Spatial Information in Offline Approximate Dynamic Programming for Dynamic Vehicle Routing with Stochastic Requests

Ansmann, Artur, TU Braunschweig, a.ansmann@tu-braunschweig.de
Ulmer, Marlin W., TU Braunschweig, m.ulmer@tu-braunschweig.de
Mattfeld, Dirk C., TU Braunschweig, d.mattfeld@tu-braunschweig.de

In this paper, we study the Dynamic Vehicle Routing Problem with Stochastic Customers (DVRPSC), a common problem setting for Courier, Express and Parcel service providers. We focus on the case in which a dispatcher must decide which dynamically occurring customer requests should be confirmed and how to integrate these into the existing tour of a vehicle. The vehicle must serve all confirmed requests and return to its depot within a given time limit. In order to confirm a maximum number of requests, anticipation of future requests for a current state’s decision is necessary. To allow real-time control, the required calculations need to be conducted offline, often by means of Value Function Approximation (VFA). The calculation’s outcome for every state is then stored in aggregated form and can be accessed efficiently in the online execution. Current VFAs for the DVRPSC are not able to integrate any spatial information in their aggregation but solely draw on temporal state attributes. Therefore, in problem settings expressing a characteristic spatial distribution of requests, these are not able to anticipate sufficiently. In this paper, we propose Anticipatory Time Budgeting with Spatial Information (ATBS) to close this gap. We compare ATBS to a state-of-the-art VFA of the literature on a set of benchmark instances with varying spatial distribution parameters. Results show that the integration of spatial information is highly beneficial.

Keywords: Dynamic Vehicle Routing, Offline Approximate Dynamic Programming, Anticipation, Spatial Information

1 Introduction

Courier, Express and Parcel (CEP) services gain in importance due to the continuing urbanization and growing number of digital buyers (eMarketer, 2016). At the same time, customers expect fast and cheap transportation of parcels, which imposes challenges faced by the CEP service provider especially for the “first mile” and “last mile”. We consider a problem setting regarding the “first mile” for a parcel pickup service, in which customers may request to pick up their parcels in advance or the same day. A vehicle is then dispatched to pick up the parcels and to transport these to a depot for further processing (e.g. for long haul transportation).
Typically, a set of a priori known customers is given by requests that were issued in advance. After the vehicle left the depot, new customer requests (corresponding to same day pick up requests) occur in the area to which the vehicle is assigned. These dynamic requests should be integrated into the tour of the vehicle. Usually, the parcels can be picked up only during a fixed service time period and thus we assume that this represents the major resource constraint. Consequently, the tour of the vehicle must be adapted such that a maximum number of dynamic requests can be served within that timeframe. Therefore, the dispatcher not only decides about the tour order, but also whether a dynamic request should be served by the vehicle on-tour or not.

The resulting decision problem is a variant of the Dynamic Vehicle Routing Problem with Stochastic Customers (DVRPSC, Ritzinger et al., 2016). Other variants of this problem are often considered in the literature (see Ulmer et al., 2016 for a comprehensive overview). Recently, approaches draw on stochastic information to anticipate future requests. To this end, the increasing volumes of transactional data gathered and stored by logistical service providers can be used (Gendreau et al., 2016). For the DVRPSC, the stochastic information comprises locations and times of future requests.

For anticipatory decision policies, stochastic information as well as potential future decision making must be incorporated into the current decision, which requires significant computational efforts. However, new requests require fast responses and thus computation time for calculation is highly limited. To overcome the time limitations, decision policies can be computed a priori using methods such as VFA from the field of Approximate Dynamic Programming (Powell, 2011). These methods conduct extensive offline simulations and store the simulation’s outcomes for decision states in an aggregated form.

Until now, these aggregations neglect spatial information of the decision states. The state-of-the-art VFA for the DVRPSC by (Ulmer et al., 2016) aggregates states using temporal information on point of time and free time budget. In many cases, considering spatial information like vehicle’s or customers’ locations may improve the quality of decision making, especially if the customers are heterogeneously distributed in the service area. We therefore propose Anticipatory Time Budgeting with Spatial Information (ATBS) to close this gap augmenting the VFA by (Ulmer et al., 2016) with spatial information. To test our assumption, we conduct computational studies on a set of benchmark instances varying in the heterogeneity of the customer locations. The results show that the proposed approach for the integration of spatial information is highly beneficial, particularly for instance settings with high spatial heterogeneity.

The remainder of this paper is organized as follows. First, the DVRPSC and its Markov Decision Process formulation are presented in Section 2. In Section 3, we describe ATBS. We evaluate our approach using a computational study in Section 4 and conclude the paper in Section 5.
2 Problem Definition

In this section, we define the DVRPSC and formulate the problem as a Markov Decision Process (MDP, Puterman, 2014).

