Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 52 (2021) 661–668

Transportation Research Procedia

www.elsevier.com/locate/procedia

23rd EURO Working Group on Transportation Meeting, EWGT 2020, 16-18 September 2020, Paphos, Cyprus

Potential of Shared Taxi Services in Rural Areas – A Case Study

Steffen Elting, Jan Fabian Ehmke*

University of Vienna, Department of Business Decisions and Analytics, Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria

Abstract

Due to modern communication and information technology, shared taxi services are on the rise. While most research and practical projects focus on services operating in densely populated areas, the advantages of individual, flexible and shared transportation services in sparsely populated urban areas have not been well explored yet. Using a constraint programming formulation, this work investigates request management policies to evaluate the economic potential of shared taxi services. Computational experiments simulate such services under various effective degrees of dynamism and demonstrate the impact of pre-bookings on their efficiency and service quality in rural areas.

Keywords: Degree of Dynamism, Ride Sharing, Dial-a-Ride Problem, Case Study

1. Introduction

In recent years, the protection of our environment has become a ubiquitous topic in politics, business, science and private spheres. With an increase in car ownership, extended car usage and the diminishing size of households, the occupancy rates of passenger vehicles have dropped from around 2.1 to around 1.6 between 1970 and 1990 (European Environment Agency, 2008). One promising field to organize transportation in a more environmental-friendly way could be the revived trend of ride sharing. Modern communication and computation technology have greatly enhanced this mode of transport by facilitating the collection of travel requests as well as a fast and appropriate matching of travelers. In this paper, we explore the performance of these services in sparsely populated rural areas considering that customers can book their seats one day in advance ("pre-bookings") or on short notice ("spontaneous bookings"). We are especially interested in the optimal distribution of pre-bookings and spontaneous bookings and in the effects of different time spans between request disclosure and desired pickup time for spontaneous bookings. Contrary to primary intuition, a high proportion of pre-bookings turned out to create significant issues in field trials of such services.

* Steffen Elting. Tel.: +43-1-4277-37915; fax: +43-1-4277-837915.
E-mail address: steffen.elting@univie.ac.at

10.1016/j.trpro.2021.01.079

Ride-Sharing applications are commonly modelled as Dial-a-Ride Problems (DARP). The first exact solution algorithm was proposed by Psaraftis (1980), who makes use of dynamic programming to iteratively solve new instances in a dynamic setting. Cordeau & Laporte (2003) provide a fundamental DARP definition and use a Tabu Search algorithm to solve the static DARP. Many other metaheuristics such as Genetic Algorithms (Jorgensen, Larsen, & Bergvinsdottir, 2007), Variable Neighborhood Search (Parragh, Doerner, & Hartl, 2010), Deterministic Annealing (Braekers, Caris, & Janssens, 2014) and hybrid methods (Berbeglia, Cordeau, & Laporte, 2012) have successfully been applied to the problem. Parallelization has also proven to be of great benefit (Attanasio, Cordeau, Ghiani, & Laporte, 2004). Molenbruch, Braekers, & Caris (2017) provide a more in-depth overview of the DARP literature. To incorporate newly arriving requests into evolving solutions, insertion techniques are commonly used (Haferkamp & Ehmke, 2019; Horn, 2002; Hosni, Naoum-Sawaya, & Artail, 2014; Madsen, Ravn, & Rygaard, 1995). Consistency with the already executed part of the route must be ensured when requests arrive during route execution. Berbeglia, Pesant, & Rousseau (2011) model this sub-problem as a dynamic DARP with fixed partial routes. Their Constraint Satisfaction formulation is also the basis for our model. Thus, we do not optimize any objective function but we only search for a feasible solution as this is sufficient to decide whether to accept or reject incoming customer requests.

