

Matching Drivers and Transportation Requests in Crowdsourced Delivery Systems

Full Paper

David Soto Setzke

Technical University of Munich
david.soto.setzke@in.tum.de

Maximilian Schreieck

Technical University of Munich
maximilian.schreieck@in.tum.de

Manuel Wiesche

Technical University of Munich
manuel.wiesche@in.tum.de

Christoph Pflügler

Technical University of Munich
christoph.pfluegler@in.tum.de

Sven Fröhlich

Technische Universität Dresden
sven.froehlich@tu-dresden.de

Helmut Krcmar

Technical University of Munich
krcmar@in.tum.de

Abstract

While the sales volume of e-commerce transactions is growing rapidly, the traditional concept of packages delivery has been challenged by innovative approaches such as crowdsourced delivery. Using individuals, for example commuters, to deliver packages from senders to receivers can provide several economic and environmental benefits. This paper illustrates an algorithm that automates and optimizes the assignment of drivers to transportation requests by matching them based on transportation routes and time constraints. We evaluated our algorithm by using a simulated setting based on mobility data recorded in a major German city. This paper contributes to theory by giving guidance for future research on matching algorithms for crowdsourced delivery systems and to practice by illustrating an algorithm that can be adapted by existing and new crowdsourced delivery platforms.

Keywords

Crowdsourced Delivery, Matching Algorithm, Sharing Economy, Dynamic Matching, Flow Networks

Introduction

Crowdsourced delivery, i.e. the concept of employing individuals from a large pool of people instead of professional couriers in order to deliver items from senders to receivers, has been increasingly attracting interest over the last few years (Hodson 2013; Rougès and Montreuil 2014; Sadilek et al. 2013). The concept of crowdsourced delivery is, as in the case of most crowdsourced activities, realized through an online platform that can be used by organizations or individuals to publish transportation requests. Other users can register on the platform, accept and carry out requests and receive a compensation from the original sender or the platform operator. In the United States, startups that offer platforms for crowdsourced delivery services, such as Postmates and Deliv, have acquired investments of 22 and 14 million US\$ respectively (Rougès and Montreuil 2014). The concept has also captured the interest of well-established retail companies such as Amazon and WalMart, which are testing crowdsourced delivery services in selected areas in the United States (Amazon.com 2017; Bender 2016).

However, existing platforms focus mainly on drivers who set time aside dedicated to carrying out submitted transportation requests. In most cases, drivers are given a list of preselected transportation requests that match their current location and their availability times in order to choose requests they wish to carry out. In their study, Rougès and Montreuil (2014) propose the use of matching algorithms for optimizing the assignment of transportation requests to drivers. Such an algorithm could improve the efficiency of crowdsourced delivery platforms by optimizing potential matches and further automating the matching process. However, there is a lack of research in the field of algorithms matching drivers and items based on transportation routes.

In this paper, we present an extendable algorithmic model that matches transportation requests with drivers whose travel plans are known in advance. To build the model artifact, we follow a design science research approach (Peffer et al. 2007). We tested the effectiveness of our model with a large set of transportation requests and planned trips that were partly generated randomly, partly based on real mobility data recorded in a major German city. We draw on the research of Agatz et al. (2011), who have constructed a solution for a related matching problem in the context of dynamic ridesharing systems. Our algorithm can be adapted and used by existing and new crowdsourced delivery platforms in order to automate the process of matching commuter trips with transportation requests and thus leverage already existing travel plans of commuters. The focus on matching requests with already announced travel plans allows the integration of deliveries into the daily tasks and routes of commuters. We expect a higher adoption of the concept of crowdsourced delivery since deliveries are combined with already planned trips. Thus, commuters need to set only minimal time aside during their daily travels in order to carry out assigned transportation requests.

The remainder of this paper is structured as follows. First, we present the theoretical background on the concept of crowdsourced delivery. Second, we present our algorithmic model and illustrate it by means of a simplified example. Third, we discuss results that we obtained by applying our algorithmic model to a larger, more realistic problem size in order to demonstrate the effectiveness of our algorithm. Fourth, we discuss the limitations of our research and provide suggestions for future research. Finally, we complete the paper with a discussion and a conclusion.

