AUTONOMOUS LOGISTIC PROCESSES OF BIKE COURIER SERVICES USING MULTIAGENT-BASED SIMULATION

Max Gath[(a)], Thomas Wagner[(b)], Otthein Herzog[(c)]

Center for Computing and Communication Technologies (TZI)
University of Bremen, Am Fallturm 1, 28359 Bremen

[(a)]mgath@tzi.de, [(b)]twagner@tzi.de [(c)]herzog@tzi.de

ABSTRACT

We present an agent-based approach for solving *pickup and delivery problems* (PDP) in dynamic environments and show its application by bike couriers in urban districts. To avoid social conflicts, the focus is on the computation of fair allocations of orders to bike couriers who are paid by commission fees. We realize autonomous logistic processes and present a mulitagent system with specially adapted negotiation mechanisms. In addition, we apply the mulitagent-based simulation platform PlaSMA for modeling, simulating, and evaluating different transport scenarios within the transport infrastructure of the city of Bremen, Germany. The customer orders are based on real data provided by a German bike courier company. The results show that the approach indeed improves the fairness of distributions significantly. Finally, we present a new business model for a commercial online dispatching platform enabled by our approach.

Keywords: Multiagent-based Simulation, Scheduling, Pickup and Delivery Problem, Courier and Express Services

1. INTRODUCTION

Transporting goods in urban districts by bike couriers provides many advantages. It contributes to a reduction of traffic and CO_2 emissions, increases the service quality through short transit times, and satisfies individual demands of customers. However, determining optimal routes to satisfy transport requests in real-world operations involves a set of practical complications. For instance, to avoid social conflicts and increase the service quality of bike couriers there is an essential demand for a “fair” distribution of orders to couriers who are paid by commission fees. In fact, in high-frequency districts many customers can be served by a commonly favored tour. Nevertheless, unpopular tours have to be done as well.

This paper addresses the so-called *Pickup and Delivery Problem* (PDP) with special requirements that are introduced in Section 2. In Section 3, we provide a short introduction to autonomous logistic processes and refer to related work. In Section 4, we present a novel agent-based approach for solving PDPs in dynamic real-world environments and show its application to bike couriers in urban districts. The participating objects and actors in our logistic scenarios are represented by intelligent software agents that act autonomously. Through specially adapted interaction protocols, agents exchange tasks with each other and improve solutions continuously. We show that the mechanism provides “fair” distributions without reducing the logistic service quality.

For the purpose of evaluation, we use the multiagent-based simulation platform PlaSMA (Warden et al. 2010) for modeling and simulating different transport scenarios within the transport infrastructure of Bremen, Germany. PlaSMA is described in Section 5. In Section 6, we specify the experimental setup and compare the formally specified fairness of the routes to the results of a robust distance minimizing algorithm. The customer orders are based on real data provided by a German bike courier company. The results show that the approach improves the fairness of distributions significantly. Finally, we argue that the developed approach enables innovative business models for commercial online dispatching platforms that can be used by bike couriers and retailers to offer reliable deliveries as well as sustainable transportation.

2. THE DYNAMIC PICKUP AND DELIVERY PROBLEM OF BIKE COURIERS

The well-known *pickup and delivery problem* (PDP) (Berbeglia, Cordeau, and Laporte 2010; Parragh, Doerner, and Hartl 2008) is concerned with determining a set of minimum cost routes for a fleet of vehicles to satisfy customer requests for transporting objects from an origin to a destination. Certain scenarios involve additional constraints and customer demands that have to be considered, e.g., time windows (Dumas, Desrosiers, and Soumis 1991).

The investigated problem is a variant of the PDP. In logistic processes of courier, express and parcel services (CEP) most requests are received on their service day and have to be operated upon within minutes. The dynamics and the complexity increase significantly with the consecutive appearance of new requests as well as the incidence of unexpected events like changing traffic conditions.

Not only the needs of the customer and of the logistic company have to be considered, but also the demands and varying capabilities of the bike couriers themselves, who are usually paid by commission fees and have an individual physical fitness. Additionally, the pickup and delivery stops are located in various districts and thereby valued differently. In inner city sectors more stops can be accomplished by short tours, whereas in outskirts only one delivery can be served by driving a long distance. Nevertheless, all incoming requests have to be serviced. Thus, there is the essential demand for the generation of “fair” distributions without privileging or overstraining any courier with respect to his fitness.

3. AUTONOMOUS LOGISTIC 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., by Bräysy and Gendreau (2005a, 2005b) and Parragh et al. (2008). However, central planning and controlling in dynamic and complex logistic processes is increasingly difficult due to the requirements of flexibility and adaptability of changing environmental influences (Scholz-Reiter et al. 2004). Often one faces scenarios where it is not possible to acquire all relevant information for the decision making process centrally.

In autonomous logistic processes the decision making is shifted from central, hierarchical planning and controlling systems to decentralized, heterarchical systems (Scholz-Reiter et al. 2004). In agent-based processes autonomously acting software agents represent logistic objects, such as shipments, trucks, and containers (Fischer, Müller, and Pischel 1995; Schuldt 2011). They have the ability to interact with other agents by means of negotiation and communication mechanisms. The general problem is split into smaller problems that agents solve locally concurrent within short time windows to optimize the behavior of the overall system. In cooperating systems the agent’s goal is to pursue a globally optimized behavior and achieve common goals whereas in competitive systems each agent acts selfish to reach its own objectives.