Commonly, service inquiries in the related field of Vehicle Routing Problems (VRPs) are divided into static and dynamic requests. For static requests, details such as location, time window and number of passengers are known at the time of constructing route plans, whereas for dynamic requests, these details are only uncovered after the execution of the route plans has already started. The Degree of Dynamism (dod) is a measure introduced by Lund, Madsen, & Rygaard (1996) to quantify the ratio of dynamic requests as a fraction of all requests ranging from 0 (all information is accessible before planning) to 1 (all information is received during execution of the incumbent route plan). Larsen, Madsen, & Solomon (2002) point out that not only the number of dynamic requests is relevant, but also the available time to fulfil a request, named the system reaction time r. This measure is a sum of the advance notice time (adv) and the time window length (u) of a request and thus encodes the dynamism of individual requests. By incorporating this additional information, they define the Effective Degree of Dynamism ($edod$) as the average dynamism of all requests. Diana (2006) analyses the DARP under varying $edod$ in Los Angeles by changing the dod as well as r. However, the analysis only covers problems with 60 % dod or more and does not investigate the effects of varying compositions of the reaction time.

In this paper, we are especially interested in how varying aspects of the $edod$ impact DARP solutions. To this end, we will solve the Shared Taxi Problem (STP) under various dynamic conditions. In Chapter 2, we formally describe our problem, model the STP, and present our solution approach. The test instances and results are discussed in Chapter 3. Chapter 4 concludes the paper. The main highlights of our research are (1) a full analysis of the partially dynamic STP with the dod ranging from 0% to 100%, (2) an in-depth investigation of the effects of different compositions of r, and (3) experiments on a real-life ride-sharing application in a sparsely populated area.

2. The Shared Taxi Problem

This chapter provides a specification of the problem's characteristics as well as the corresponding mathematical constraint programming formulation. We also outline our solution approach.

2.1. Problem Description

The shared taxi service at hand works as follows. A service operator provides vehicles and drivers and collects requests from individual travelers that are dispersed around a limited geographical area. Based on every request's origin and destination as well as the desired pickup time, the service time and the number of passengers, the service provider matches vehicles and travelers and builds a route plan for each vehicle. Travelers will share (fractions of) their trip with other travelers. Requests arrive over the planning horizon of length T. Let t_0 and T be the start and end of the planning horizon, respectively, and t_s be the start of the transportation service with $t_0 \leq t_s \leq T$. We define $H = H_p \cup H_s$ such that a request $h \in H_p$ is called a pre-booking (request) if it is received at a time $t < t_s$. All $h \in H_s$ are considered spontaneous (requests), received at or after t_s. Spontaneous requests are, therefore, processed after the vehicles have already started their routes. Each request specifies a pickup time window of length u. Each customer must receive immediate feedback about acceptance, which is immutable once reported to the customer.

Note that in this problem, no request is truly static because even pre-bookings are known before t_s, but not necessarily at t_0 as would be the case for static requests. Consequently, an environment where all requests are disclosed before the start of the service t_s will be called pseudo-static. The term dod will then describe the ratio of spontaneous requests and thus $dod = 0.0$ refers to exactly the aforementioned pseudo-static case. Moreover, the notion of adv and r is only applied for spontaneous requests in the STP. Although, technically, both types of requests (pre-booking and spontaneous) have adv and r attributes, we assume that a decision maker can only have control over the spontaneous requests' attributes. In addition, adv and r of pre-bookings have little impact on the solution.

For the operator of a shared taxi business, it is relevant to know the impact of the $edod$, because the dod and the r can be influenced to some degree. To offer an on-demand transportation service, the operator may set an upper bound on the resources that can be reserved by pre-bookings. Moreover, control over the system reaction time can be achieved by setting lower bounds on u and adv of spontaneous bookings. E.g., in a mobile application, the operator allows customers to send only requests for which the resulting latest possible pickup time is sufficiently far in the future. If all necessary arrangements have been made such that the operator is in control of the dod, adv and u, a *request policy* can be determined. A request policy is a triple of values $\langle dod, adv, u\rangle$ that regulates the incoming stream of requests as well as their service reaction times.