Theoretical Background

In their study, Schöder et al. (2016) show that B2C e-commerce sales on a global scale are expected to almost double from 2013 to 2018, as can be seen in Figure 1. This increase implies a rise of package deliveries and along with it, negative environmental impacts such as increased pollution and traffic noise. Consequently, several studies have been conducted on alternative concepts for the last mile of package delivery (Hübner et al. 2016; Joerss et al. 2016; Slabinac 2015).



Figure 1: Global B2C e-commerce sales and percentage of GDP from 2013 to 2018 (in Billion US\$) (Schöder et al. 2016)

One of those concepts is crowdsourced delivery, i.e. the application of the concept of crowdsourcing to the process of package delivery. According to Estellés-Arolas and González-Ladrón-de-Guevara (2012), crowdsourcing is “a type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals [...] the voluntary undertaking of a task”. According to McKinnon (2016), crowdsourced delivery entails a variety of benefits, such as a cut of traffic levels, energy use, and emissions, as well as competitive differentiation for retailers regarding delivery speed and costs. Especially for traditional brick-and-mortar stores that often lack the resources for building up or joining an existing delivery network, crowdsourced delivery represents a mean of remaining competitive (Dörrzapf et al. 2016). The short-term availability of so-called crowd workers (Estellés-Arolas and González-Ladrón-de-Guevara 2012) could allow small and medium-sized retailers to offer premium services such as same-day delivery which, at the moment, is offered mostly by large retail corporations such as Amazon.com (MacKenzie et al. 2013). Nonetheless, crowdsourced delivery could also be used by private individuals who want to avoid the potentially high costs and delivery times for sending packages (Mladenow et al. 2015).

As McKinnon (2016) points out, the market for crowdsourced delivery services is relatively young, with most of the platforms having been launched over the past 6-7 years. While startups such as Deliv, Zipments and Postmates are growing and acquiring larger investments, also already established transport businesses like Uber and DHL are expanding their platforms and services to include crowdsourced transportation of goods by testing their services uberRUSH and DHL MyWays in selected areas. According to McKinnon (2016), especially the recent entry of Amazon with its service Amazon Flex, which is currently testing crowdsourced package delivery in 29 U.S. cities, will further intensify the competition among existing crowdsourced delivery platforms.

Rougès and Montreuil (2014) suggest that the process of matching transportation requests and potential couriers can be automated by using algorithms. Services like Amazon Flex or uberRUSH provide certain filter criteria such as item size, availability times and location for preprocessing possible matches. Based on these criteria, potential couriers can accept or reject incoming requests or choose them from a list. Note that in most cases, the task of delivering an item is considered the main activity of the courier at a current point in time. Another match criterion can be introduced by considering drivers who deliver packages on their way to another destination such as their workplace. In this case, we can evaluate potential matches on how well they fit to the already planned route of the courier. The crowdsourced delivery service Hitch is targeted at commuters as potential couriers. However, instead of calculating and suggesting the optimal matches according to the commute and the transportation route, it displays delivery routes on a map and requires potential couriers to choose their matches on their own (Caluzo 2014). By considering travel routes and suggesting matches based on these routes, we can improve the efficiency of crowdsourced delivery tasks, since this helps couriers to integrate these tasks into their daily life activities, without the need to set time aside dedicated to delivering items with random delivery routes.

As of now, there is a lack of research on matching drivers and items based on their transportation routes. The concept of ridesharing systems, i.e. “an automated system that facilitates drivers and riders to share one-time trips close to their desired departure times” (Agatz et al. 2012) is a similar problem that has been covered more extensively by existing literature (Masoud and Jayakrishnan 2015). In ridesharing systems, drivers’ ride offers and riders’ ride requests need to be matched taking into account ride routes and time constraints, i.e. earliest departure and latest arrival times. In their work, Agatz et al. (2011) propose the use of a maximum-weight bipartite matching model to represent the dynamic ridesharing problem. While the initial problem of their work seems to be similar to the crowdsourced delivery problem, their solution cannot be transferred directly to this problem domain since their assumptions about riders’ behavior and properties differ in certain ways from those of items. For example, all the riders in the case of Agatz et al. (2011) assumingly own a car that they can use to complete the ride by themselves and to pick up other riders if no suitable ride offer can be found. Furthermore, ride requests are assumed to be aimed at planned rides such as work trips, which typically have a low time flexibility between the earliest departure and the latest arrival date. As opposed to this, items tend to have a much larger time flexibility, even in the case of same-day delivery.

