Problem-Specific State Space Partitioning for Dynamic Vehicle Routing Problems

Ninja Soeffker¹, Marlin W. Ulmer¹, and Dirk C. Mattfeld¹
¹ Technische Universität Braunschweig, Institut für Wirtschaftsinformatik, Lehrstuhl Decision Support, n.soeffker@tu-braunschweig.de

Abstract

In many dynamic vehicle routing applications, anticipation of the future is essential for efficient and effective decision making. In order to incorporate possible future events into current decision making, it is necessary to evaluate every decision’s impact to the overall objective. This is typically achieved by an evaluation of the post-decision state. For dynamic vehicle routing problems (DVRP), the size of the state space can be infinitely large. Therefore, states are often aggregated to enable an evaluation. In many cases, the aggregated state space has to be partitioned and values are assigned to partitions, i.e., sets of single states. The partitioning has a major impact on the solution quality. This paper presents a generic approach that exploits information about the problem characteristics for a problem-specific state space (PSP). The proposed approach is evaluated for a DVRP with one vehicle and stochastic customer requests (DVRPSR). Results show that the solution quality can be increased significantly by a problem-specific partitioning and that anticipatory decision making can be achieved efficiently.

1 Introduction

In many real world routing applications of parcel services, decisions have to be made under uncertainty since information only becomes available over the decision horizon. It is therefore necessary to adapt the current routing, i.e., to make decisions in successive points of time. Problems with this characteristic are modeled as DVRP. In every decision point, it is necessary to not only consider the immediate reward resulting from the decision, but also the possible future decisions and rewards. Perfect information about the future is not available. The value of the impact of a decision in a specific state therefore has to be approximated. Methods that approximate values for an anticipating decision making are provided by Approximate Dynamic Programing (Powell, 2011). However, the state space is often still large and the evaluation requires the storage of values for every state. The number of possible states can be infinitely large for two reasons: a) if many describing features are necessary to identify a state, b) if one of the describing features is continuous. By selecting relevant features of a state (e.g., point of time), states can be aggregated. The dimensionality of the state space is reduced which enables the above mentioned approximation of the value of a state. Since the aggregated state space is often still vast if, for example, one of the
chosen features (like the time) is continuous, the aggregated state space is often partitioned. Typically, this is done in an equidistant and problem-independent way. These partitionings may not be efficient since some areas might be considered in too much detail and are therefore not observed frequently to achieve a reliable approximation while other areas might not be detailed enough and aggregate dissimilar states resulting in inefficient decision making. This paper therefore introduces a problem-specific state space partitioning approach. PSP exploits information about the problem and structures the state space partitioning such that areas are represented that are observed frequently and where a detailed representation is necessary for a reliable approximation.

In this paper, PSP is applied to the problem of a vehicle serving customer requests in a service area within a time limit. Some requests are known before the start of the tour and have to be served. Other requests occur during the day, neither the request time nor the request location are known before the request occurs. The objective of parcel services is to serve as many customer requests as possible. Due to the time limit, however, it is mostly not possible to confirm all customer requests. It is therefore necessary to confirm requests such that the number of confirmed requests over the day can be maximized. Customers far off the preliminary tour can lead to significant detours that consume more of the time limit than other requests. This negatively impacts the future possibilities to confirm requests.

For the given problem, Ulmer et al. (2015) proposed the aggregation approach Anticipatory Time Budgeting (ATB) to aggregate states to vectors of time and slack. This paper proposes to use ATB in combination with PSP to partition the state space according to observed characteristics of the problem. Results suggest that an efficient state space partitioning can be achieved.

This paper is organized as follows. This paper presents literature on DVRPs and on state space partitioning in Section 2. The problem and its Markov Decision Process model are described in Section 3. The concepts of Anticipatory Time Budgeting, Approximate Value Iteration (AVI, Powell, 2011), and PSP are presented in Section 4. Results are summarized in Section 5 and conclusions and ideas for future research can be found in Section 6.

2 Literature Review

The following literature review focuses on both the considered problem and the methodology. Therefore, literature dealing with dynamic vehicle routing problems or with state space partitioning is reviewed.