2.2. Model

Based on Berbeglia et al. (2011), we model the Shared Taxi Problem (STP) following ideas of the Dial-a-Ride-Problem (DARP). We use a Constraint Satisfaction Problem formulation defined on a complete and directed graph $G = (V, A)$, where V is the set of vertices and A the set of edges. Let $K = 1, \ldots, m$ be the set of available vehicles and Q be the capacity of each of these vehicles and let n denote the total number of travel requests. As a Constraint Satisfaction model, the STP does not directly optimize any objective function. Constraint Satisfaction is sufficient to make acceptance or rejection decisions and a maximization of the number of overall accepted requests is pursued implicitly through the rolling horizon strategy explained below.

We set $V = S \cup R$ where S represents the set of vehicle depots with S^+ and S^- representing m start and end depots, respectively. Thus, for every vehicle $k \in K$, there exists a pair: $start(k) \in S^+$ (start depot) and $end(k) \in S^-$(end depot). R ($|R| = 2n$) is the set of vertices corresponding to traveler locations. The set R is divided into the set of pickup nodes R^+ and the set of drop-off nodes R^-. With every arc $(i, j) \in A$ is associated a non-negative travel cost $c_{i,j}$. The travel costs satisfy the triangle inequality. The vertices $i \in V$ have a time window $[e_i, l_i]$ with e_i and l_i defining the earliest and latest allowed arrival time, respectively. The non-negative service time D_i is required to fulfill the pickup or drop-off activity, and q_i is the number of travelers associated with location i, i.e. the number of travelers that will be picked up (positive value) or dropped off (negative value). The set of requests is represented by $H = 1, \ldots, n$. Every request $h \in H$ defines a pickup vertex $i^+ \in R^+$, a delivery vertex $i^- \in R^-$, and the number of travelers q. Based on the number of travelers, the following demand values are imposed on the pickup and delivery vertices of the request: $q_{i^+} = q$ and $q_{i^-} = -q$. Furthermore, every request comes with a maximum ride time L_h that sets an upper bound on the time that the associated travelers are willing to spend on the vehicle and with a time window of length u on the pickup vertex.

In our mathematical programming model, for each vertex $i \in V$ there are the following decision variables:

(1) $s[i] \in V$ indicates the immediate successor vertex of i

(2) $l[i] \in [0, Q]$ is the vehicle load right after visiting i

(3) $v[i] \in K$ shows the vehicle that i is assigned to

(4) $t[i] \in [e_i, l_i]$ is the arrival time at vertex i

Finding a solution is subject to the following formal constraints:

(5) $s[end(j)] = start(j)\ \forall\ j \in K$

(6) $v[end(j)] = v[start(j)] = j\ \forall\ j \in K$

(7) $allDifferent(s)$

(8) $v[i^+] = v[i^-]\ \forall\ h \in H$

(9) $v\big[s[i]\big] = v[i]\ \forall\ i \in V$

(10) $t[i^+] \leq t[i^-]\ \forall\ h \in H$

(11) $t[i] \leq t\big[s[i]\big] - c_{i,s[i]} - D_i\ \forall\ i \in V^+ \cup R$

(12) $e_i \geq t[i] \geq l_i\ \forall\ i \in V$

(13) $l\big[s[i]\big] = l[i] + q_{s[i]}\ \forall\ i \in R$

(14) $l[i] \leq Q\ \forall\ i \in R$

(15) $l[start(k)] = 0\ \forall\ k \in K$

(16) $t[i^-] - (t[i^+] + D_{i^+}) \leq L_h\ \forall\ h \in H$

With this model, the STP consists of finding m vehicle routes while satisfying the following conditions: Constraints (5) and (6) ensure closed tours for all vehicles. The $allDifferent$ constraint (7) guarantees that there are no two vertices with the same successor. The origin and destination vertices of a request h must be served by the same vehicle (8) and also any chain of succeeding locations must have the same vehicle assignment (9). The correct precedence of pickup and delivery locations is ensured in (10), and (11) enforces the consideration of travel and service times. In (12), the time window constraints ensure that service at each vertex starts in the interval $[e_i, l_i]$. The vehicle capacity is adjusted at each location based on boarding and alighting of passengers (13) and it cannot exceed Q at any point in time (14). In the beginning of their tour the vehicles are empty (15). Finally, constraints (16) ensure the maximum ride time limit for each request such that the time difference between the pickup and the delivery vertex of each request h does not exceed L_h.

