MECHANICAL ENGINEERING RESEARCH
An International Journal

Full Paper

MERAIJ, 1(1), 2016 [001-009]

Anticipatory routing of a service vehicle

Nan Marin Silviu*, Constantin P.Bogdan
Faculty of Mechanical and Electrical Engineering, Department of Mechanical Engineering, Electrical and Transport, University of Petrosani, The town of Petrosani, (ROMANIA)
E-mail nan.marins@gmail.com; tica1234ticabogd@yahoo.com

ABSTRACT

The following sections introduce a selection of approaches to anticipatory optimizationfor dynamic routing of a service vehicle and covers perfect anticipation, which – from a practical point of view – is limited to small problem instances. Nevertheless, the realization of perfect anticipation forsmall instances provides valuable insights with respect to approaches featuring lower degrees of anticipation. A number of new approaches featuring a lower degree of anticipation develops actor-critic methods for the problem of dynamic routing of a service vehicle. Proposes a variety of non-reactive anticipatory approaches to dynamic routing of a service vehicle. Part of the non-reactive approaches are inspired by ideas present in the literature while others are entirely new approaches ever to realization of approximate anticipation for dynamic vehicle routing with late customer requests.

KEYWORDS

Optimization;
Mecanics;
Robotics.

INTRODUCTION

However, a successful realization ofperfect anticipation requires some additional reasoning that. Based on thatprovides empirical results showing that forward dynamicprogramming generates a perfect solution. Finallysummarizes the limited effectivness of perfectanticipation with respect to larger instances of the problem of dynamic routing of aservice vehicle.

State sampling

However, a sufficient number of updates for every state must be guaranteed in order to permit identification of an optimal policy. Convergence to the true state values even requires that every state may potentially be visited infinitely often. Following the real-time dynamic programming principle as selects the next state to receive an update by means of making a decision of type

$$d_t^* \leftarrow \arg \max_{d \in D_t(s_t)} [ct(s_t,d) + \bar{V}_t^{d,\pi^{n-1}}(s_t^d) \quad (1)$$

Such a decision fully determines the next post-decision state, which will receive anupdate as soon as a new sample estimate of the value of the following pre-decisionstate is available. This means that the next state to receive an update heavily dependson the current value function estimates $\bar{V}_t^{d,\pi^{n-1}}(s_t^d)$ resulting from the previous iterations.
However, exploiting the experience made up to the current iteration forchosing the next state to receive an update may turn out to be a pitfall. Assume that an adequate stepsize has been selected for the case of

Full Paper

$_{t2}^{*}$= 3. In this case, an optimal policy π* must for example rejectcustomer 2 if he requests for service at t_2 = 1 without any other customer havingrequested before. Thus, starting from state s_1= (0,0,1,0), π* must chose state s_1^d =(0,0,3,0)over both state s_1^d = (0,0,2,0)and state s_1^d =(2,0,3,0). This, however, atleast requires $\bar{V}_1^{d,\pi^{n-1}}$ (0,0,3,0) , because otherwise the contribution $c_1$receivedfrom confirmation of customer 2 be exceeded. As the estimated values of the states (0,0,3,0), (0,0,2,0), (2,0,3,0)$\in s_1^d$ areinitializedas $\bar{V}_1^{d,\pi^0}$ (0,0,3,0) = (0,0,2,0) = $\bar{V}_1^{d,\pi^0}$ (2,0,3,0) = 0, s_1^d = (0,0,3,0)needs to be updated in order to permit $\bar{V}_1^{d,\pi^0}$. Unfortunately, such anupdate will never happen as long as the next state to be updated is selected accordingto a decision represented by Formula 7.1. Basically this modification consistsin separating more clearly the issue of selecting the next state to receive an update from the issue of generating a sample estimate. An illustration of the resulting typeof state sampling is given in Figure 1. The figure shows one (*gray-shaded*) state trajectory,which is generated for the purpose of selecting states that receive an update. On the contrary, the non gray-shaded states within the figure are visited only for the purpose of obtaining a sample estimate of the value of one of the gray-shaded post-decision states. Starting from the initial state s_{τ_0} = (0,0,0,0)at the first decision time τ_0= 0 two decisions aremade. The first decision d_0^* results fromapplication of Formula 1 and represents the first step of a partial state trajectory which is generated for obtaining a sample estimate of the value of the initial state. The ubsequent decisions required for reaching the terminal state $s_T$of this partial system trajectory are again made according to Formula 7.1. The single contributions occuring in the course of this trajectroy are used for updating the value of the pre-decision state s_{τ_0} , which at the same time may be considered as a virtual post-decision state, preceding . s_{τ_0} The second decision made in state s_{τ_0} is of type