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 (Brooks 1986; Rao and Georgeff 1995). The potential of autonomous logistic processes is even bigger in open, unpredictable, dynamic and complex environments (Böse and Windt 2007; Kuske, Luderer, and Tönnies 2010; Schuldt 2011; Wessels, Jedermann, and Lang 2010; Windt and Hülsmann 2007). A comprehensive survey and the state of the art in research on autonomous logistic processes are provided by Hülsmann, Scholz-Reiter, and Windt (2011), by Hülsmann and Windt (2007) and by Schuldt (2011).

Examples of multiagent systems for resource allocation, scheduling, optimization, and controlling within industrial applications are provided by Himoff, Rzevski, and Skobelev (2006), by Neagu, Dorer, Greenwood, and Calisti (2006) and by Skobelev (2011).

4. EXTENDING INTERACTION PROTOCOLS FOR EVENHANDED DISPATCHING

Similar to other autonomous logistic processes (Schuldt 2011), autonomously acting, intelligent agents represent bike couriers and orders. Orders must be classified as popular or unpopular. However, the subjective value of an order depends on the courier (e.g., on the position, the currently accepted requests, and the individual fitness). Nevertheless, the probability to combine orders successfully in one tour is increasing in high-frequency areas. Therefore, the trading area is clustered in high-frequency and low-frequency areas using historical data. We apply a density based clustering (Ester, Kriegel, Sander, and Xu 1996) to determine interesting clusters of different sizes, diameters and shapes. The pickup and delivery locations of a shipment define whether it belongs to commonly popular orders or to unpopular orders.

Whenever a new request has to be acted upon, an agent is created that represents the given shipment. The goal of the agent is to find a proper transport service provider for carrying the shipment from its origin to the destination with respect to the given time windows.

The agent starts negotiating with other agents representing bike couriers. These agents are utility based agents that act selfish to optimize their individual utility function. Beside the monetary costs the agent considers its fitness value for computing the utility value of orders.

Definition 1 (The Orders Revenue). *Let* A *denote the set of agents,* Ω *the set of orders,* $\omega_{\text{Value}} \in \mathbb{R}^+$ *the monetary value of order* $\omega \in \Omega$ *and* $\alpha_{cost}: 2^{\Omega} \rightarrow \mathbb{R}^+$ *a mapping that determines the costs of agent* $\alpha \in A$ *of picking up and delivering orders* $S \in 2^{\Omega}$ *by solving a TSPTW. The mapping* $r: A \times 2^{\Omega} \rightarrow \mathbb{R}$ *computes the expected monetary revenue of orders S for Agent* α *by*

$$\mathrm{r}(\alpha, \mathrm{S}) = \sum_{\omega \in S}(\omega_{\text{Value}}) - \alpha_{\text{cost}}(\mathrm{S}) \,. \tag{1}$$

Definition 2 (The Fitness of a Bike Courier). *Let* $\alpha_{fitness} \in \mathbb{R}^+$ *denote the constant physical fitness value of the courier represented by agent* α *and* $dist: 2^{\Omega} \rightarrow \mathbb{R}^+$ *a mapping determineing the total distance that a courier has to drive for shipping orders S. The mapping* $f: A \times 2^{\Omega} \rightarrow \mathbb{R}^+$ *yields the fitness of the courier after delivering orders S. It is defined as*

$$f\big(\alpha, dist(\mathrm{S})\big) = 1 + \frac{a_{fitness}}{dist + a_{fitness}} \tag{2}$$

Definition 3 (The Bike Courier's Utility of Orders). *The mapping u:* $\mathrm{A} \times 2^{\Omega} \rightarrow \mathbb{R}$ *determines the utility of orders S for agent* α*. The mapping u is defined as*

$$\mathrm{u}(\alpha, \mathrm{S}) = \mathrm{r}(\alpha, \mathrm{S}) \cdot \mathrm{f}(\alpha, \mathrm{S}) \tag{3}$$

where $r\colon \mathrm{A} \times 2^{\Omega} \rightarrow \mathbb{R}$ *and* $f\colon \mathrm{A} \times 2^{\Omega} \rightarrow \mathbb{R}^{+}$ *are defined in Definition 1 and Definition 2, respectively.*

In transport logistics the costs of an order are commonly based on the additional distance that has to be driven by the courier for picking up and delivering the auctioned shipment. To compute the distance, the agent has to solve the Traveling Salesman Problem with Time Windows (TSPTW) (Dumas, Desrosiers, Gelinas, and Solomon 1995). We adapted the well established Solomon I1 insertion heuristic (Solomon 1987) to solve the TSPTW while supplementary considering the essential sequence constraints. The algorithm is not optimal, but is often applied because it obtains quite good solutions in a short time. Figure 1 shows the main procedure.

Input: The pickup stop, the delivery stop,
the current path
Output: A path containing the pickup and the
delivery stop, if no path exists it returns null

```
Procedure InsertionHeuristic(pickup,delivery,path)
begin
    path ← Insert(pickup, path)
    path ← Insert(delivery, path)
    if (isEmpty(path)) then
        path ← Insert(delivery, path)
        path ← Insert(pickup, path)
    end if
    return path
end
```

Figure 1: The Procedure of the Insertion Heuristic.