2.3. Solution Procedure

We apply a rolling horizon strategy that triggers an insertion algorithm whenever a new request is uncovered. This means that for every incoming request at time t, a corresponding new STP instance containing only the requests until that time is created and solved. To maintain consistency between two successive solutions, the parts of a solution that have already been realized before time t need to be preserved in the next solution (Berbeglia et al., 2012). This is done in a pre-processing step that adds the necessary constraints to the current model to copy e.g. the affected vehicle assignments of the previous solution. Finding a solution for a given STP instance corresponds to proving the feasibility of a Dial-a-Ride Problem with partially fixed routes. This task is NP-complete (Berbeglia, Pesant, & Rousseau, 2012). Hence, we apply a simple pairwise insertion heuristic to find suitable positions for the new request's pickup and drop-off points. The heuristic checks all possible insertions into a vehicle's route plan before moving on to check insertion for the next vehicle. If a feasible insertion is found, it is applied and the next request is processed.

3. Computational Study

To assess the efficiency of the STP under varying $edod$, we repeatedly solve it through an implementation in Python 3.7 using Google's open-source library OR-Tools 7.4. In this chapter, we will describe the two sets of problem instances and present our computational results.

3.1. Test Instances and Request Policies

To select locations for the depot and the requests, a map of 16 cities was constructed based on the Oberharz region in Germany, which was the area of operation of the EcoBus pilot project from August 2018 to February 2019 (www.ecobus.info). The travel times between those cities were computed with a constant speed of 50 km/h and haversine distances. First, a single depot location is drawn randomly. Second, to generate requests, origin and destination cities are drawn randomly with sampling probabilities proportional to each city's population size. For each request $h \in H$, two cities (pickup and drop-off) are drawn without replacement. Replacement is prohibited because transportation within the same city is restricted. This is a generalized constraint from the original project (Herminghaus, 2019). All requests comprise a single traveler, i.e. $q = 1\ \forall\ h \in H$, and we create 250 requests for each experiment. In all cases, the planning horizon T has a total length of one day and 22 hours. Following the Oberharz field trial, the actual transportation takes place on the second day, at which service starts at 06:00 in the morning, and drivers must return to the depot at 22:00.

Parameter Values	dod	adv	u
for request policies of set $S1$	$\{0.0, 0.25, 0.5, 0.75, 1.0\}$	$\{0, 5, 10, 20, 40\}$	$\{600\}$
for request policies of set $S2$	$\{1.0\}$	$\{0, 5, 10, 20, 40\}$	$\{0, 5, 10, 20, 40\}$

Table 1: All tested parameter values for sets S1 and S2.

To analyze the impact of the ratio of pre-bookings and spontaneous requests, each instance has exactly $n - (n \cdot dod)$ pre-bookings and $n \cdot dod$ spontaneous requests. The earliest pickup times of all requests are uniformly distributed over the service period, i.e. $e_{i^+} \sim U(t_s, T)\ \forall\ h \in H$. Moreover, a pre-booking has a uniform chance of arriving at any second between the beginning of day one and the start of service on day two, i.e. $t \sim U(t_0, t_s)\ \forall\ h \in H_p$. The disclosure time t of real-time requests is computed by subtracting the advance notice time (as defined by the request policy) from the sampled earliest pickup time. Like the EcoBus project, fleet size and capacity of the vehicles are fixed for all instances such that $m = 7$ and $Q = 8$. These values have proven to be appropriate in preliminary tests.