$$d_t^z \leftarrow Z(D_t(s_t)) \qquad (2)$$

with $Z(\cdot)$ being a function that randomly returns one of the decisions within thecurrent set of feasible decisions. That is, in state s_{τ_0} , $d_0^z \leftarrow Z(D_0(s_0))$. Assigning thesame probability of selection to each of the feasible decisions at the current pointin time allows for every possible successor post-decision state to appear. The state $s^d{}_{\tau_0}$, resulting from decision d_0^z , is selected as the next state to receive an update.

In order to obtain a sample estimate for carrying out this update, another partialstate trajectory is enerated. This trajectory starts from state of the gray-shadedtrajectory and evolves by making decisions ccording to Formula 1. Moreover, atthe same decision time τ_1a decision $d_0^z \leftarrow Z(D_0(s_0))$. is made in order to step forwardto the next post-decision state

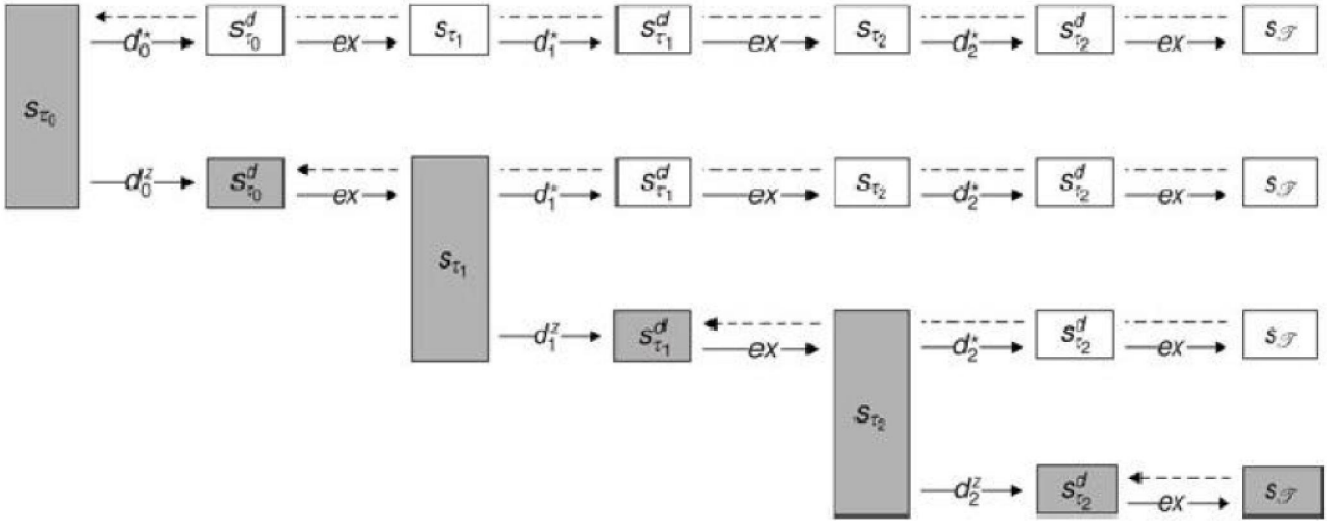

Figure 1 : Exploration sampling for TD(1)

to reveive an update. This alternating procedure of generating a post-decision state to be updated and a partial trajectory for obtaining a sample estimate is repeated until the terminal state of the gray-shaded trajectory is reached. Subsequently the procedure starts over from the initial state. The key issue of this sampling procedure consists in the fact that the states to be updated are no longer selected by exploiting the experience gained from the previous trajectories. Instead the randomized decisions introduce a type of exploration sampling that enforces updates of even such states that do not seem to be worth a visit with respect to the experience gained up to the current iteration.