Figure 2 shows the insertion procedure. The new stop is inserted between all other stops in the current path successively. The subroutine *append*(x,y,z) appends stop x in path y before stop z and returns a new list. Within the subroutine *checkTimeAndSeqConstraints*() we check for all orders, if the delivery stop is behind the pickup stop as well as all time window constraints. The *cost*() function returns the distance of the path by summing up the weights of all edges.

Input: The stop to insert, the current path
Output: A path containing the new stop,
if no path exists it returns an empty path

```
Procedure Insert(stop, path)
begin
    newPath ← empty
    for all (stop) s ∈ path do
        tempPath ← append(stop,path,s)
        if(checkTimeAndSeqConstraints(tempPath) ∧
            (isEmpty(newPath) ∨
            cost(newPath) > cost (tempPath))) then
            newPath ← tempPath
        end if
    end for
    return newPath
end
```

Figure 2: The Insert Procedure.

Figure 3 shows the improvement procedure. Each stop of the current path is removed with the subroutine *remove*() and reinserted using the insert procedure. 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 current path is returned. In the worst case, the improvement algorithm creates all permutations of the stops (O(n!)). In order to reduce the worst case complexity, we integrated the *abort*() subroutine to stop the procedure after a defined number of cycles and to return the best valid solution.

Input: The current path
Output: An optimized path

```
Procedure ImprovementHeuristic(path)
begin
    tempPath ← empty
    while cost(path) > cost(tempPath) || abort() do
        if (isEmpty(tempPath)) then
            tempPath ← path
        end if
        path ← tempPath
        for all (stop) s ∈ tempPath do
            remove(s,tempPath)
            insert(s, tempPath)
        end for
    end while
    return path
end
```

Figure 3: The Improvement Procedure.

To ensure that autonomous selfish agents generate fair distributions, we developed a stable interaction protocol.

Definition 4 (The Fairness of a distribution). *The surjective mapping* $d\colon \Omega \rightarrow A \cup \{\lambda\}$ *maps an order* ω ∈ *Ω to an agent α* ∈ *A or to the symbol λ, if* ω *is not allocated to any agent. It creates a partition of all elements* ω ∈ *Ω. Let dom(d) denote the domain of d. The inverse mapping*

$$d^{-1}\,(d, \alpha) = \{\omega \mid \omega \in dom(d) \wedge d(\omega) = \alpha\} \tag{4}$$

defines all orders $\omega \in dom(d)$ *of an agent* α*. The standard deviation σ of each participant's individual utility*

$$\sigma_{\alpha \in A} u(d^{-1}\,(d, \alpha), \alpha) \tag{5}$$

determines the fairness of a distribution. The fairest distribution minimizes the standard deviation σ with

$$\min_d \ \sigma_{\alpha\in A} u(d^{-1}(d,\alpha),\alpha)\,. \qquad (6)$$

For sure, it is not a valid solution if all orders ω are allocated to λ, although Equation 5 and Equation 6 are zero.

We extended the Vickrey auction (Vickrey 1961) by introducing a currency for bidding on shipments. Every agent α gets a fixed start-up budget β_α^0 that he may use for bidding at time 0. The function $u^t(\alpha,\omega)$ defines the utility value of order ω by agent α at time t similarly to Definition 3 with respect to already accepted orders. The bid is computed by the following equation:

$$\mathrm{b}(\alpha,\omega,\mathrm{t}) = \begin{cases} \mathrm{u}^{\mathrm{t}}(\alpha,\omega), & \text{if } \mathrm{u}^{\mathrm{t}}(\alpha,\omega) \le \beta_\alpha^t \\ \beta_\alpha^t, & \text{else} \end{cases}. \qquad (7)$$

Consequently, the agent can at most bid the maximum amount of its current balance.

The winning agent α_{winner} is determined by the shipment agent with

$$\alpha_{\text{winner}} = \arg\ \max_{\alpha\in A} \beta(\alpha,\omega,\mathrm{t}). \qquad (8)$$

While using the Vickrey auction the winning agent has to pay the second highest bid:

$$\mathrm{p} = \ \max_{\alpha\in A\backslash\{\alpha_{\text{winner}}\}} \beta(\alpha,\omega,\mathrm{t}). \qquad (9)$$

Bidding the true valuation is the dominant strategy in private value Vickrey auctions for every participant (Shoham and Leyton-Brown 2009, S.319). Thus, the negotiation process is stable and no agent has any incentive to reveal false valuations and manipulate the negotiation process. Consequently, it is not possible for an agent to manipulate the outcome of a negotiation for its own purposes. Additionally, it must be guaranteed that agents can also increase their individual credit balance. To subsidize the transport of unpopular service requests, they get a higher amount of money for handling unpopular orders than for popular ones:

$$\mathrm{reward}(\omega) = \begin{cases} \mathrm{c}_1, & \text{if } \omega \text{ is popular} \\ \mathrm{c}_2, & \text{else} \end{cases} \text{ with } \mathrm{c}_1 > \mathrm{c}_2 \qquad (10)$$

The reward motivates the couriers to accept also unpopular orders to increase their credit balance. In addition, it prevents that one agent purchase only popular orders by auction if the budget reduces to zero.