Customer flexibility is reflected in the time window length u. The length of this time window has a direct impact on the system reaction time. The maximum ride time is set to 1.5 times the travel duration of a direct connection for all customers, i.e. $L_h = 1.5 \cdot c_{i^+,i^-}\ \forall\ h \in H$. This value is assumed to be a realistic representation of passengers' ride time flexibility. Furthermore, the service time per passenger is consistently set to $D_i = 5$ minutes. This is for the boarding and alighting times at each stop. 100 instances are generated for each policy to mitigate the effect of the sampling variability.

The request policies tested in the following section can be divided into two sets: $S1$ examines the sensitivity of the STP's efficiency to varying dod and varying system reaction times. With $S2$, the impact of the two components of r, namely adv and u, are analyzed. For $S1$, request policies are created that vary in the dod and r for the spontaneous requests. In general, r can be modified by shortening or extending either adv or u or both. For now, only adv will be altered and u will remain constant. $S2$ will then explore in more detail the impact of different ratios of adv and u to see to which extent the composition of r affects the efficiency of the STP. $S1$ contains 20 unique request policies that can be created by combining the values from Table 1. Note that with $dod = 0.0$, only one unique policy can be generated, as adv and u are irrelevant for cases without dynamic requests. This particular pseudo-static policy $\langle 0.0, -, - \rangle$ will be the baseline for the upcoming evaluations. With $S2$, the dod is fixed, and only the other two parameter values are changed, yielding 25 different combinations. A subset of policies for which all but one parameter values are equal is referred to as an experiment (E.g. $\bigcup_{dod} \langle dod, 0, 10 \rangle$).

3.2. Results

In the following, we analyze the overall efficiency of the STP by measuring the average rejection rate, which is the fraction of requests that got turned down out of all requests. We will also investigate how frequently ride-sharing occurs by looking at the average number of passengers in a vehicle and see its impact on the travel time of the customers. Evaluating the rejection rate's sensitivity to changes in adv and u concludes the analysis.

3.2.1. Rejection Rate

Figure 1 shows the average rejection rate that results from varying dod. Series A, B, C, D, E represent different settings for the advance notice time in minutes, which correspond to 0, 5, 10, 20 and 40 minutes, respectively. When all the requests are pre-bookings, around 29% out of all the 250 customers cannot be served. In experiment A, where customers demand immediate service, the number of rejections monotonically grows with increasing dynamism. This experiment reaches a rejection rate of about 57% once all customer queries are spontaneous requests. This is a foreseeable result: as the dod increases, the service provider is more often confronted with little flexibility because parts of the route have already been executed and the time to get to the customer before his or her time window closes is just too short to react. With increasing advance notice time in experiments B, C, D and E, the curve of the rejection rate flattens and eventually slopes downwards. For instance, the highest rejection rate for trace C is observed with half of the customers sending pre-booking requests. With 20 (D) and 40 (E) minutes advance notice time, from 0% to 50% dod, the rejection rate remains relatively invariable to the change in dynamism. However, with the dod greater than 50%, the solution approach achieves better results than in the pseudo-static scenario.

