is updated with

$$\chi_T = \begin{cases} \max(\chi_C, \chi_T + \omega), & \text{if } s \text{ is a delivery stop} \\ \max(\chi_C, \chi_T), & \text{otherwise.} \end{cases} \qquad (12)$$

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

$$\tau \geq \chi_T. \qquad (13)$$

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 I 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_T 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_T is increasing. Consequently, it is not possible to add a delivery stop in depth 5 although other pickup stops are included afterwards.

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. ■

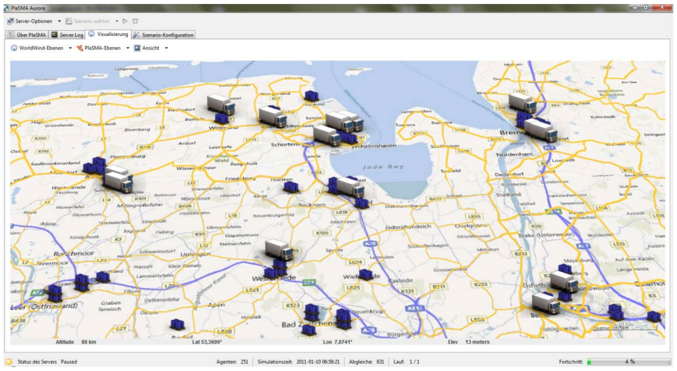
IV. Agent-Based Simulation

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.

PlaSMA [8] (see: http://plasma.informatik.uni-bremen.de/) is an agent-based event driven simulation platform that has

TABLE I
An example for checking the capacity constraints of a truck in the tree structure.

depth	χ_C	χ_T	is pickup stop	plan is valid (Equation 13)
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



Altitude 88 km   Lat 53,3699°   Lon 7,8741°   Elev 13 meters

Status des Servers Paused   Agenten: 251 | Simulationszeit: 2011-01-10 06:58:21 | Abgleiche: 831 | Lauf: 1 / 1   Fortschritt: 4 %

Fig. 2. The PlaSMA simulation platform.

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) [24] for agent communication and coordination and provides discrete time simulation that ensures correct synchronization while satisfying time model adequacy, causality, and reproducibility.

In order to simulate industrial and transport processes reliably, it enables the simulation of real-world infrastructures and supports their import from OpenStreetMap. 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.

Particularly, on real infrastructures shortest-path searches are cost-intensive operations. Therefore, we developed a efficient implementation of Dijkstra's shortest-path algorithm [25] based on radix-heaps [26] for computing a distance matrix as input for TSP solver.

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, as well as automated measurements of individually defined performance indicators allow fast and significant process evaluations.

V. Evaluation

To verify system 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. Next, we focus on the performance of the decision making process. While the performance of the optimal solver is provided by [20], we look at the impact of interrupting the solver if a fixed number of expansions is exceeded.

A. Experimental Setup

In our investigation we integrated 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.

The dispatched orders are provided by our industrial partner. We started 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 with 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 did not consider these orders. In each experiment $7,575$ orders are distributed within a whole week.

We simulated a heterogeneous fleet of vehicles with varying capacities of 7.5 tons, 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 19 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 II depicts the vehicles and its capacities modeled within the experiments. Therefore, we investigate the strategy to raise the number of small-sized trucks, to increase the transport volume.

The reproducibility of results with the same input data is guaranteed by the simulation platform (see Section IV).

B. Results

Figure 3 depicts the amount of transported delivery and pickup orders as well as the number of service requests that cannot be satisfied.

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

TABLE II
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

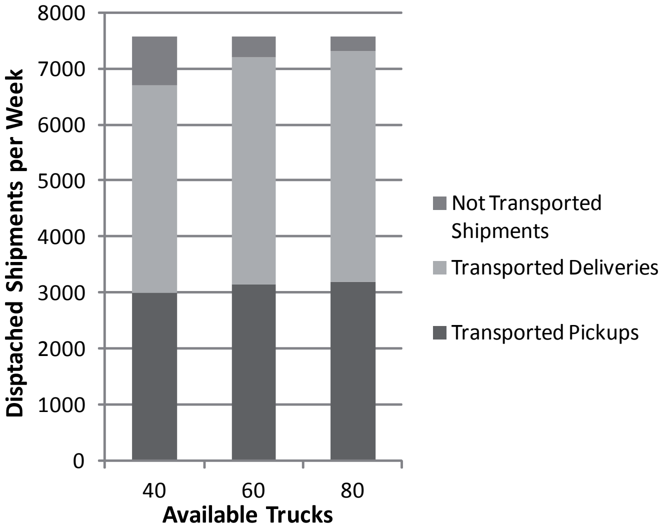


Fig. 3. The delivered, picked up, and not transported shipments after the simulation of three scenarios of a whole week with a varying number of trucks.