2.1 Dynamic Vehicle Routing Problems

In this paper, a dynamic vehicle routing problem is considered. This refers to vehicle routing problems where the availability of information changes over time which requires subsequent decision making. Literature in this research area is vast. For detailed reviews on dynamic vehicle routing problem research, the interested reader is referred to Pillac et al. (2013) and Psaraftis et al. (2015). A DVRP where requests occur stochastically in known customer locations is discussed for example in Thomas (2007). The author discusses waiting strategies that lead to promising results despite the unknown customer requests. For a dynamic vehicle routing problem with multiple vehicles, some unknown customer locations, and time windows, Bent and van Hentenryck (2004) propose an approach with multiple scenarios including already known and possible future requests.
The considered DVRPSR is described for example in Ulmer et al. (2015). The authors develop a heuristic called Anticipatory Time Budgeting that manages the vehicle’s time budget anticipating the future.

2.2 State Space Partitioning

The partitioning of a state space is relevant not only in the Approximate Dynamic Programming (ADP) research, but also in the research area of reinforcement learning (e.g., Kröse and van Dam (1992)). As an example, Lee and Lau (2004) use reinforcement learning to learn state-actions pairs such that the overall rewards over the time horizon can be maximized. The authors use a temporal difference approach with adaptive vector quantization. Here, cells in the state space are created and new states are associated with the closest cell. The organization of the cells is managed by adding or merging cells.

Ulmer et al. (2015) discuss the issue of state space partitioning. The authors compare static and weighted lookup-tables with a new concept, the dynamic lookup-table. While this concept adapts its state space to the problem-specifics during the approximation process, it may require extensive memory if areas need to be considered in detail.

3 Dynamic Vehicle Routing Problem with Stochastic Customer Requests

In this section, the dynamic vehicle routing problem with stochastic customer requests is defined and modeled as a Markov Decision Process.

3.1 Problem

A vehicle starts its tour at a depot, serves customer requests in a service area within a time limit, and returns to the depot. Some requests for pickup services are known before the start of the tour, these have to be served. The remaining requests occur subsequently during the day. Neither request location nor request time are known beforehand, but they follow spatial and temporal distributions that are assumed to be known. These requests can be rejected. Typically, the time limit does not allow to confirm all customer requests. The situation of rejecting customers could be interpreted as forwarding them to other service providers or a backup service. This, however, is not desirable for a service provider.

3.2 Markov Decision Process

A Markov Decision Process can be used to model stochastic dynamic decision problems. The described problem is stochastic due to the probabilistic characteristics of the requests. It is also a dynamic problem due to the possibility of subsequent decision making. The Markov Decision Process in a decision point $k$ consists of a state $S_k$, an action $x$, a reward $R_k(S_k, x)$, a post-decision state $S^x_k$, and a transition to the next state.

For this problem, a decision point occurs whenever the vehicle arrives at a customer. A state $S_k$ is defined by the time $t$, the confirmed but not yet served customers (that is, their locations), the location of the vehicle, and the positions of the new requests. An action $x$ consists of two parts. Every request has to be either confirmed or rejected. In addition, one confirmed customer has to be chosen as the next customer to be visited. It is also possible to wait at the current location. The
reward $R_k$ is the number of immediate confirmations defined by the state and the chosen action. The combination of state and chosen action defines the known post-decision state (PDS) $S_k^x$. A post-decision state consists of time, the vehicle’s position, the confirmed but not yet served customers (who depend on the chosen action), and either the next customer to be visited or the decision to wait. After the post-decision state, a transition leads to the next state. The transition contains not only the travel to the next customer, but also all probabilistic customer requests that occur during the travel time.

The objective is to maximize the expected number of overall confirmations. This can be achieved by maximizing the sum of immediate confirmations and expected future confirmations. In every decision point, these expected future confirmations are defined as the value of the post-decision state $V(S_k^x)$. A policy $\pi$ assigns an action to every state. An optimal policy chooses in every decision point the action that maximizes the sum of immediate reward and expected future rewards. This concept was formulated in Bellman’s Equation (Bellman, 1957) and can be found in Equation (1).

$$X_k^\pi(S_k) = \arg \max_{x \in X(S_k)} \{ R_k(S_k, x) + V(S_k^x) \}$$  \hspace{1cm} (1)

The optimal policy $\pi^*$ applies Bellman’s Equation in every decision point.