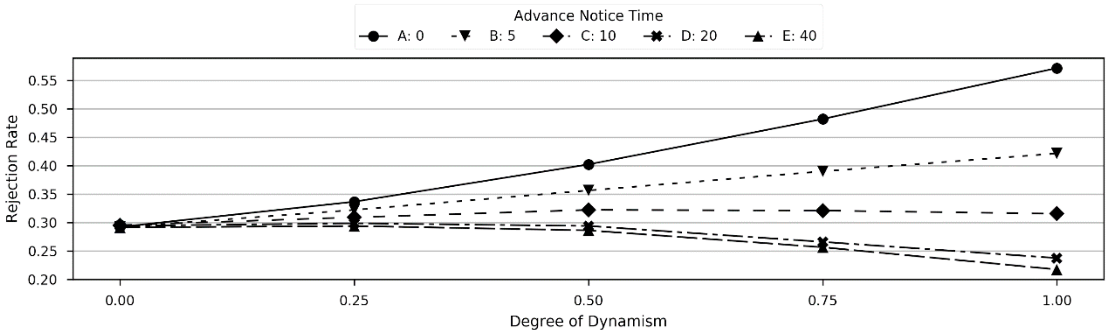

Figure 1: Average rejection rate as a function of the Degree of Dynamism

When all requests are spontaneous ($dod = 100$), an adv value of 40 minutes (E) yields an improvement of about 8% over the baseline. This result is surprising, as theoretically, the pseudo-static case offers greater planning flexibility. It demonstrates that sufficient advance notice time can compensate for the negative impact of a higher dod. Note that this does not hold vice versa: A lower dod is beneficial only if the policy defines a short advance notice time for spontaneous requests. Indeed, with high enough adv times, lower $dods$ are detrimental for efficiency. Moreover, increasingly long lead times deliver diminishing returns as the benefits they bring do not scale proportionally with the advance notice time. Note that the adv values were doubled for each experiment B-E. The resulting gain in efficiency, on the other hand, decreases. The explanation is that building the routes in the sequential order of desired pickup time is advantageous for the insertion algorithm. Recall that pre-bookings' earliest pickup times are not ordered sequentially as opposed to the desired collection times of spontaneous requests. For the former, the difference between t and e_{i^+} is random with relatively large values, whereas it is stable and takes smaller values for the latter ($e_{i^+} - t = adv \; \forall \; h \in H_s$). Because the rolling horizon insertion procedure does not try to optimize any request-to-vehicle assignment decisions that it has made in the past, the assignments that turn out to be detrimental in the context of additional information cannot be revised. If the desired pickup times are not random but ordered (due to the constant advance notice time) as in a scenario without any pre-bookings ($dod = 1.0$), then resources are only blocked for the immediate future, which ensures a more efficient vehicle assignment. The route plan is built step by step; hence, the chances of rejection are similar for every ride seeker in that scenario.

3.2.2. Occupancy Rates and Excess Ride Times

A key concept of our Shared Taxi Service is the ability for the service provider to pool customers that have similar itineraries and have them drive some fraction of their route together. But as it has been discussed in the introduction, the rural setting of the regarded case study poses a big challenge with respect to finding such matches. This is also reflected in the overall rather low absolute occupancy rates as seen in our experiments (Figure 2). The occupancy rate

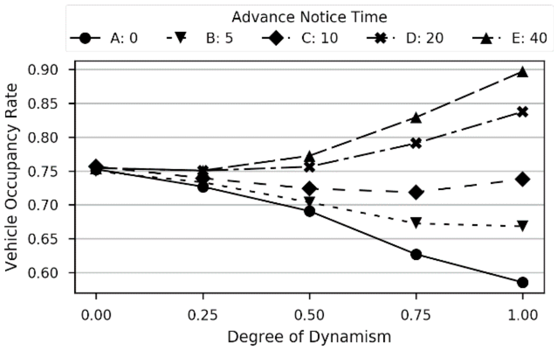

Figure 2: Average number of passengers on a taxi per leg

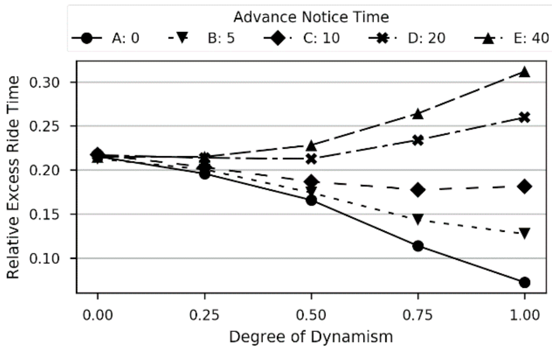

Figure 3: Average excess ride time for any customer relative to the direct ride time of the corresponding request.

r	(a) u only	(b) mixed	(c) adv only
10 Minutes	57.84 %	57.06 %	59.41 %
20 Minutes	30.57 %	30.81 %	40.21 %
40 Minutes	16.28 %	17.04 %	34.18 %