In the following section, we construct an algorithm that matches drivers and items, taking into account the special properties of items in crowdsourced delivery systems as opposed to the properties of riders in ridesharing systems.

Problem Description and Matching Algorithm

In the following, we present a formal problem description and an algorithmic model that solves the task of matching transportation requests to drivers. Furthermore, we illustrate the execution of the presented algorithm by providing and calculating an exemplary problem instance. We close the section by explaining how the algorithm at hand can be extended and adapted to different requirements.

Problem Description

Our model considers a set of driver trips and a set of transportation requests. Every trip starts at a departure location and ends at a destination location while every item corresponding to a request has a pickup location and a delivery location. Trips have both an earliest departure time and a latest arrival time while items have an earliest pickup time and a latest delivery time. We measure the travel distance between two locations in minutes. Since the assignment of senders and receivers is based on travel routes, the locations and times of

all trips and requests need to be known in order to be matched. However, this information is allowed to change as long as the trip or request has not been matched yet. Table 1 lists the symbols that we are going to use throughout the remainder of this paper.

Symbol	Definition and Description
T	Set of trips
R	Set of transportation requests
P	Set of all entities ($P = T \cup R$)
T_p^{ED}	Earliest departure/pickup time for an entity $p \in P$
T_p^{LA}	Latest arrival/delivery time for an entity $p \in P$
S	Set of all coordinates of the relevant area
O_p	Departure/Pickup location for an entity $p \in P$ ($O_p \in S$)
D_p	Destination/Delivery location for an entity $p \in P$ ($D_p \in S$)
d_{ab}	Travel time in minutes between two locations $a, b \in S$
$time_feasible$	Predicate that determines whether a match $(t, r) \in T \times R$ is time feasible, i.e. whether the transportation request can be carried out during a given trip without violating the time constraints given by $T_t^{ED}, T_r^{ED}, T_t^{LA}$ and T_r^{LA}

Table 1: Symbols, definitions, and descriptions

Given this information, we are looking for an optimal assignment of trips to transportation requests, i.e. finding a maximum matching with a certain minimal cost, which will be further explained later, for a bipartite graph consisting of nodes representing the trips T and the transportation requests R .

Algorithmic Model

In order to compute a maximum matching for our bipartite graph, we construct and solve a minimum-cost maximum-flow problem. In a flow network, a certain amount of flow is sent from a source vertex s along several edges to a sink vertex t . Each edge has a cost, a maximum capacity and a flow value. In the minimum-cost maximum-flow problem, we aim to find the maximum flow with minimum cost. We construct a directed graph $G = (V, E)$, where:

$$V = T \cup R \cup \{s, t\}$$

$$E = \{(i, j) \in T \times R \mid time_feasible(i, j)\} \cup \{(s, i) \mid i \in T\} \cup \{(j, t) \mid j \in R\}$$

We inserted a newly introduced source vertex s and a sink vertex t . We included all trips and requests as nodes and made sure that any potential match is time feasible. Furthermore, we introduced edges from the source vertex s to all trips and from all requests to the sink vertex t . In order to quantify the quality of a match and assign a cost to each edge, i.e. a potential match, we introduce a cost function $\Gamma: E(G) \rightarrow \mathbb{R}$:

$$\Gamma(i, j) \begin{cases} d_{O_i, O_j} + d_{O_j, D_j} + d_{D_j, D_i} - d_{O_i, D_i}, & (i, j) \in T \times R \\ 0, & else \end{cases}$$

This cost function computes the additional time that a driver i needs to spend in order to carry out a transportation request j . For any edge that is connected to the source or the sink vertex, we set the cost to 0. We aim for a set of matches with low weights, i.e. matches with less additional driving time are regarded “better” than matches with more additional driving time. The cost function can be adapted to take other factors into account, which we will explain in more detail in the closing of this section. Furthermore, we assign a capacity and a flow value of 1 to each edge $(i, j) \in E(G)$. We can now use this model to send a flow from the source vertex s along the edges to the sink vertex t . Since we are looking for a maximum match, we want to maximize the flow that is being sent to the sink vertex. The uniform capacity of 1 ensures that there will be at most one outgoing edge for any trip and at most one incoming edge for any request.