2.1 Dynamic Vehicle Routing with Stochastic Customers

A single uncapacitated vehicle must visit a set of confirmed customers \( C_{\text{conf}} \subseteq C = \{c_1, ..., c_n\} \) and return to a depot \( c_0 \) within a given time frame \( T = [0, 1, ..., t_{\text{max}}] \). The vehicle starts and ends its tour at a depot that is located at a fixed point in service area \( A \). It travels with constant velocity between customers and the depot. Also, the vehicle is allowed to wait at its current position. Service times at customers are constant.

Each customer \( c_i \) has a request time \( t_i \in T \) and a location \( l_i \in A \), with \( i \in I = \mathbb{N} \). The number of customer requests \( n \), the request times \( t_i \) and the locations \( l_i \) follow predefined probability distributions. A subset of customer requests (called early request customers) is known before the vehicle leaves the depot \( C_{\text{early}} = \{c_i \in C: t_i = 0\} \). These requests are already confirmed and must be visited. The order of visits is given by a tentative tour.

The other set of customers (called late request customers) represents requests that occur dynamically during the shift, i.e. \( C_{\text{late}} = \{c_i \in C: t_i > 0\} \), and the complete set of late requests is only known a posteriori. The proportion of \( C_{\text{late}} \) to \( C_{\text{early}} \) is specified by the degree of dynamism (Larsen et al., 2002).

At \( k_j \in K \subseteq T \) time points of the time period, late customer requests occur. At these points, it must be decided whether to confirm or reject a request and how to adapt the tentative tour in order to include a confirmed request. The objective is to maximize the number of confirmed late request customers.

2.2 Markov Decision Process Formulation

We model the DVRSC as a MDP in which a decision \( x_k \) from a set of feasible decisions \( X_k^{\text{feas}} \) at successive decision points \( k \in K \) must be selected. A decision point occurs when a new customer request occurs. The decision consists of a confirmation action and a movement action. The confirmation action involves the confirmation or rejection of the request. The movement action comprises the selection of the next customer \( c_k^{\text{next}} \) that should be visited by the vehicle after reaching its current destination. Thus, the vehicle does not change its route while it is traveling between two consecutive nodes. The next customer \( c_k^{\text{next}} \) is selected from the set of confirmed, but not visited customers \( C_k^{\text{conf}} \).

The customer to which the vehicle is currently traveling is considered as already visited. A decision is considered as feasible, if the duration of the tour induced by the decision plus the time already elapsed until that decision point is within \( t_{\text{max}} \).

The system state at a decision point \( k \) is described by a tuple \( s_k = (l_k, t_k, c_k^{\text{conf}}, c_k^{\text{req}}) \) containing the position of the vehicle \( l_k \), the elapsed time \( t_k \), the positions of confirmed customer requests \( C_k^{\text{conf}} \) and the position of a new customer request \( c_k^{\text{req}} \) that occurred
after the last decision point. The selection of a decision \( x_k \) in state \( s_k \) is associated with an immediate reward
\[
R_k(s_k, x_k) = \begin{cases} 
1, \text{if the request is confirmed} \\
0, \text{if the request is rejected}
\end{cases}
\]
and leads to a post decision state (PDS) \( s_k^\pi = (l_k, t_k, C_k^{conf}) \). Like a state \( s_k \), the PDS contains the position of the vehicle and the elapsed time since it left the depot. Other than a state, the PDS contains only the set of confirmed customers, which includes the new customer request if confirmed. Executing the selected decision leads to a stochastic transition to the next state \( s_{k+1} \). The process repeats until all confirmed customers are served and the vehicle has returned to the depot, i.e. \( s_K = (c_0, t_k, \emptyset, \emptyset) \).

The objective in the context of MDP is to find an optimal decision policy \( \pi^* \in \Pi \) maximizing the expected reward that is received from an initial state \( s_0 \) until a terminal decision point \( K \) is reached, i.e. \( \pi^* = \arg \max_{\pi \in \Pi} \mathbb{E}[\sum_{k=0}^{K} R_k(s_k, x_k^\pi(s_k))|s_0] \). In our formulation, a policy \( \pi \) is a deterministic mapping from a state to a decision \( x = \pi(s_k) \).

### 3 Anticipatory Time Budgeting with Spatial Information

In this section, we describe how ATBS generates decision policies drawing on VFA. We then present the state space aggregation applied in ATBS and how routing is conducted using spatial information.