Table 2: The rejection rates under different compositions of the service reaction time illustrates the sensitivity to changes in u and adv. In the mixed case, the total reaction time is made up of equal parts of u and adv.

shows the average number of passengers on a taxi per leg, where a leg is a route section between two locations. Even a policy of $\langle 1.0, 40, 10\rangle$ can only pool 0.9 passengers on one leg, i.e. on average, there are 10% empty legs. These represent legs that a vehicle covers to get to the next pickup location and when there is currently no passenger in the vehicle. Excluding empty legs, the same policy yields an occupancy of 1.3. The figure also gives evidence that high occupancy rates require long adv times whereas, with very short adv, fewer taxi rides include a shared leg.

Higher vehicle occupancy implies a detour for at least one of the passengers in most cases. Thus, a higher occupancy rate goes along with increased travel times and consequently, the plots for the average occupancy rates (Figure 2) and the average excess ride time (Figure 3) have a very similar appearance. Figure 3 shows the average excess ride time per request proportional to the corresponding duration of the direct route. A minimum service level is guaranteed by the maximum ride time constraints (15) such that no customer will travel more than 1.5 times the direct connection time. The maximum ride time is essential for the decision maker's ability to pool requests. However, due to the simplicity of the insertion scheme, the full potential cannot be exploited. The baseline policy scores an average excess ride time of about 22%. With $dod > 0$, the respective adv value determines whether enough reaction time is provided to take advantage of the immanent benefits of the ride time flexibility. Experiments A-C perform worse than the baseline, but with lead times of 20 minutes or more (D and E), higher relative excess travel times of up to 31% can be utilized to accommodate more requests in the route plan. Summarizing, again, we see that a high number of pre-bookings is not always better. Instead, the interplay of r and dod must be considered carefully to exploit the excess ride time to maximize vehicle occupancy rates and consequently the number of accepted requests.

3.2.3. Sensitivity to the Composition of Service Reaction Times

For $S1$, changes in r were induced only by changes in adv. In this section, the composition of the system reaction time is examined more closely as it can be suspected that its two components adv and u have a different impact on the efficiency. Therefore, in $S2$, we set $dod = 1.0$ and iterate through different ratios of adv to u.

About 98% of customers are rejected when both dod and adv are set to 0. In this scenario, request acceptance can only happen by chance, i.e. only if a vehicle happens to be at a requested pickup location at the right time. Depending on the request policy, any non-zero system reaction time is composed in one of three ways: (a) time windows only (i.e. $r = u$), (b) a mixture of time windows and advance notice time (i.e. $r = u + adv$) or (c) only advance notice time (i.e. $r = adv$). Table 2 compares system reaction times of equal length but varying composition. With $r = 20$, the adv only scenario is almost 10% worse in terms of rejected customers than both other compositions. And when $r = 40$, roughly twice as many customers are rejected in case (c) compared to (a) and (b). It becomes apparent that u has a stronger impact on the rejection rate than adv. Long time windows contradict the vision of a flexible, spontaneous public transport service but our results show that the immanent benefits of longer time windows can already be harnessed when they make up only half of the total extension of the reaction time.

4. Conclusion

The primary aim of this paper was to analyze the STP and to quantify its performance in terms of the number of served passengers under varying dynamic settings. A combinatorial optimization model was set out to represent a Shared Taxi Service in rural areas of low demand density and long travel distances. The results of the computational study lead to the following findings and recommendations for Shared Taxi Services. An intuitive presumption that more spontaneous bookings generally bring about a greater rejection rate only holds in highly dynamic cases where spontaneous bookings have a very low system reaction time. By imposing lower bounds on the reaction time of these requests, we were able to consistently produce solutions that accommodate more customers than in a setting of only pre-bookings. This is due to the more ordered pickup times of spontaneous bookings that allow a better vehicle assignment for the insertion algorithm. A sensitivity analysis showed that out of the two components (adv and u) of the system reaction time, the time window length is the more decisive factor for the service's efficiency. In conclusion, a Shared Taxi Service operator must be aware of the significant effects that different components of the $edod$ have on the performance of the system and should aim at tuning them continuously.