Figure 2 illustrates the network that we constructed. The edge labels indicate the cost value of the corresponding edge, i.e. the additional travel time needed for this particular match. We omitted capacity and flow values, as well as the cost values for any edges connected to the source or sink vertices since these will remain constant for any problem instance.

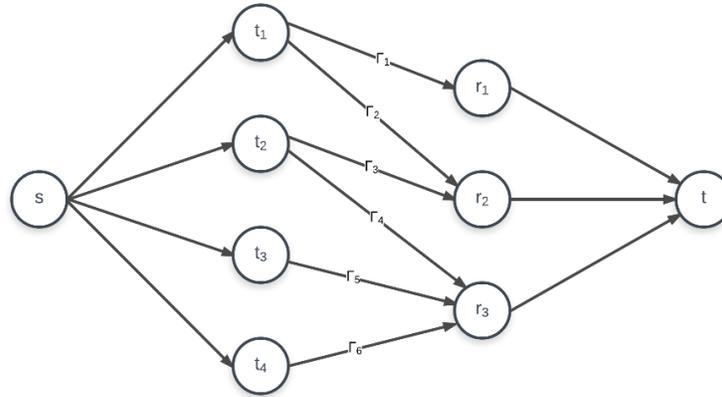


Figure 2: Minimum-cost maximum-flow problem

Example Calculation

In the following, we present an exemplary problem instance in order to demonstrate the algorithm execution. Table 2 and Table 3 respectively list fictional trips and transportation requests that are known at the beginning of a day. Each trip and request is presented along with the information needed to calculate the cost of a potential match: Departure/Pickup times, arrival/delivery times, and start/destination locations. Note that in a real-world setting, the list of trips and transportation requests is expected to be significantly larger.

Symbol	Earliest Departure Time	Latest Arrival Time	Departure Location	Arrival Location
t_1	7:00 a.m.	8:00 a.m.	l_1	l_2
t_2	5:00 p.m.	7:00 p.m.	l_2	l_1
t_3	10:00 a.m.	11:00 a.m.	l_3	l_4
t_4	5:00 p.m.	7:00 p.m.	l_4	l_3

Table 2: Example trips

Symbol	Earliest Pickup Time	Latest Delivery Time	Pickup Location	Delivery Location
r_1	4:00 p.m.	6:00 p.m.	l_5	l_6
r_2	1:00 p.m.	5:00 p.m.	l_7	l_8
r_3	7:00 a.m.	6:00 p.m.	l_5	l_8

Table 3: Example transportation requests

Recall that we need to calculate the time that a driver needs to spend additionally in order to carry out a transportation request corresponding to a certain potential match. Figure 3 illustrates the initial situation of our problem instantiation. For all potential matches, the additional travel time is displayed as a label on the corresponding edge.

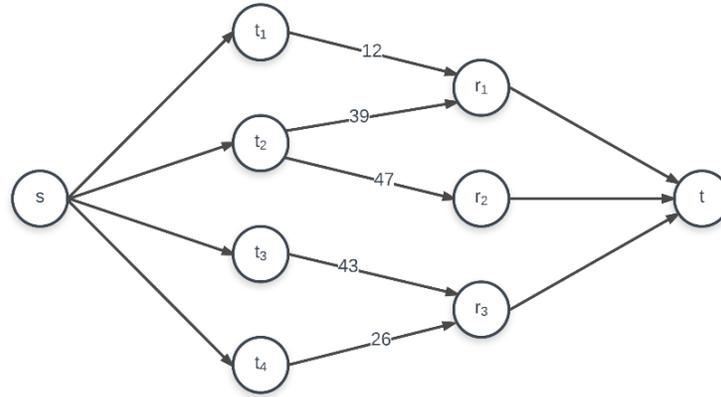


Figure 3: Initial situation of the example problem instantiation

We now compute the maximum flow with minimum cost for this problem instance, which transforms the network from Figure 3 to the solved state that is shown in Figure 4. Of the five initial edges, two have been eliminated and we are left with the set of optimum matches, i.e. request r_1 should be delivered during the trip t_1 , r_2 during t_2 , and r_3 during t_4 .