4 Solution Methodology

The solution methodology will be explained in this section. The concept of ATB is described in subsection 4.1. The proposed approach PSP provides a problem-specific state space partitioning for ATB. The concept of PSP for ATB will be explained in subsection 4.2.

4.1 Decision Making: ATB

Due to the sizes of post-decision state space and transition space, values cannot be calculated exactly and have to be approximated (cf. Powell, 2011). ATB is an offline procedure since the approximation is an iterative process that requires more calculation time than is available in decision making. The approximation of values is therefore conducted in a learning phase. The approximated values then provide a policy which is applied in an execution phase. The concept of the value approximation in AVI is depicted in Figure 1. In AVI, states are evaluated using simulation. In every decision point within the simulation, the decision is determined by applying Bellman’s Equation using the policy induced by the approximated values. The decision is applied in the simulation and the stored approximated values are updated after one simulated run using the realized value.
Since an individual evaluation of post-decision states (in the following only called states) is not possible for the given problem due to the size of the state space, states have to be aggregated. It is important for the aggregation that only states are aggregated that are similar in terms of their value. To this end, the authors determined two main features that influence the value. One is the current time $t$ because it describes how much time is still left. The other one describes how much of the time is actually free and not already needed for serving the customers that are confirmed but have not been served yet. This property of a state is called slack $s$ and is an artificial key attribute derived by state properties and a planned tour. Using these two features, the state description can be aggregated to a vector to allow the application of AVI. For the given problem, the aggregated state space can still be large as time and slack are temporal attributes that are usually depicted minute by minute or even continuously. This aggregated state space can therefore be partitioned for a faster approximation process. One possible partitioning approach is a lookup-table, typically with partitions of equal length. The concept of lookup-tables, however, has some disadvantages. If the intervals are small, the memory consumption is problematic and the learning process may be impeded as the detailed states are visited only rarely. Further, adding additional state space features may lead to computational intractability. If the intervals are larger, dissimilar states are associated with the same partition and therefore the same value. Since it depends on the specific problem and instance which areas are needed in which detail, generic state space partitionings like lookup-tables may be both inefficient and ineffective.

4.2 Problem-Specific State Space Partitioning PSP

In order to provide a generic approach for the state space partitioning, exploitation of the problem structure is necessary. There is research done on adaptive state space partitioning, for example by Lee and Lau (2004). The state space is also adapted to the problem specifics to learn the value of state-action pairs. This is not applicable for problems in vehicle routing since both state and action spaces can be very large. It is therefore the objective to generate an approach that is also applicable to problems with state and action spaces of higher dimensionality. Due to this idea, the approach to be developed is applicable to problems with a state space which can include many different problem areas.

In this paper, PSP is presented as a problem-specific state space partitioning approach to evaluate post-decision states. To the best of our knowledge, such an approach is not available yet in the field of dynamic vehicle routing. The approach has three steps: a design phase, an approximation phase,
and an evaluation phase. In the design phase, states are sampled in a simulation. These states are then clustered to find representatives. These representatives define the state space partitioning. In the approximation phase, the representatives are used and the values of the representatives are approximated using simulation. In the evaluation phase, the policy achieved in the approximation phase is applied. An example of this development for the presented problem can be seen in Figure 2. On the x-axis, time is depicted, while the y-axis represents slack. The squares (a.) represent a problem-unspecific and equidistant partitioning of the aggregated state space which could also be described by representatives which are depicted as blue circles (also a.). Since the slack cannot exceed the remaining time, only representatives below or on the diagonal in the \((t, s)\)-plane are required in the design phase (a.). The lines of the squares in the diagram depict that the state space partitioning in the design phase resembles a lookup-table with intervals of equal length when using equidistant representatives. Since no further information about the problem is available, all representatives have the same value. The sampled states (b.) can be seen in the middle of Figure 2, they are depicted as yellow circles. After the sampling, the states are used for the clustering which leads to a new state space partitioning with representatives for the problem-specific partitions (c.). This partitioning of the aggregated state space is used for the approximation phase. The partitioning used in the approximation phase can be seen on the right side of Figure 2 and focuses on areas that appeared to be relevant for the problem. After the approximation phase, the approximated values are tested in the evaluation phase.

5 Computational Evaluation

This section first presents the instance details and continues with the parameters for the experiments. Furthermore, results are presented in this section.