Solution properties

In order to try to keep the amount of time spent on stepsize selection at a low level, it makes sense to check aut the performance of a simple constant stepsize first. Thus, in preparation of the empirical results illustrated in this section, a hundred iterations for each of five different constant stepsize values. As a result of these experimental runs a constant stepsize of 0.0005 was selected. Both Procedure 8 and its exploration sampling variant were applied with this stepsize. Moreover, a variety of settings of the customers' request probabilities were considered. Each of these settings shows similar effects, so that within the remainder of this section, the focus may be placed on one illustrative case. A nice illustration of the behavior of the two variants can be obtained for instance from the following setting. Let the individual request probabilities of both customer 1 and 3 be identical. In particular, assume that their probabilities of issuing a request within one single time unit are set to 0.125, which implies $t^*_2 = 4$. Moreover, let the probability of a request of customer 2 within one unit of time be 4.5×0.125. This fairly large value is helpful for clearly demonstrating the impact of being aware of t^*_2. Subject to this setting of request probabilities, an optimal policy must yield an expected number of $\bar{C} \approx 1.5687$ confirmations

Figure 2 outlines the evolution of the expected number of confirmations over the first 1,000 iterations of both the exploitation variant. Both variants update the states of one single trajectory within one iteration.

At each iteration, the expected number of confirmations of the current valuefunction estimates valuefunction estimates $\forall t \ \forall s_t^d : \bar{V}_t^{d,\pi^n}(s_t^d)$ is estimated $\pi^n$to the same set of 10,000 randomly generated test trajectories. Thus,in Figure 2 a data point denotes the average number of confirmations over these10,000 trajectories.

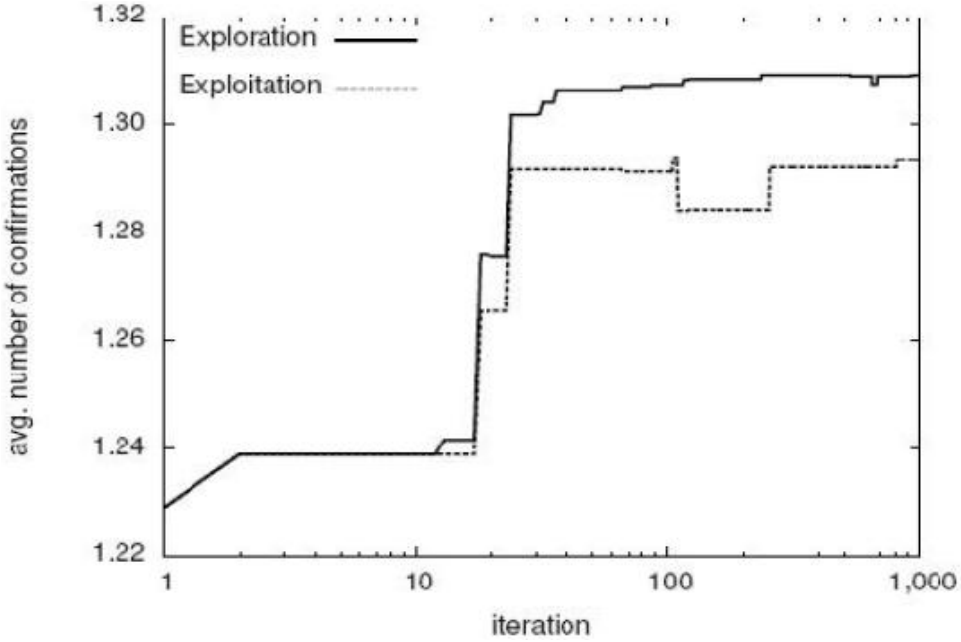

Figure 2 : Evolution of the solution quality over the first 1,000 iterations of Procedure 8 executing either exploration or exploitation state sampling

Full Paper

Limited effectiveness

The previous section shows that perfect anticipation works well for dynamic routing of a service vehicle with three customer locations. In general, however, perfect anticipation suffers from limited effectiveness.

Note that the additional challenge of evaluation of expected values has already been overcome by formulation of Bellman's equations around the post-decision state variables.

The tremendous reduction of the computational burden that is enabled by the omission of expected values is accompanied by a significant reduction of the size of the state space. As a customer $i \in I$ may never be in state r_{ti}= 1 immediately after a decision has been made, the overall size of the state space reduces from $(T+1)\times(|I|+2)\times 4|I|$ down to $(T+1)\times(|I|+2)\times 3|I|$ if post-decision states are considered.