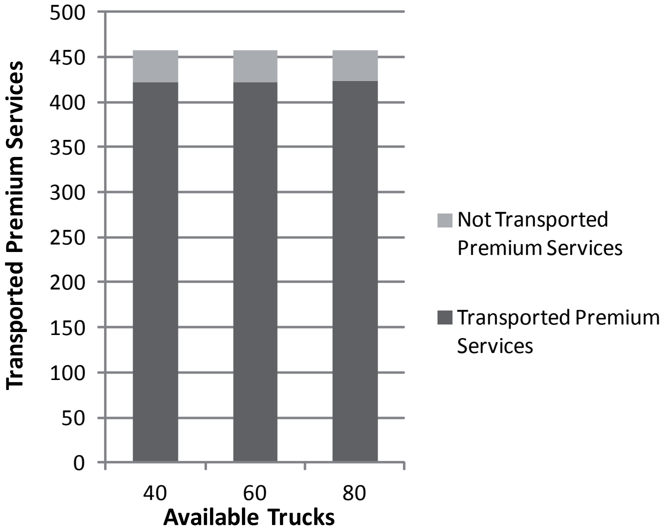


Fig. 4. Transported and not transported premium services in the corresponding experiment.

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.

Moreover, the results shown in Figure 4 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.

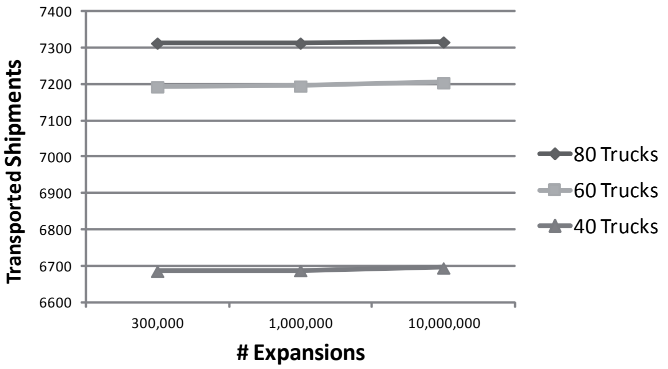


Fig. 5. Expansions of each TSP within the agent's decision making process in correlation with transported shipments in scenarios with 40, 60, and 80 trucks.

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 1. Figure 5 shows the maximum number of expansions of Algorithm 1 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.

All simulation runs are computed within a few hours on a laptop computer (equipped with an Intel Quad-Core i7 processor). Consequently, it satisfies the runtime requirements for application in real planning and control processes.

VI. Conclusion and Outlook

To face the high complexity in groupage traffic as well as 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 with the actions of represented vehicles and shipments. Therefore, internal changes as well as changes of the environment can be considered during runtime and induce a reactive behavior. We focused on the planning and decision making processes of the agents and developed an efficient TSP solver that is crucial for negotiation with service customer agents. The TSP solver is designed to meet the special requirements in groupage traffic. It supports the combination of pickup and delivery tours without exceeding the maximum capacity of the vehicles and considers time windows, handling times as well as request priorities. Applying bitvector operations enables constraint checks in $O(1)$ time and space. Moreover, the solver maximizes the number of transported premium services as well as the processing amount of conventional orders. To evaluate the dispatching system, we modeled several scenarios within the PlaSMA simulation platform with real world 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. Moreover, it showed that the anytime behavior of the TSP algorithm accelerates the search without significant impact to the solutions' quality. Both carriers and dispatchers profit directly from tour and routing proposals.

In addition, the integration of historical and anticipated orders into the simulation platform can be used by forwarding agencies for the evaluation of different transport strategies, e.g., the effects of engaging more trucks can be investigated.

Further research will focus on the integration of traffic simulation within our simulation framework as well as on modeling and simulation of unexpected events like delays on incoming goods departments. This enables evaluations in environments with even higher dynamics.

To increase the efficiency of forwarding agencies and reduce the amount of orders that have to be transported by cost intensive external operators, also profit sharing methods for freight carriers may be considered (e.g., [27]).

Future investigations will include as well considering different optimization criteria like the reduction of CO_2 emissions. Longer tours can be CO_2-efficient given that a smaller load requires less fuel consumption.

Acknowledgment

The presented research was partially funded by the German Research Foundation (DFG) within the Collaborative Research Centre 637 Autonomous Cooperating Logistic Processes: A Paradigm Shift and its Limitations (SFB 637) 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] 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.

[10] H. Min, "The multiple vehicle routing problem with simultaneous delivery and pick-up points," *Transportation Research Part A: General*, vol. 23, no. 5, pp. 377–386, 1989.

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

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] S. Edelkamp and M. Gath, "Optimal Decision Making in Agent-Based Autonomous Groupage Traffic," in *Proceedings of the 5th International Conference on Agents and Artificial Intelligence (ICAART)*. Barcelona: SciTePress, 2013 (to appear).

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

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

[23] 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.

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

[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.