Theorem 1 (Upper bounds of c_1 and c_2). The highest utility value of an order is the upper bound for c_1 and c_2.

Proof. "A solution x is Pareto efficient – i.e., Pareto optimal – if there is no other solution x' such that at least one agent is better off in x' than in x and no agent is worse off in x' than in x." (Sandholm 1999, S.202). An advantage of using the Vickrey auction is that every order is auctioned "Pareto efficient to the bidder that values it the most" (Sandholm 1999, S.213). If c_1 and c_2 are higher than the maximum utility value of an order, Equation 7 will not intervene the result. The mechanism is transformed into the standard Vickrey auction. Thus, the agents positioned in high-frequency areas will get the most popular orders because their additional costs are lower than the cost of other agents. Consequently, a Pareto efficient and cost minimized solution is constituted, which maximizes the social welfare (the total sum of all agent's utility). □

Theorem 2 (Lower bounds of c_1 and c_2). The lowest utility value of an order is the lower bound for c_1 and c_2. Otherwise, the negotiation maximizes neither the social welfare (the total sum of all agent's utility) nor it will lead to fair allocations.

Proof. The order is not sold to the bidder, who values it at most, if Equation 7 reduces the bid because of an insufficient budget. Consequently, the social welfare is not optimized. If c_1 and c_2 are always lower than the minimum computed utility value, the budget of all agents reduces monotonously and converges to zero. As a result, the allocation will not lead to fair allocations, but the orders are sold successively without considering the potential for allocating efficient solutions. □

It addition, the initial budget β_α^0 has to be sufficient low that it impacts the allocation. To support waiting couriers who are working but not processing any order, they are rewarded by receiving a small amount w for each waiting unit.

However, the theorems hold only for the auctioned item in the actual situation. If another shipment is auctioned afterwards, the situation can change significantly and earlier profitably valued orders can be rendered worse. In addition, the complexity is aggregated by the high degree of dynamics that result from unexpected events like changing traffic conditions and delays. To adapt tours and timetables in respect to the current situation, we apply the Vickrey auction iteratively to exchange tasks between the agents. In this case the auctioneer is also a service provider and can optionally refuse a deal. To avoid this problem, the auctioneer only accepts a transaction if he receives the profit that he would gain by serving the request by itself from the winner of the auction.

5. EVALUATION USING MULITAGENT-BASED SIMULATION

Multiagent-based simulation (MABS) combines concepts of mulitagent systems and qualitative simulation. Applying MABS for evaluating multiagent systems before their deployment in real applications is an accurate cost and time reducing method (Schuldt, Gehrke, and Werner 2008). This holds especially for scenarios with run-time agent interactions that cannot be predicted in advance (Jennings 2001).

PlaSMA (Warden et al. 2010) is a simulation middleware that extends the well-known JADE framework (Bellifemine, Caire, and Greenwood 2007) which implements the standards for agent interaction and communication defined by the *IEEE Foundation for Intelligent Physical Agents* (FIPA). PlaSMA provides a discrete time simulation that ensures a correct conservative synchronization with time model adequacy, causality and reproducibility (Schuldt et al. 2008). The GUI provides information about time progression and the current simulated processes, e.g., how many active agents are registered at the platform as well as the positions of the physical objects.

The physical world 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 traffic density on an edge. If the edge is denoted with a high traffic density, the maximum possible velocity is reduced respectively. To simulate dynamics of the environment and the appearance of unexpected events (e.g., accidents and changing traffic conditions), the values can vary during simulation runs.

In order to model sound panning and controlling processes in the logistic domain, we extended PlaSMA to import transport infrastructures from OpenStreetMap (see: www.openstreetmap.org) (OSM) databases. This enables the integration of high 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 world speed limits, the distance as well as the type of an edge.

Particularly within large infrastructures determining the shortest path between nodes is an essential, costly, and time-consuming procedure within the decision making process of an agent (see Section 4). Consequently, we implemented Dijkstra’s single-source shortest paths search (Dijkstra 1959) that is realized by a memory-efficient joint representation of graph and radix heap nodes (Ahuja, Mehlhorn, Orlin, and Tarjan 1990). Therefore, we can guarantee that the search is optimal and has linear time complexity.

To induce meaningful results, the simulation platform supports batch runs for the evaluation of a set of scenarios as well as multiple runs of a certain scenario with varying random seeds for the reproducible generation of random variables. In each run individual performance indicators are measured and saved in a database. Thus, they can be verified with an arbitrary spreadsheet program.

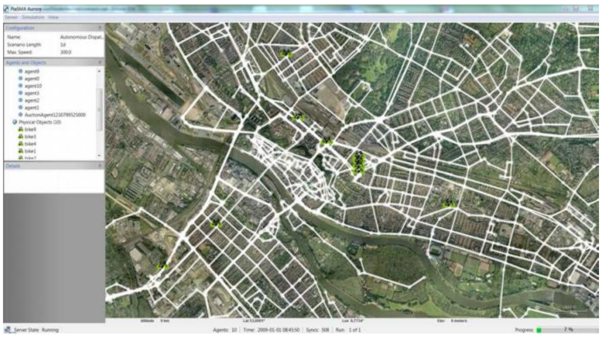
Further detailed information about the PlaSMA Simulation framework is provided, e.g., by Gehrke and Ober-Blöbaum (2007) and by Warden et al. (2010) as well as on the corresponding website http://plasma.informatik.uni-bremen.de.