Figure 3 illustrates the state space dimensionality at an arbitrary point in time t for both the pre-decision state variables and the post-decision state variables. The more customer locations a problem instance comprises, the bigger becomes the gap between the numbers $|S_t|$ and || of pre-decision states and post-decision states at time t. The small instance features |I| = 3, which results into $|S_t|$ =320 and $|s_1^d|$ = 135. However, increasing the number of customers for example to |I| = 50 implies $|S_t| \approx 10^{32}$ in contrast to a umber $|s_1^d|$ of "only" about 10^{25} post-decision states at one single point in time t. In spite of the benefits gained from making use of post-decision states, a realization of perfect anticipation quickly becomes prohibitive as the number of customers increases beyond |I| = 3. Although some states will usually be relevant only in theory, the potential $(T+1)\times|s_1^d|$ states to be handled are likely to imply an overwhelming computational burden already in the case of |I| = 10 for example. Additionally, the explosion of the computational burden may become much bigger, because a larger number of customers leads to an increase of the set of potential decisions.

If |I| = 3, both the sets $\bar{R}_t$ of customers that can be confirmed at t without a violation of the time horizon and the set $\bar{M}_t$ of feasible locations to move to next, may be determined quite easily by means of full enumeration. However, as |I| increases the number of possible decisions grows rapidly. As a consequence, merely verifying the feasibility of one single candidate decision may result in a computational task of nonpolynomial complexity. In view of the limited effectiveness of perfect anticipation, the next section proposes approaches to approximate an-

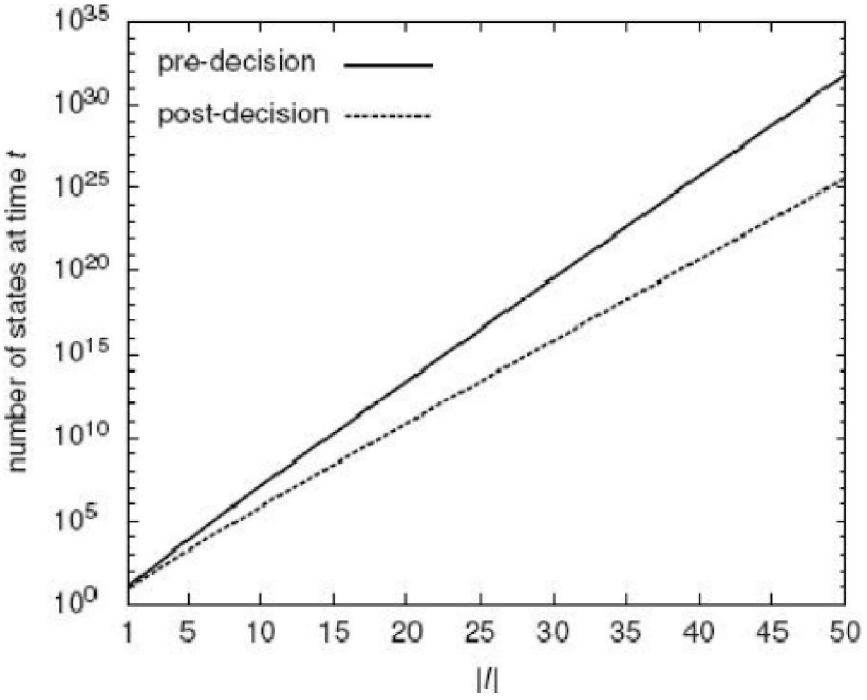

Figure 3 : The total number of states that may occur at an arbitrary point in time *t*.

ticipation for dynamic routing of a service vehicle.

CONTENTS

The following approach to approximate anticipation for dynamic routing of a service vehicle falls under the general actor-critic framework. In particular, it relies on value function approximation by means of regression. Although such a high-level categorization of the approach is accomplished quite easily, the realization of approximate anticipation is by no means straightforward. In contrast, Sect. 1.2.2 adresses the question of how to derive a decision model subject to a given system state.