5.1 Instances

The vehicle travels at a constant speed of 25 km/h in a service area of 20km x 20km, the time limit is 360 minutes. The tour starts and ends at a depot located in the middle of the service area. The vehicle is uncapacitated. 100 customers are expected to request service, 25 of them are requests known in advance. The remaining customers request service during the day and the request times are assumed to be distributed uniformly over time. The request locations are distributed either uniformly (U) or in two customer clusters (2C). For 2C, the customers are distributed normally around two cluster centers at (5km, 5km) and (15km, 15km) of the service area. For this customer distribution, the vehicle has to change customer clusters at some point of the tour. The different segments of the resulting tour require different levels of detail for the value approximation.
5.2 Parameters for Experiments

For obtaining and comparing results, three different approaches were used. One approach is myopic and accepts greedily as many requests in every decision point as possible. This heuristic is used in order to analyze the approaches regarding anticipation. The other two approaches apply ATB with two different state space partitionings. The first one uses equidistant representatives which resembles a lookup-table, the second one uses representatives obtained by PSP. For both approaches that apply ATB, 300 representatives are used.

For PSP, states are sampled over 300 simulation runs in the design phase. After the sampling, a k-medoids algorithm is used to find representatives. This clustering algorithm appears to be a good choice as the representatives have to be actual states that can be reached. This may be not relevant for the current combination of problem and state description. It could, however, be important for other problem features that are discrete and where states in between do not have a meaning (e.g., days). The chosen number of representatives is 300, k is therefore set to 300. Since not only the Euclidean distance in the \((t, s)\)-plane is important to determine the similarity of two states \(S_1\) and \(S_2\), the absolute difference of the observed values \(v_1\) and \(v_2\) is also included in the distance function that is used in the clustering. This distance function is presented in Equation (2). ATB with equidistant representatives does not require a design phase.

\[
d = \sqrt{\sum (t_2 - t_1)^2 + (s_2 - s_1)^2} \cdot \left| v_2 - v_1 \right|
\]  

Equation (2)

For both partitioning approaches, the values are approximated during 150,000 simulation runs in the approximation phase. Since realized values are not available, the Euclidean distance is used for the approximation process.

The quality of the approximated values is then evaluated by 10,000 simulation runs in the evaluation phase.

5.3 Partitionings

The first result regards the positions of the representatives obtained by the state space partitioning approach and how the positions of the representatives depend on the distribution of the customer requests. In Figure 3, the sets of representatives used for ATB are presented for both customer settings.

**Figure 3: Positions of Representatives in State Space**

The left part shows the scenario of uniformly distributed customers. Here, the representatives focus on the lower left area, there is no representative with a slack of more than 200 and only a few
representatives are needed for states with a time of more than 200. The representatives for 2C show a different structure. In general, many of the representatives have a higher slack and there are more representatives for those areas with a time of 200 and above. The reasons for this are linked with the explanation for the confirmation rates that are presented in Subsection 5.4.

5.4 Solution Quality

In Figure 4, the approaches using ATB are compared with a myopic confirmation policy.

![Figure 4: Confirmation Rates](image)

For uniformly distributed customers, early request customers are spread in the whole service area. Hence, the initial tour is already quite long. As a consequence, not many dynamic requests can be confirmed. If the customers are distributed in two customer clusters, the initial tour only concentrates on these areas and is typically shorter. This leads to more slack and facilitates more confirmations of dynamic customer requests.

It can also be seen that ATB with representatives obtained by PSP leads to better results in terms of the confirmation rate in comparison to ATB with equidistant representatives. For U, a greedy confirmation policy confirms customers far off the initial tour, while ATB results in higher confirmation rates as the anticipation leads to more rejections of those customers. Notably, ATB with PSP representatives only performs slightly better than the greedy approach which is more successful than ATB with equidistant representatives for 2C. For this scenario, customers are located in one of the two clusters with the same probability. The early customer requests, which have to be served, are very likely to lead to an initial tour already covering both customer clusters. The greedy confirmation policy accepts as many customer requests as possible. Since the requests are also in one of the two clusters, the additional time to serve the customer is typically not very high for this customer distribution scenario. The approach following a greedy confirmation policy therefore performs well as there are no customers far off a tour. ATB with only a relatively small number of equidistant representatives, however, aggregates states that are not similar. In this case, representatives are 15 time units apart in both dimensions. ATB with representatives obtained by PSP, on the other hand, partitions the state space in a more problem-specific way and is therefore able to approximate values in more detail which leads to higher confirmation rates.