6. RESULTS

In several preliminary scenarios we determined plausible values for an appropriate start-up budget and the amount of money an agent receives for transporting popular or unpopular orders. Thereby, we respected the upper and lower bounds deduced from Theorem 1 and Theorem 2. Afterwards, the evenhanded mechanism was evaluated by the comparison with a distance minimizing algorithm.

6.1. Experimental Setup

In this investigation we modeled the road network of the City of Bremen, Germany (see Figure 4). The whole modeled transport infrastructure of Bremen contains 2,103 Nodes and 4,448 Edges.



Name: Autonomous Dispat...
Scenario Length: 5d
Max. Speed: 300.0

Figure 4: The Picture Shows a Part of the Transport Infrastructure.

The costs of handling a specific task are equal to the additional distance in kilometers, which must be driven by the courier. Accordingly, we assume that the revenue for delivering an order is twice the distance in kilometers between its origin and destination. This assumption reflects real pricing systems. The maximum possible velocity of the couriers is set to 19 kilometers per hour and the fitness value $a_{fitness}$ to 30 for all agents that represent bike couriers. With respect to Definition 2, it implicates the realistic assumption that the fitness of each courier is half as much after driving 30 kilometers.

To investigate the fairness of the mechanism, we compared the computed allocations with solutions that were computed by a distance minimizing algorithm. Therefore, we implemented a distance minimizing algorithm that is based on our approach. However, this requires totally cooperating agents that accept every order even if it is less profitable for themselves. A currency is not needed. As a result, the orders are sold by the use of the standard Vickrey auction to the bike courier, who minimizes the total costs, which are equal to the driven distances. Thus, the first shipments will probably be served by one courier, because the additional costs are likely lower than the costs of the other couriers that are waiting at the depot. The negotiation protocol for changing orders during operations is adapted accordingly. In order to implement reactive behavior in both mechanisms, each agent starts a negotiation with another arbitrary bike agent consecutively in every minute.

The customer orders in the simulated scenarios are based on real customer orders of a representative week provided by a German bike courier company. The data include the origin, the destination, the time windows, and the dispatching time. We simulated a whole week with more than 1000 orders that have to be dispatched. The number of orders varied between 155 and 273 each day.

6.2. Parameter Optimization

We started simulating several preliminary scenarios with varying values for c_1, c_2, and the start-up budget β_a^0. We observed that the parameters c_1 and c_2 adjust the influence of the mechanism. If higher values are used, the influence of the mechanism is lower. As a result, the total distance of all couriers is decreasing, while it does not lead to fair allocations. Choosing low values we observed the same effect which is described in Theorem 2. The tours are neither fair nor do they have a short distance. The same effect could be observed by varying the start budget β_a^0, respectively. The best results within the infrastructure of Bremen were obtained with the values $c_1 = 6$, $c_1 = 5$, and $\beta_a^0 = 10$. With these values the mechanism computed fair allocations with an adequate total distance. The parameters scale with the distances an agent must drive. Consequently, they are dependent on the infrastructure of the catchment area.

6.3. Comparison of fairness

To evaluate the approach, we compared the generated distribution of both algorithms. We simulated different scenarios with five to ten couriers working concurrently.

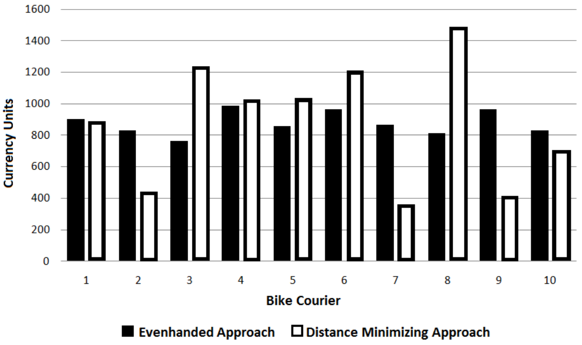


Figure 5: The Black Bars Show the Revenue of 10 Bike Couriers After One Week using the Evenhanded Dispatching Approach, while the White Bars Depicts the Revenues using the Distance Minimizing Algorithm.

Figure 5 shows the revenue of the bike couriers after a whole week. Respectively, Figure 6 depicts the covered distances. While the maximum pay gap in distance minimized distributions is 75.7%, the highest difference in evenhanded allocations is only 22.6%. Additionally, the maximum difference in covered kilometers varies up to 72.0% using the distance minimizing algorithm and up to 30.1% using the evenhanded approach. Indeed, after one week the pay gaps using distance minimizing algorithms are significantly higher than using the evenhanded approach.