Value function approximation

As a first step towards value function approximation, the problem of dynamic routing of a service vehicle is considered from a system perspective. The elements of a refined system model are summarized in Figure 1. This model comprises five types of objects. Both the variable and the constant attributes of customers, depots as well as road links are identical to those appearing in the basic system model. Merely the vehicle is represented at a slightly more detailed level. The basic model does not take into account the fact that in the strict sense the amount T of time available is an attribute of the vehicle. At first glance, the time available may seem to be an attribute of each of the system objects. Yet, on closer inspection it turns out to belong to the vehicle only, as, from an economic perspective, the vehicle is a resource to be utilized and consequently T determines the overall resource capacity. The decision making agent is defined by four variable attributes. the agent's decision at time t resolves into a confirmation operation dtc and a vehicle move operation dtm. Moreover, the evaluation attributes are given by the contribution ct as well as the value Vd t of the post-decision state sd t at time t. Note that the elements of a post-decision state $s_t^d = (n_t^d, r_{t1}^d, ..., r_{t|I|}^d)$ must not be represented explicitly in the system model, as they result implicitly from the predecision state $s_t = (n_t, r_{t1}, r_{t2}, ..., r_{t|I|})$ and the decision $d_t = (d_{tc}, d_{tm})$ at time t.

Besides two evaluation attributes and two decision attributes, Ãt comprises a number of variable system attributes given by the current position of the vehicle and the customers' request states. The remaining regular attributes of $\tilde{A}_t$ are constant. At a point in time t the value of any attribute except V_t^d can be observed. Thus, information about system structure is required in order to derive adequate surrogates for . Part of the system structure is represented by the value functions $\forall t \in T : V_t^d : S_t^d \rightarrow R$. Yet, value function approximation requires hypotheses about system structure. Such hypotheses may be formulated based on the following concepts and quantities, each of which can be extracted from a system appearance σ_t at time t:

Planned route

The planned route x_t^d is an ordered set whose elements are given by the customers $\{i | r_{ti}^d = 2Vi = n_t^d\}$ as well as the end depot E. It imposes a sequence on these elements and represents the current plan for serving every confirmed customer while traveling from n_t^d to the end depot, i.e., the first element of x_t^d is equal to n_t^d, while the last element is equal to E.

Route length

The quantity $l(x_t^d)$ represents the length of the planned route x_t^d. The length is equal to the total of those distances dist(·, ·), that are associated with the road links connecting the geographic locations of the elements of x_t^d according to the order imposed by x_t^d.

Presumed deviation

The quantity pd_t(i,) is the product of the request probability αi of customer *i* and the minimum additional distance to be traveled, if *i* must be inserted into . The presumed deviation is of relevance only with respect to customers that did not request for service yet. Thus, if at minimal cost such a customer *i* is inserted between two subsequent elements j,k x_t^d, the presumed deviation turns out to be $pd_t(i, x_t^d)$

Full Paper

$=\alpha_i(\text{dist}(j, i)+\text{dist}(i,k)+\text{dist}(j,k))$.

•*presumed number of requests*: The quantity pr_t represents the number of requests, that would be expected to occur throughout the time horizon, if only those customers were considered that did not issue a request until the current point in time *t*. Thus, the presumed number of requests can be derived from the requests probabilities αi as

$$pr_t = \sum_{i|r_{ti}} \alpha_i$$

Given a policy π, the true value $V_t^{d,\pi}(s_t^d)$ of a post-decision state s_t^d at time *t* may be expressed in terms of accumulated contributions $C_{t'}(s_{t'})$ of successor states $s_{t'}$, i.e.,

$$V_t^{d,\pi}(s_t^d) = \mathrm{E}[\mathrm{C}_{t'}(s_{t'})$$

In general, an accumulated contribution $C_{t'}(s_{t'})$ is – to a certain extent – determined by the number of customers that issues a request within (t...T]. Consequently, there is an influence of the number of these customers on the value $V_t^{d,\pi}(s_t^d)$. At time *t*, an upper bound of the number of future requests is given by the number of customers with $r_{ti} = 0$. Additionally taking into account the customers' request probabilities suggests that there is a correlation between the presumed number of requests *prt* and the post-decision state value . In the special case of *prt* = 0, the presumed number of requests fully determines $V_t^{d,\pi}(s_t^d)$ i.e., $\mathrm{pr}_t = 0 \Rightarrow V_t^{d,\pi}(s_t^d) = 0$. In many other cases, a larger value of pr_t is ikely to result into more future confirmations, which implies a larger value of the post-decision state *st*.