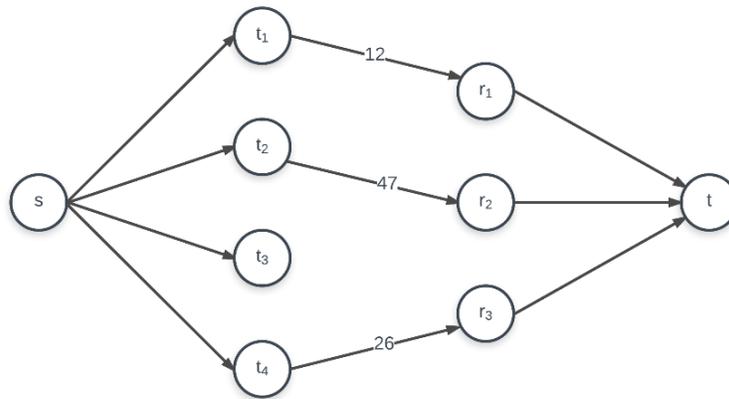


Figure 4: Solved state of the example problem instantiation

Expandability and Adaptability

The cost function Γ can be modified in order to consider other factors, thus representing a more general cost function where additional travel time spent is seen only as one of several contributing factors or, alternatively, not even considered. For example, a real-world crowdsourced delivery platform could include a rating system which assigns a rating to each courier. If a courier does not satisfyingly carry out a request, e.g., a package is delivered too late or damaged, the platform could assign a bad rating to the corresponding courier while couriers who carry out requests to the satisfaction of the contracting company would receive better ratings. The rating could then be included as a factor in the cost function which either raises or lowers the cost for a potential match. Furthermore, certain companies might prefer to employ couriers who have already delivered other packages from this company in the past. In this case, the cost function could give a preference to these couriers when considering them for a potential match. These preferences could also be coupled to certain dynamic conditions, i.e. at times when there are many packages waiting for delivery and a relatively low amount of available drivers, the cost function could neglect these preferences in order to aim for a maximum coverage of transportation requests.

Evaluation

In this section, we are going to present the evaluation of our developed algorithm by using a simulated setting. As a performance indicator for our algorithm, we used the achieved match rate, i.e. the ratio of matched requests to the total number of requests for a certain period. For our evaluation, we also examined the impact of the following two factors on the match rate:

- Ratio of trips to transportation requests
- Planning horizon interval, i.e. the time interval that is used for the calculation of matches throughout a single day

For our simulated setting, we used floating car data of about 500 taxis recorded throughout the course of several months in a major German city. The taxis were equipped with GPS trackers that send their current position every 5 seconds to a central server. Furthermore, additional information, such as whether the taxi is currently on a passenger trip, was sent. We randomly chose one day of data from the entire recorded timeframe and extracted all the passenger trips for this day. Since the departure and destination location were already given, we only needed to generate the earliest departure and latest arrival time. For each passenger trip, we assigned a time flexibility of one third of the actual trip time. Regarding the transportation requests, we generated random pickup and drop-off locations located in the same city. For generating time windows, we assumed that items can be picked up between 8 a.m. and 12 noon. In order to simulate same-day delivery, each item needs to be delivered latest at 7 p.m.

For analyzing the impact of the used planning horizon, we assumed that transportation requests are created during the course of the day and that trips are planned in advance by the corresponding drivers. Furthermore, we assumed that requests need to be announced by no later than 1 hour before their earliest pickup date. We assume that drivers can either accept or reject matches. In the case where a match is rejected, the corresponding driver and request will be considered again in the next match calculation. In order to model this system property, we assumed that the decision whether a match is accepted or rejected depends on the additional time that needs to be spent by the driver in order to deliver the corresponding item. We computed the ratio between the additional driving time and the total time window of a request and used it as the probability of a match rejection. At last, we simulated the continuous calculation of matches throughout the course of the day by dividing the total time window of operations (8 a.m. to 6 p.m.) into several time windows, depending on the chosen time interval. At each calculation, we included only requests that would already have been announced at the time of calculation. In order to compare instant match calculation with a planning horizon approach, we additionally approximated an instant match strategy by using a time interval of 1 minute. At each interval step, we calculated matches only if a new request would have been announced at this point in time. For each rejected match, we instantly calculated new matches until either a match was accepted or no further matches were found.

The calculations were performed using the Google Optimization Tools framework. We used a fixed trip set size of 1000 trips and ensured that generated trips have a total trip time between 5 minutes and 2 hours and a total trip distance between 5 and 100 kilometers. For analyzing the impact of the request/trip ratio, we considered 10 scenarios with request set sizes between 100 and 1000. For each scenario, we solved the problem 10 times, each time with newly generated data, and computed the average of the resulting match rates.