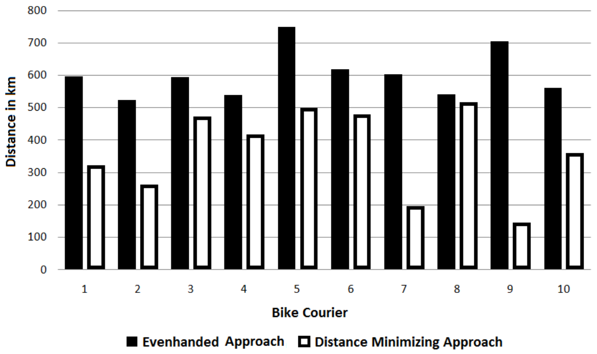


Figure 6: The Black Bars Show the Covered Distances of 10 Bike Couriers After One Week using the Evenhanded Dispatching Approach, while the White Bars Depict the Covered Distances using the Distance Minimizing Algorithm.

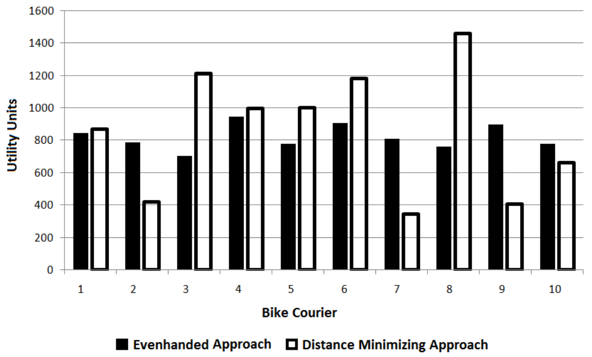


Figure 7: The Black Bars Show the Bike Courier's Utility of Orders of 10 Bike Couriers After One Week using the Evenhanded Dispatching Approach, while the White Bars Depicts the Utility Values using the Distance Minimizing Algorithm.

Figure 7 shows the bike courier's utility of all orders defined by Definition 3 of the whole week. As the fitness value affects the utility value, we computed the utility value for each day and summed up the results. The utility values are used to determine the fairness of the distribution with respect to Definition 4. While the average standard deviation σ of each participant's individual utility value is 1153.82 utility units using the distance minimizing algorithm, it is only 226.69 units in the evenhanded allocation. The obtained results reveal deviations higher than 80.36% concerning the fairness of the allocations.

However, Figure 6 indicates that the total distance increases from 3672,78 km to 6038,24 km and every courier has to drive more kilometers than in the distance minimized distribution. Therefore, we investigated separate days individually. Figure 8 shows that the evenhanded approach is even significant fairer after one day in contrast to the distance minimizing algorithm. In addition, it shows that courier No. 8 has to drive nearly half of the distance of the entire week on one day.

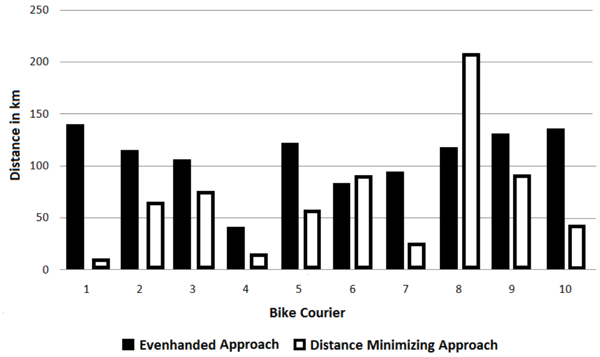


Figure 8: The Black Bars Show the Driven Distances of 10 Bike Couriers After One Day using the Evenhanded Dispatching Approach, while the White Bars Depict the Driven Distances using the Distance Minimizing Algorithm.

Other results from scenarios with less bike couriers (not shown in diagrams) pinpoint that the reduction of bike couriers lead to a decreasing number of orders operated within the guaranteed time windows.

6.4. Discussion

Several simulated scenarios based on real customer orders revealed that distance minimizing algorithms are not applicable for the dispatching of bike couriers who are paid by commission fees on a daily basis. The results show that the utility, the driven distances as well as the revenue of the couriers are unequally distributed and varying significantly using the distance minimizing algorithm. In contrast, our approach creates fair distributions if we compare the effort as well as the individual revenue of all couriers. In addition, the fairness as specified in Definition 4 indicates that the evenhanded approach computes fair distributions.

However, the driven distances of the whole week are higher for all couriers. Nevertheless, the results show that it is reasonable to cover larger distances in the entire week, if the total distance is equally distributed over the days. The courier does not prefer driving the most kilometers on one day while waiting for requests on all other days. He would pass over his physical limits on one day. Moreover, the intention of a courier is to carry shipments and not to wait at the central depot and earn even less money. The reduction of operating couriers is not reasonable because customer requests cannot be determined in advance and must be serviced within short time windows. In real scenarios, cost-intensive external carriers are instructed to deliver the orders in time, if not enough couriers are available.

Autonomous logistic processes based on mulitagent technology ensure a high flexibility and adaptivity in dynamic environments. In addition, it allows the modeling of a system with heterogeneous agents having individual capabilities. For instance, the maximum possible velocity and the fitness of represented couriers may differ.

In summary, the evenhanded approach enables a fair allocation of orders to bike couriers by a multiagent system with autonomous selfish acting agents with a reasonable increase of the driven distances.

7. CONCLUSION AND OUTLOOK

In this paper we presented a new approach to generate fair distributions for the dynamic *pickup and delivery problem* (PDP) and showed its application to bike couriers.