However, the correlation between *prt* and $V_t^{d,\pi}(s_t^d)$ may be weak. A larger value of pr_t will not result into an increase of $C_t(s_t)$ without the capability of confirming additional requests. At time *t* this capability is determined by both the available surplus of travel time and the travel time that is required for visiting additional customer locations. While the former is represented by the slack *slt* at *t*, an approximation of the latter may be gained by summing up the presumed deviations *pdt* of those customers that did not request for service until *t*. Both a larger amount of slack and a smaller total of presumed deviations are likely to increase the capability of making additional confirmations.

Bringing together the influence of the presumed number of requests and the influence of the capability of making additional confirmations gives rise to the following hypothesis.

Hypothesis 1: The higher the ratio of slack and average presumed deviation turns out to be at time t, the larger is the expected accumulated contribution at the following decision time. Any one unit increase of an arbitrary value of this ratio at time t implies an increase of the expected accumulated contribution by the same fixed amount. Thus, at time t

$$\mathrm{E}[\mathrm{C}_{t'}(s_{t'})\alpha \frac{sl_t}{\frac{\sum_{i|r_{ti}} pd_t}{pr_t}} = \frac{sl_t}{\sum_{i|r_{ti}} pd_t} pr_t \quad \textbf{(3)}$$

Note that the right hand side of Eq. 3 yields customer as unit. Thus, the ratio $\frac{sl_t}{\sum_{i|r_{ti}} pd_t} pr_t$

may be considered as the *presumed number of confirmed customers*, in opposition to the expected number of confirmed customers denoted by the leftmost expression of Eq. 3.

Hypothesis 1 may be consulted for a value function approximation that is basedon $\mathrm{g_t}(s_t^d) = (g_\sigma^0(\sigma_t))$, with

$$g_\sigma^0(\sigma_t) = \frac{sl_t}{\sum_{i|r_{ti}} pd_t} pr_t \quad \textbf{(4)}$$

Letting $r_t = (r_t^0)$ and imposing an information structure

$$\tilde{V}_6^d(r_t, g_t(s_t^d)) = r_t^0 g_\sigma^0(\sigma_t) \quad \textbf{(5)}$$

leads to a total of *T* different approximate value functions. This kind of value function approximation makes use of a single preprocessing operation $g_\sigma^0(\alpha_t)$. A value returned by (α_t) relies on the slack, the presumed number of requests as well as the presumed deviations extracted from the system appearance α_t at time *t*. As postulated by Hypothesis 1, a value function in the sense of Eqs. 4 and 5 assumes the same marginal utility r_t^0 of a presumed confir-

mation for any any state $s_t^d \in S_t^d$ that occur at time *t*.

While this assumption applies to many situations, there also are situations in which it proves to fail. In this scenario, the requesting customer 1 receives a confirmation and the vehicle could either remain at *gS* or move on to *g*1 immediately. The discussion shows that moving on to $g_1$increases the likelihood of an additional confirmation. Yet, applying $g_6(\cdot)$to both the decision state $s_6^{d,m}$ = (1,3,0,0)resulting from decision d_6^m = ({1},1)and the post-decision state = (0,2,0,0)resulting from decision d_6^w = ({1},0) produces exactly the same numbers of presumed confirmations. At this point, an approximate value function $\tilde{V}_6^d$ derived from Eq. 5 ignores the fact that the utility of a presumed confirmation is higher if thevehicle moves on to *g*1 instead of waiting at g_S. Consequently, the following refinement of Hypothesis 1 may be considered.

Hypothesis 2: The higher the ratio of slack and average presumed deviation turns out to be at time t, the larger is the expected accumulated contribution at the following decision time. Subject to the current vehicle position, a one unit increase of an arbitrary value of this ratio at time t implies an increase of the expected accumulated contribution by the same fixed amount. Thus, for each possible vehicle position at time t,

$$\mathrm{E}[C_{t'}(s_{t'})\alpha \frac{sl_t}{\frac{\sum_{i|r_{ti}} pd_t}{pr_t}} = \frac{sl_t}{\sum_{i|r_{ti}} pd_t} pr_t \qquad (6)$$