Figure 5 visualizes the impact of the request/trip ratio on the match rate, i.e. the percentage of matched transportation requests. The match rate remains almost constantly at 100% until a request/trip ratio of roughly 40% has been reached and then declines rapidly. For our setting, the request/trip ratio with the optimal match rate can thus be found at around 40%.

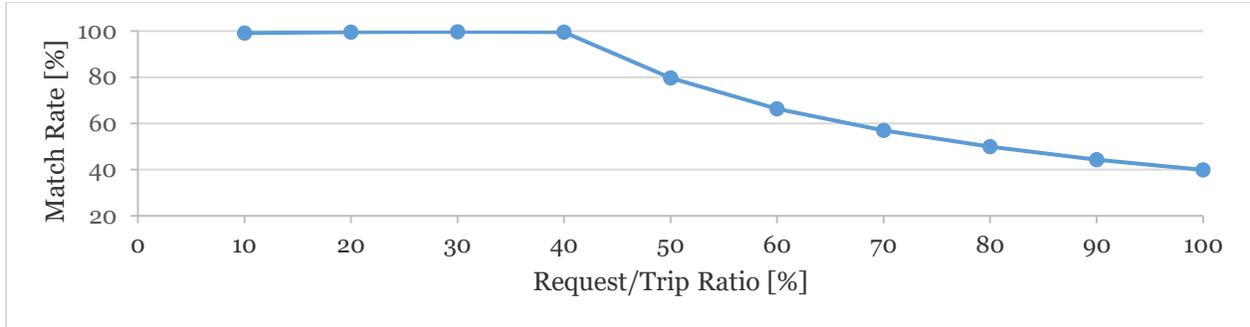


Figure 5: Impact of the request/trip ratio on the match rate

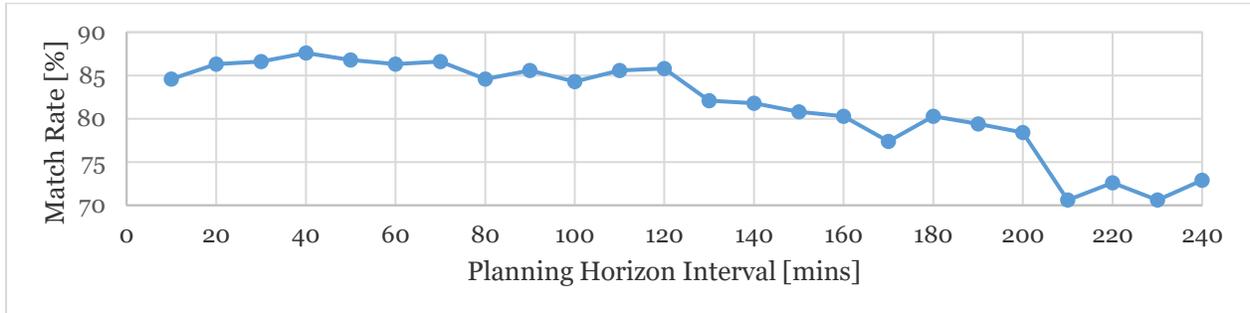


Figure 6: Impact of the planning horizon interval on the match rate

For analyzing the impact of the planning horizon interval, we generated a scenario of trips and requests with an optimum request trip ratio of 40%. We then used different planning interval times to test the scenario, each time increasing the planning interval by 10 minutes. The values for each scenario were obtained by simulating the passing of a day by executing several match computation cycles and updating a simulated clock. At each computation, the value of the clock was incremented accordingly to the planning interval and at each computation, we used the clock to determine which requests would already have been announced at this point in time and which requests and trips could still be completed. Requests would then be accepted or rejected according to the mentioned probability distribution.

Figure 6 shows the impact of the planning horizon interval on the match rate. We can see that the match rate remains relatively high until it reaches its optimum at an interval of 40 minutes with a rate of 87.6%. After that, it slowly declines but does not fall under 70%. We therefore suggest an interval between 20 and 80 minutes for achieving optimal results. Note that, as opposed to the analysis of the impact of the request/trip ratio, in the case of a rolling horizon strategy we are not achieving a match rate of 100% since transportation requests are published as the day progresses, when certain matchable trips may have already been matched with past requests.