#### 3.1 Value Function Approximation for the DVRPSC

An optimal decision policy selects the action that maximizes the sum of immediate reward and expected future rewards in every decision point, as stated by the Bellman equation (Bellman, 1957):
\[
x_k^\pi(s_k) = \arg \max_{x \in X(s_k)} (R_k(s_k, x) + \mathbb{E}[\sum_{j=k+1}^{K} R_j(s_j, x_j^\pi(s_j))|s_k])
\]
However, solving this equation is computationally intractable for real-world sized DVRPSC instances (Powell, 2011). Instead, the expected reward is not calculated exactly, but approximately via VFA. To this end, the expectation term is expressed as a function of the PDS variable:
\[
V(s_k^\pi) = \mathbb{E}_{x \in X(s_{k+1})} \left[ \max_{x \in X(s_{k+1})} (R_{k+1}(s_{k+1}, x) + V(s_{k+1}^\pi)) \right]
\]
To approximate function \( V(s_k^\pi) \) with sufficient accuracy, an aggregation of PDS to lower dimensional vectors \( v \) with suitable parameters describing a PDS is necessary (Ulmer et al., 2016). The function is represented by a lookup table, which contains the approximated expected reward for a given \( v \). The values of the lookup table are approximated offline in a learning phase. The learning phase starts with a lookup table in which a constant default value is assigned to each vector. Next, a large number of \( n \) Monte-Carlo simulations is conducted in which feasible decisions are selected partly according to the Bellman equation and partly randomly in order to explore the state space.

After a simulation run, the default values are replaced with the realized rewards for each \( v \) that occurred in the simulation run. If the same \( v \) occurs in the next simulation run...
run, then the new values are used for decision making and updated using a smoothing parameter at the end of the run. This process is repeated until \( n \) simulation runs have been conducted. The lookup table is stored and can be used for decision making in the online phase. If a vector \( v \) occurs for which only a default value is available (i.e. this \( v \) did not occur during the learning phase), then a value with zero expected rewards is returned.

3.2 Aggregation of Post Decision States

Equally to the approach proposed by (Ulmer et al., 2016), ATBS uses two temporal parameters, time \( t_k \) and time-budget \( b_k \) to aggregate a PDS. Parameter \( t_k \) is directly adopted from the according PDS. Parameter \( b_k \) is calculated by subtracting the sum of the time \( t_k \) and the duration of the tentative tour from \( t_{max} \). In addition to these two parameters, another parameter that represents spatial information is included in the aggregated vector space.

Ideally, this parameter would describe all coordinates that are available in the PDS \( S_k^X \), but this would result in an intractably large vector space. Even considering detailed location information of just one element like the current position of the vehicle or next customer can easily lead to an intractable vector space in larger instances. To obtain a tractable vector space, a coarser representation of a location is necessary. To this end, the service area \( A \) can be partitioned into a grid of non-overlapping subareas or customer requests can be grouped by spatial proximity into cluster shaped subareas. Each subarea is represented by an integer value. This value is used as a proxy for the part of the service area that is covered by a particular subarea, similar to a postal code.

For our approach, we use the cluster assignment \( h_k \) of \( c^\text{next}_k \) as the attribute for representing spatial information. Therefore, each PDS is aggregated to a three-dimensional vector space \( v = (t_k, b_k, h_k) \).

3.3 Routing

We further exploit the spatial information for the selection of the movement and the confirmation action as follows. First, ATBS generates a set of candidate customers \( C^\text{cand}_k \) for \( c^\text{next}_k \) in which each candidate \( c^\text{cand}_k \) is required to be from another cluster, i.e. a candidate for each cluster. From each cluster, the customer with the shortest travel time from the current destination customer is selected as \( c^\text{cand}_k \). The rest of the tour is calculated for each \( c^\text{cand}_k \) using Cheapest Insertion (CI, Rosenkrantz et al., 1974) with a penalty on insertion costs for customers that have a different cluster assignment than the \( c^\text{cand}_k \). In this way, we force the construction of candidate tours each favoring a different cluster.

The idea is to evaluate whether leaving the current cluster and changing to another cluster is worthwhile or not, even if the next customer is not the cheapest to insert at the current state. For evaluation, ATBS compares the value of \( v \) of each candidate tour and selects the decision that maximizes overall rewards. Since changing the cluster leads to a lower time budget and thus to a lower value, the best candidate tour would always
correspond to the tour induced by CI. However, a lower time budget does not imply lower values when incorporating spatial information, since each cluster may contain a different value for expected rewards. Therefore, spatial information is necessary in order to evaluate decisions regarding a cluster change.

4 Computational Study

In this section, we first present the details of the instances that were used for the computational study. Furthermore, we compare ATBS with ATB, proposed by Ulmer et al. (2016), and a myopic approach.