The chosen distance function has a considerable effect on the resulting representatives. This can be seen comparing values and counters of the representatives after the approximation phase. The counter counts the number of observations associated with the partition. Figure 5 presents the counters and values of the representatives of the customer scenario U after the approximation phase.
The representatives obtained by PSP are not exactly located in areas with the highest counter because the value differences are also incorporated in the distance function used for the clustering. Due to this, areas are represented where it is important to capture differences in the values. Since the values for states late in the decision horizon are typically small, the absolute value differences are small and only a few representatives are found for this area. The same results for customer scenario 2C are presented in Figure 6.
Here, the overall values are typically higher as explained above. Differences of observed values can therefore be higher, a differentiation is required. The distance function in the clustering approach is able to provide such a differentiation while also focusing on frequently observed areas. The shape of the representatives obtained by PSP resemble the shape of the most frequently observed equidistant representatives and also focuses on areas with only some slack left. These areas are important for the value approximation because they have a small value due to the slack being close to zero. If these representatives were not present, states in this area would be overestimated.

Figure 7 shows an important area of the representatives obtained by PSP in more detail.

![Figure 7: PSP Representatives after Approximation Phase, 2C](image)

In the center of the figure, there is an area with only few representatives and the differences of the representatives’ values are rather large. The two representatives surrounded by black circles can be seen as an example for this property. The upper representative (134,88) has a value (an expected number of future confirmations) of 23.53, the lower one (134,50) has a value of 17.47. These differences can be explained by the problem and the customer distribution. Since customers are distributed in two clusters, the vehicle has to switch customer clusters at some point. In this area, a cluster change is likely to happen. Since the cluster change requires some time, many requests can occur in this time. As a result, the number of confirmed requests in the first decision point of the second cluster is typically high and the value of expected confirmed requests after a decision point drops drastically. The upper representative therefore represents a situation in which the vehicle did not drive to the second cluster yet, there are still many expected confirmations and the value is therefore high. In contrast, the lower representative represents a first PDS after the cluster change, the value and the slack dropped because many requests were confirmed. The large area with $100 \leq t \leq 140$ represents the area where only few observations occur due to the cluster change.
This theory can be supported by Figure 8 which shows the arrival of a vehicle in the second cluster over time. It can be seen that the cluster change happens in a very specific area when using equidistant representatives and that the representatives obtained by PSP allow more flexibility for the cluster change.

![Figure 8: Arrival Times after Cluster Change, 2C](image)

In essence, PSP is able to partition the state space according to the problem and instances which efficiently provides sufficient approximation and an anticipatory policy.

### 6 Conclusion and Future Research

Decision making for dynamic vehicle routing problems is difficult due to the necessary anticipation of the future. This anticipation is often achieved by methods of approximate dynamic programming. Because of the size of the state space, both an aggregation and a partitioning of the state space are required for DVRPs.

In this paper, a state space partitioning approach is proposed and in combination with the aggregation approach ATB applied to a vehicle routing problem with stochastic customer requests. The combination of ATB and PSP utilizes the approximation of values as a method of approximate dynamic programming and aims on designing the representatives of the state space for the value approximation in a problem-specific way. The locations of the achieved representatives are reasonable considering the proposed distance function. Results show that the solution quality can be improved and anticipation is enabled when using the problem-specific representatives instead of the same number of equidistant representatives. The main purpose of this paper is to show the potential of PSP in the field of DVRPs.

One main advantage of PSP is that various distance functions can be used for the clustering. This may allow adding additional attributes to the aggregated state space. Using other distance functions that are more specific may, however, require some insight into the specific problem. Using additional attributes or different distance functions might be subject of future research. This would also allow, for example, to consider a multi-periodical perspective where customers do not get rejected but have to be served the next day. Further, PSP is able to find the important areas of the
state space rather fast and could even be used to create a state space that adapts over time. In the future, multiple clustering might be used for a more effective learning. These modifications might further improve the efficiency of PSP.

7 Literature