For our instant match approximation, we obtained a match rate value of 47.2%, which is significantly lower than the optimum in the rolling horizon case. Thus, we recommend the use of a rolling horizon strategy over the use of instant match calculation.

Discussion

Limitations and Future Research

Our findings are subject to some limitations. First, our performance evaluation is mostly based on hypothetical and generated data that might not represent conditions in a real-world setting. For instance, a uniform distribution of coordinates in a certain area might not be suitable for modeling item delivery origin and destination locations since origin locations tend to be situated in industrial or commercial areas while destination locations tend to be situated in residential areas. With regard to potential couriers, taxi drivers as well as passengers are part of the target group for a crowdsourced delivery platform. However, the target group also includes, for example, commuters who use their privately owned car for driving to

their places of work. In order to further validate the effectiveness and usefulness of our algorithmic model, future research could evaluate the algorithm in more comprehensive simulation settings.

Another limitation of our conceived algorithm is its current inability to assign multiple requests to one trip. We expect that the rate of matched transportation requests can be increased even more if drivers can carry out multiple transportation requests at once. A possible solution for this problem could include an adaptation of the edge capacity values so that a trip can be matched with multiple transportation requests. Furthermore, we would need to modify the time feasibility check and the cost function in order to account for more complex routes that involve pickup and delivery locations for more than one item.

Another potential optimization of our algorithm could be the possibility to transfer items between multiple drivers. In some cases, a driver might not be able to deliver an item to its final destination because of time constraints but he could take it to a certain location where it is passed on to another driver who ultimately delivers it to the final drop off location. However, this would also create the need for mechanisms for short-term recalculations, e.g. in the case when one or multiple drivers in the delivery chain do not arrive at the transfer location in time.

While an effective matching process can be an important asset for a crowdsourced delivery platform, it also needs to be aligned with non-technical aspects, such as the platform's business model and business processes. This includes pricing and driver compensations, as well as handling critical events, such as short-term package delivery rejections from drivers. Although so far, little research has been conducted on business models for crowdsourced delivery (Rougès and Montreuil 2014; Sadilek et al. 2013), future research can draw on business models for general crowdsourcing platforms (Cefkin et al. 2014) and, in particular, crowd logistics platforms (Mladenow et al. 2015).

Implications

We contribute to theory and practice by giving guidance on future research on matching algorithms for crowdsourced delivery systems. Our algorithm can be used as a basis for further development and afterwards, the integration into new or existing crowdsourced delivery platforms that aim at engaging commuters for delivering packages. Depending on the concrete business models and processes, developers will need to adapt or extend the algorithm. As of now, it ignores aspects such as performance ratings of couriers or preferences for specific couriers although this might be vital for certain platforms.

In order to complete the matching process, the algorithm needs start and destination locations as well as time constraints for driver and item transportation routes. Consequently, this means that platform users, i.e. both item senders and potential couriers need to provide this data which will then be stored by the platform. Along with the fact that this data needs to be updated in the case of changes in order to further guarantee correct assignments, this raises certain challenges for platforms adopting our algorithm. Users might find it tedious or might forget to update changed trip data. In order to avoid outdated trip or request data and to facilitate the data administration process, the platform could retrieve data directly from the calendars of registered potential couriers and, in the case of commercial item senders, online sales systems. However, even in the case of changes that have not been entered or propagated correctly, or in the case of short-term changes that affect matches that were already fixed, the platform needs to make sure that affected matches will be recalculated so that the item can be delivered in time. While, based on our evaluation results, we recommend a rolling horizon strategy for computing matches, in this situation a dynamic adaption of the match computation strategy might be appropriate.

Conclusion

As an alternative and innovative concept for the last mile of package delivery, crowdsourced delivery has been gaining more and more attention from both researchers and practitioners. Initial studies show that it has the potential to provide several economic and societal benefits. In this paper, we developed and evaluated a dynamic matching algorithm for crowdsourced delivery platforms that assigns items to potential drivers. Our algorithm aims at integrating the process of delivering items into already planned driver routes in order to minimize the additional travel time that drivers need to spend and to use already available transportation resources more effectively.

Acknowledgements

We thank the German Federal Ministry for Economic Affairs and Energy for funding this research as part of the project 01MD15001D (ExCELL).