4.1 Problem Instances

For the computational evaluation, we sample instance realizations with the following characteristics. The vehicle travels with a constant velocity of 25 km/h within a service area of size 20 km x 20 km. The depot is located in the middle of the service area at coordinates (10 km, 10 km). The time frame is set to 360 minutes. Customer request times are generated by a Poisson process and arrive over the entire time frame. The expected total number of requests is 100 and the degree of dynamism is 0.75, i.e. 75 percent of the requests are expected to be late customer requests. The request locations are following a two-dimensional normal distribution with standard deviation of 2 km around abstract cluster centers forming customer clusters. The first cluster center is located at (5 km, 5 km) and the second cluster center is located at (15 km, 15 km).

In order to study the behavior of ATBS, we vary the probability for the occurrence of a request in a certain cluster via coefficient $\alpha$. We evaluate three different values of $\alpha$ describing three exemplary instance configurations. For the first configuration, we set $\alpha = 0.5$ meaning that customers are equally likely to appear in both clusters (“Homogeneous Clusters”). For the second configuration, we set $\alpha = 0.7$ resulting in 70 percent of requests coming from the first cluster (“Medium Heterogeneous Clusters”). In the third configuration, we set $\alpha = 0.9$ (“Highly Heterogeneous Clusters”). We expect that ATBS’s solution quality increases with an increasing degree of heterogeneity of the clusters, assuming that spatial information becomes more important with increasing $\alpha$.

4.2 Results and Analysis

We examine the solution quality in terms of the average ratio of confirmed late customer requests to total late customer requests for ATBS and ATB. For both approaches, 1 million simulation runs are conducted for every instance configuration in the offline learning phase. For the evaluation, we use 10,000 instances for each instance setting. As a benchmark, we also include results for a myopic policy that confirms requests as long as the resulting tour remains feasible. Table 1 contains the average ratio of late customer confirmations (in percent) and the standard deviation for each value of $\alpha$. 
Table 1: Average Served Late Requests and Standard Deviation (Percent)

<table>
<thead>
<tr>
<th></th>
<th>Homogeneous Clusters ((\alpha = 0.5))</th>
<th>Medium Heterogeneous Clusters ((\alpha = 0.7))</th>
<th>Highly Heterogeneous Clusters ((\alpha = 0.9))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Myopic</td>
<td>63.01 (±7.21)</td>
<td>64.11 (±6.86)</td>
<td>67.77 (±6.40)</td>
</tr>
<tr>
<td>ATB</td>
<td>64.83 (±5.95)</td>
<td>64.45 (±6.23)</td>
<td>68.00 (±8.09)</td>
</tr>
<tr>
<td>ATBS</td>
<td>64.42 (±5.70)</td>
<td>68.68 (±6.29)</td>
<td>75.59 (±5.98)</td>
</tr>
</tbody>
</table>

ATBS outperforms ATB and the myopic approach in the heterogeneous instance settings significantly. As expected, the performance gap increases with increasing heterogeneity of clusters.

The points of time when the vehicle changes between the clusters can be used to explain why ATBS performs better than ATB (and the myopic approach) with increasing heterogeneity of clusters. Similar points of time for cluster changes indicate that the routing is similar across the test instances leading to more “robust” solutions. Figure 1 depicts the frequency of time points in which the first and second cluster change occurred for ATB and ATBS in the homogeneous and highly heterogeneous cluster setting. We observe that ATB shows two distinct cluster changes for the homogeneous distribution (Figure 1a) but a high variance in changes for the heterogeneous distribution (Figure 1b) while ATBS shows an opposed behavior.

This behavior is connected to the solution quality keeping in mind that ATB performs well for homogeneous and poorly for heterogeneous distributions.

Figure 1: Cluster Changes per Point of Time for ATB and ATBS

Spatial information allows ATBS to differentiate between the clusters. For homogeneous clusters, this information is not valuable and even impedes the approximation process. For heterogeneous clusters, the spatial information becomes
important and ATBS provides high quality solutions while the solution quality of ATB declines.

5 Conclusion and Future Research

The DVRPSC is an important problem in urban transportation. CEP service providers must operate efficiently to stay competitive. This requires the planning of efficient and flexible tours. Anticipation of future customer requests enables deriving such plans. To this end, offline approximate dynamic programming approaches can be used.

Offline approaches provide fast and high quality solutions during the execution phase by extensive precomputing of decision policies in a learning phase. Current offline approximate dynamic programming approaches for the DVRPSC use temporal features in order to approximate the value function in the learning phase.

In this paper, we propose an offline approximate dynamic programming approach that considers spatial information. Therefore, the service area is partitioned into subareas represented by integer values similar to postal codes. This information is then utilized to approximate the value function in the learning phase. The results obtained from simulation study indicate that explicit consideration of spatial information is beneficial, especially in highly heterogeneous cluster settings.

Future research might include the evaluation of other instance settings such as instances with three or more clusters and instances in which the spatial distribution of requests is dependent on the time.

References