The mulitagent-based simulation platform PlaMSA served to evaluate the approach within the transport infrastructure of Bremen. We dispatched real customer requests provided by a German bike courier company.

The results reveal that applying distance minimizing algorithms is not pertinent, whereas our fairness approach motivates couriers and distributes the effort as well as the revenue equally to the bike couriers with respect to their fitness.

Automated evenhanded dispatching enables new business models for commercial internet dispatching platforms of bike couriers implementing autonomous logistic processes. Couriers can register themselves with the agent system by their mobile application for receiving orders. Afterwards, the mulitagent system computes evenhanded allocations of orders to bike couriers and suggests routes while taking into accout the dynamics of the environment. Therefore, the approach enables large scale online dispatching platforms which can be used by couriers and retailers to offer reliable deliveries as well as sustainable transportation.

With respect to new transport visions and the increasing usage of e-bikes for the transport of heavy goods the relevance of automated dispatching platforms for logistic transport service providers paid by commission fees is increasing.

To decrease the total effort and distribute the revenue equally, also profit sharing methods for freight carriers may be considered (e.g., Krajewska, Kopfer, Laporte, Ropke, and Zaccour 2007). However, profit sharing concepts changes the pricing system and couriers are not paid by commission fees anymore. Consequently, this effects the agent's strategies as well as their incentive to deliver shipments.

Further investigations will focus on the integration of traffic simulation within the PlaSMA simulation framework as well as on modeling and simulation of unexpected events to evaluate the evenhanded mechanism in dynamic environments. Another point of interest is how a heterogeneous fleet affects the fairness of a distribution. We seek to carry over techniques well-proven in autonomous logistic processes such as team formation and cooperation of selfish acting agents.

In order to develop new potential for horizontal cooperation, extra change depots in inner city districts where couriers can exchange shipments are of special interest.

ACKNOWLEDGMENTS

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

Ahuja, R. K., Mehlhorn, K., Orlin, J. B., & Tarjan, R. E. (1990). Faster Algorithms for the Shortest Path Problem. *Journal of the ACM*, *37*(2), 213-223.

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

Berbeglia, G., Cordeau, J.-F., & Laporte, G. (2010). Dynamic Pickup and Delivery Problems. *European Journal of Operational Research*, *202*(1), 8-15.

Brooks, R. (1986). A Robust Layered Control System for a Mobile Robot. *IEEE Journal of Robotics and Automation*, *2*(1), 14-23.

Bräysy, O., & Gendreau, M. (2005a). Vehicle Routing Problem with Time Windows, Part I: Route Construction and Local Search Algorithms. *Transportation Science*, *39*(1), 104-118.

Bräysy, O., & Gendreau, M. (2005b). Vehicle Routing Problem with Time Windows, Part II: Metaheuristics. *Transportation Science*, *39*(1), 119-139.

Böse, F., & Windt, K. (2007). Catalogue of Criteria for Autonomous Control in Logistics. In M. Hülsmann & K. Windt (Eds.), *Understanding Autonomous Cooperation and Control in Logistics – The Impact of Autonomy on Management, Information, Communication and Material Flow* (pp. 57-72). Berlin: Springer-Verlag.

Dijkstra, E. W. (1959). A Note on Two Problems in Connexion with Graphs. *Numerische Mathematik*, *1*(1), 269 - 271.

Dumas, Y., Desrosiers, J., Gelinas, E., & Solomon, M. (1995). An Optimal Algorithm for the Traveling Salesman Problem with Time Windows. *Operations Research*, *43*, 367-371.

Dumas, Yvan, Desrosiers, J., & Soumis, F. (1991). The Pickup and Delivery Problem with Time Windows. *European Journal of Operational Research*, *54*(1), 7-22.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining* (pp. 226-231).

Fischer, K., Müller, J. P., & Pischel, M. (1995). Cooperative Transportation Scheduling: An Application Domain for DAI. *Journal of Applied Artificial Intelligence*, *10*, 1-33.

Gehrke, J. D., & Ober-Blöbaum, C. (2007). Multiagent-Based Logistics Simulation with PlaSMA. In M. Koschke, R.; Herzog, O.; Rödiger, K.-H.; Ronthaler (Ed.), *Informatik 2007 - Informatik trifft Logistik* (pp. 416-419).

Himoff, J., Rzevski, G., & Skobelev, P. (2006). Multi-Agent Logistics i-Scheduler for Road Transportation. *Proceedings of 5-th International Conference on Autonomous Agents and Multi Agent Systems (AAMAS 2006)* (pp. 1514-1521).

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

Hülsmann, M., & Windt, K. (Eds.). (2007). *Unterstanding Autonomous Cooperation and Control in Logistics*. Berlin: Springer-Verlag.

Jennings, N. R. (2001). An Agent-Based Approach for Building Complex Software Systems. *Communications of the ACM*, *44*(4), 35-41.

Krajewska, M. a, Kopfer, H., Laporte, G., Ropke, S., & Zaccour, G. (2007). Horizontal Cooperation among Freight Carriers: Request Allocation and Profit Sharing. *Journal of the Operational Research Society*, *59*(11), 1483-1491.