REFERENCES

- Agatz, N., Erera, A., Savelsbergh, M., and Wang, X. 2012. "Optimization for Dynamic Ride-Sharing: A Review," *European Journal of Operational Research* (223:2), pp. 295-303.
- Agatz, N.A.H., Erera, A.L., Savelsbergh, M.W.P., and Wang, X. 2011. "Dynamic Ride-Sharing: A Simulation Study in Metro Atlanta," *Transportation Research Part B* (45:9), pp. 1450-1464.
- Amazon.com, Inc. 2017. "Amazon Flex." Retrieved 23/02/2017, from <https://flex.amazon.com>
- Bender, M. 2016. "Piloting Delivery with Uber, Lyft and Deliv." Retrieved 23/02/2017, from <http://blog.walmart.com/business/20160603/piloting-delivery-with-uber-lyft-and-deliv>
- Caluzo, Inc. 2014. "Hitch Crowdsourced Delivery." Retrieved 23/02/2017, from <http://www.hitchit.co>
- Cefkin, M., Anya, O., Dill, S., Moore, R., Stucky, S., and Omokaro, O. 2014. "Back to the Future of Organizational Work: Crowdsourcing and Digital Work Marketplaces," in: *Proceedings of the Companion Publication of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*. Baltimore, Maryland, USA: ACM, pp. 313-316.
- Dörrzapf, L., Berger, M., Breitfuss, G., and Remele, E. 2016. "Crowd Delivery Als Neues Lieferkonzept Zur Stärkung Des "Lokalen Marktplatzes"," *Proceedings of 21st International Conference on Urban Planning, Regional Development and Information Society*, Vienna, Austria, pp. 197-206.
- Estellés-Arolas, E., and González-Ladrón-de-Guevara, F. 2012. "Towards an Integrated Crowdsourcing Definition," *Journal of Information Science* (38:2), pp. 189-200.
- Hodson, H. 2013. "Hand-Delivered Parcels Find Their Way to You Via the Crowd," *New Scientist* (218:2917), 5/18/, pp. 17-18.
- Hübner, A., Kuhn, H., and Wollenburg, J. 2016. "Last Mile Fulfilment and Distribution in Omni-Channel Grocery Retailing," *International Journal of Retail & Distribution Management* (44:3), pp. 228-247.
- Joerss, M., Schröder, J., Neuhaus, F., Klink, C., and Mann, F. 2016. "Parcel Delivery: The Future of Last Mile," McKinsey & Company.
- MacKenzie, I., Meyer, C., and Noble, S. 2013. "How Retailers Can Keep up with Consumers," McKinsey & Company.
- Masoud, N., and Jayakrishnan, R. 2015. "A Real-Time Algorithm to Solve the Peer-to-Peer Ride-Matching Problem in a Flexible Ridesharing System," University of California, Irvine.
- McKinnon, A.C. 2016. "Crowdshipping: A Communal Approach to Reducing Urban Traffic Levels?," Kühne Logistics University.
- Mladenow, A., Bauer, C., and Strauss, C. 2015. "Crowdsourcing in Logistics: Concepts and Applications Using the Social Crowd," *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services*, Brussels, Belgium, pp. 1-8.
- Peffers, K., Tuunanen, T., Rothenberger, M.A., and Chatterjee, S. 2007. "A Design Science Research Methodology for Information Systems Research," *Journal of Management Information Systems* (24:3), pp. 45-77.
- Rougès, J.-F., and Montreuil, B. 2014. "Crowdsourcing Delivery: New Interconnected Business Models to Reinvent Delivery," *Proceedings of 1st International Physical Internet Conference*, Québec City, Canada.
- Sadilek, A., Krumm, J., and Horvitz, E. 2013. "Crowdphysics: Planned and Opportunistic Crowdsourcing for Physical Tasks," *Proceedings of the Seventh International AAAI Conference on Weblogs and Social Media*, Cambridge, Massachusetts, USA, pp. 536-545.
- Schöder, D., Ding, F., and Campos, J.K. 2016. "The Impact of E-Commerce Development on Urban Logistics Sustainability," *Open Journal of Social Sciences* (4), pp. 1-6.
- Slabinac, M. 2015. "Innovative Solutions for a "Last-Mile" Delivery - a European Experience," in: *15th International Scientific Conference Business Logistics in Modern Management*. Osijek, Croatia: pp. 111-129.