Future research can further explore the behavior of the STP under varying dynamism by using more advanced insertion heuristics or intermediate optimization steps that exploit the time in between feasibility checks to optimize an incumbent solution. Sensible and common extensions of the DARP also include the presence of outbound requests (which provide a latest possible drop-off time rather than an earliest pickup time), time-dependent demand fluctuations and travel times (e.g. rush hours in the morning and evening) as well as an integration of the service into an existing public transit network of buses and trains.

References

Attanasio, A., Cordeau, J.-F., Ghiani, G., & Laporte, G. (2004). Parallel Tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*, *30*, 377–387.

Berbeglia, G., Cordeau, J.-F., & Laporte, G. (2012). A Hybrid Tabu Search and Constraint Programming Algorithm for the Dynamic Dial-a-Ride Problem. *INFORMS Journal on Computing*, *24*, 343–355.

Berbeglia, G., Pesant, G., & Rousseau, L.-M. (2011). Checking the Feasibility of Dial-a-Ride Instances Using Constraint Programming. *Transportation Science*, *45*, 399–412.

Berbeglia, G., Pesant, G., & Rousseau, L.-M. (2012). Feasibility of the Pickup and Delivery Problem with Fixed Partial Routes: A Complexity Analysis. *Transportation Science*, *46*, 359–373.

Braekers, K., Caris, A., & Janssens, G. K. (2014). Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B: Methodological*, *67*, 166–186.

Cordeau, J.-F., & Laporte, G. (2003). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, *37*, 579–594.

Diana, M. (2006). The importance of information flows temporal attributes for the efficient scheduling of dynamic demand responsive transport services. *Journal of Advanced Transportation*, *40*, 23–46.

European Environment Agency (2008). *Are we moving in the right direction?: Indicators on transport and environment integration in the EU.* TERM, 1.2000. Luxembourg: Office for Official Publ. of the Europ. Communities.

Haferkamp, J., & Ehmke, J. F. (2019). An Efficient Insertion Heuristic for On-Demand Ridesharing Services. *Transportation Research Procedia*.

Herminghaus, S. (2019). Homepage - EcoBus Projekt. https://www.ecobus.jetzt/home.html. Accessed 10.12.2019.

Horn, M. E.T. (2002). Fleet scheduling and dispatching for demand-responsive passenger services. *Transportation Research Part C: Emerging Technologies*, *10*, 35–63.

Hosni, H., Naoum-Sawaya, J., & Artail, H. (2014). The shared-taxi problem: Formulation and solution methods. *Transportation Research Part B: Methodological*, *70*, 303–318.

Jorgensen, R. M., Larsen, J., & Bergvinsdottir, K. B. (2007). Solving the Dial-a-Ride problem using genetic algorithms. *Journal of the Operational Research Society*, *58*, 1321–1331.

Larsen, A., Madsen, O., & Solomon, M. (2002). Partially dynamic vehicle routing—models and algorithms. *Journal of the Operational Research Society*, *53*, 637–646.

Lund, K., Madsen, O. B. G., & Rygaard, J. M. (1996). *Vehicle routing problems with varying degrees of dynamism.* Technical report IMM-REP, 1996,1. Lyngby: Technical Univ.

Madsen, O. B. G., Ravn, H. F., & Rygaard, J. M. (1995). A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research*, *60*, 193–208.

Molenbruch, Y., Braekers, K., & Caris, A. (2017). Typology and literature review for dial-a-ride problems. *Annals of Operations Research*, *259*, 295–325.

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2010). Variable neighborhood search for the dial-a-ride problem. *Computers & Operations Research*, *37*, 1129–1138.

Psaraftis, H. N. (1980). A Dynamic Programming Solution to the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem. *Transportation Science*, *14*, 130–154.