Kuske, S., Luderer, M., & Tönnies, H. (2010). Autonomous Units for Solving the Traveling Salesperson Problem Based on Ant Colony Optimization. In H.-J. Kreowski, B. Scholz-Reiter, & K.-D. Thoben (Eds.), *2nd International Conference on Dynamics in Logistics (LDIC 2009)* (pp. 289 - 298). Springe-Verlag.

Neagu, N., Dorer, K., Greenwood, D., & Calisti, M. (2006). LS/ATN: Reporting on a Successful Agent-Based Solution for Transport Logistics Optimization. *Distributed Intelligent Systems: Collective Intelligence and Its Applications, 2006. IEEE Workshop* (pp. 213-218).

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008). A Survey on Pickup and Delivery Problems Part II: Transportation between Pickup and Delivery Locations. *Journal für Betriebswirtschaft*, *58*(2), 81-117.

Rao, A. S., & Georgeff, M. P. (1995). BDI Agents: From Theory to Practice. *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)* (pp. 312-319).

Sandholm, T. W. (1999). Distributed Rational Decision Making. In G. Weiss (Ed.), *Multiagent Systems: A*

Modern Approach to Distributed Artificial Intelligence. The MIT Press.

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

Schuldt, A. (2011). *Multiagent Coordination Enabling Autonomous Logistics*. Heidelberg, Germany: Springer-Verlag.

Schuldt, A., Gehrke, J. D., & Werner, S. (2008). Designing a Simulation Middleware for FIPA Multiagent Systems. *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology* (pp. 109-113). IEEE Computer Society Press.

Shoham, Y., & Leyton-Brown, K. (2009). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press.

Skobelev, P. (2011). Multi-Agent Systems for Real Time Resource Allocation , Scheduling , Optimization and Controlling : Industrial Applications. In V. Mařík, P. Vrba, & P. Leitão (Eds.), *Holonic and Mulit-Agent Systems for Manufacturing: Volume 6867 of Lecture Notes in Computer Science* (pp. 1-14).

Solomon, M. (1987). Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*, *35*, 254-265.

Vickrey, W. (1961). Counterspeculation, Auctions, and Competitive Sealed Tenders. *Journal of Finance*, *16*(1), 8-37. JSTOR.

Warden, T., Porzel, R., Gehrke, J. D., Herzog, O., Langer, H., & Malaka, R. (2010). Towards Ontology-based Multiagent Simulations: The PlaSMA Approach. In A. Bargiela, S. Azam Ali, D. Crowley, & E. J. . Kerckhoffs (Eds.), *24th European Conference on Modelling and Simulation. European Council for Modelling and Simulation* (pp. 50 - 56).

Wessels, A., Jedermann, R., & Lang, W. (2010). Transport Supervision of Perishable Goods by Embedded Context Aware Objects. *WSEAS Transactions On Circuits And Systems*, *9*(5), 295-304.

Windt, K., & Hülsmann, M. (2007). Changing Paradigms in Logistics – Understanding the Shift from Conventional Control to Autonomous Cooperation and Control. In M. Hülsmann & K. Windt (Eds.), *Understanding Autonomous Cooperation and Control in Logistics – The Impact of Autonomy on Management, Information, Communication and Material Flow* (pp. 4-16). Berlin: Springer-Verlag.

Max Gath received his diploma in Computer Science from the University of Bremen (Universität Bremen) in October 2011. Subsequently, he joined the Artificial Intelligence group at the Center for Computing and Communication Technologies (TZI). He is associated with the Collaborative Research Centre 637 Autonomous Logistic Processes - A Paradigm-Shift and its Limitations (CRC 637) and working on the transfer project “Autonomous Groupage Traffic - Agent-Based Autonomous Control in Groupage Traffic”.

Thomas Wagner is a senior researcher at the Artificial Intelligence Research Group of the University of Bremen. He studied Artificial Intelligence and Computerlinguistics at the University of Osnabrück. He holds a PhD from the University of Bremen on "*Qualitative Navigation in Unstructured Environments*" and worked on different projects with varying topics like knowledge based configuration, qualitative spatial reasoning (e.g., in the *Robocup*), knowledge representation and machine learning.

Otthein Herzog started out to study Computer Science at the Universitaet Karlsruhe in summer 1969 and received his diploma in Applied Mathematics and Informatics from the Universitaet Bonn in 1972. He received his PhD from the Informatics Department of the Universitaet Dortmund in 1976. From 1977 to 1993 he worked with IBM where he held several technical and managerial positions in international software product development and research. His projects included mainframe operating systems, quality assurance (DOS/VSE and Unix, 1977-1984), communications software, full-text information retrieval systems (IBM SearchManager), CIM repository systems, and a system for environmental impact analysis (1991-1993). From 1985 to 1991 he directed the Institute for Knowledge-based Systems in the Scientific Center of IBM Germany, where he headed numerous AI research projects, the most important one being LILOG - Natural Language Processing and Text Understanding. From March 2000 to February 2002, Dr. Herzog was on leave from the university and held the CTO position at Lenze AG where he was responsible for hardware and software development. From 1993 to 2009, Dr. Herzog held the chair on Artificial Intelligence in the Department of Mathematics and Computer Science at the Universität Bremen and headed the Artificial Intelligence Research Group. He continues to supervise PhD students, to direct research projects, and to acquire new projects.