Hypothesis 2 represents an alternative to Hypothesis 1 and may be consulted for a value function approximation that is based on $g_t(s_t^d) = (g_\sigma^0(\sigma_t), g_\sigma^1(\sigma_t))$ with

$$g_\sigma^1(\sigma_t) = n_t^d \qquad (7)$$

An information structure allowing for different marginal utilities $r_t = (r_t^0, \ldots, r_t^{|I|})$of a presumed customer with respect to different post-decision vehicle positions n_t^d at time *t*, is then given by

$$\tilde{V}_t^d(r_t, g_t(s_t^d)) = r_t^0 g_\sigma^0(\sigma_t) \qquad (8)$$

This structure generates *T* different approximate value functions, with a single value function comprising |I|+1 parameters . Naturally, the more fine-grained Hypothesis 2 is reflected in a larger number of parameters to be determined. As a consequence, the computational effort for deriving the approximate value functions from Eq. 8 is likely to be significantly higher than the effort implied by Eq. 6.

Decision model identification

Selection of a type of value function approximation determines the structure of the objective functions for decision making within an actor-critic method. In the case of dynamic routing of a service vehicle, the structure of the full decision model at a time *t* comprises one of the value function approximations. As a result, the optimization problem to be solved at *t* is of type

$$P_t^{'} = (\{d_t \mid d_{tc} \in \bar{R}_t \wedge d_{tm} \in \bar{M}_t\}, \mid d_{tc} \mid + \tilde{V}_t^d(r_t, g_t(s_t^d)) \qquad (9)$$

with $\tilde{R}_t$ being the sets of requesting customers that may be confirmed without a violation of the time horizon and $\tilde{M}_t$ being the permitted vehicle moves subject to the confirmation decision d_{tc}.

Two steps are required for deriving a decision model from Eq. 8. On the one hand the parameters $r_t$of the approximate value function must be set. On the other hand, the candidate decisions making up the set of feasible decisions at *t* must be identified. A candidate decision *dt* = (d_{tc},d_{tm}) is feasible, if a planned route x_t^d with $sl_t(x_t^d) \geq 0$ exists. Thus, feasibility can be taken for granted if a route satisfying this condition can be specified. One approach to checking the feasibility of a candidate *dt* consists in solving the optimization problem

$$P_t^x = (X_t^d, sl_t(x_t^d)) \qquad (10)$$

with X_t^d being the set of planned routes that may be constructed at time *t* subject to *dt*. Solving Problem P_t^d by means of an exact algorithm returns a planned route $x_t^{d,*}$ 1

$$\forall x_t^d \in X_t^d : sl_t(x_t^{d,*}) \geq sl_t(x_t^d) \qquad (11)$$

In general, the availability of $x_t^{d,*}$ is both necessary and sufficient for making a definite statement

Full Paper

about the feasibility of *dt*. Clearly, feasibility is granted if $sl_t(x_t^{d,*}) \geq 0$, while *dt* is for sure infeasible in case $sl_t(x_t^{d,*})$ 0. However, finding $x_t^{d,*}$ may be prohibitive, as the optimization problem of Eq. 9 represents a variant of the traveling salesman problem. The latter belongs to the class of NP-complete problems, which means that an exact algorithm that guarantees a global optimum within polynomial runtime cannot be provided up to now.

Fortunately, in many cases a definite statement about the feasibility of *dt* can be made by considering a route $x_t^d \in x_t^d$ with x_t^d $xd\ t \neq x_t^{d,*}$. Insofar as implies $slt(x_t^d) \geq 0$, the candidate decision d_t can for sure be categorized as feasible. Thus, a non-optimal solution to Problem 7.9, which can be specified at a relatively low computational cost, may be sufficient. As discussed such a solution can be generatedby means of either heuristics or metaheuristics. Like most optimization techniques, the GRASPmetaheuristic has been successfully applied in many different contexts. Moreover, GRASP shows nice scaling properties, allowing for the method to either degenerate into a normal heuristic or to gradually unfold the qualities of a metaheuristic at the expense of a higher computational effort. For these reasons, GRASP is used for checking the feasibility of a candidate decision in the following. The resulting Procedure requires a candidate decision *d|k* and a planned route $x_{\tau_{k-1}}^d$ as input.

GRASP($d_{\tau_k}, x_{\tau_{k-1}}^d$)

$\varepsilon \leftarrow 0$

FOR NUMBER OF GRASP-ITERATIONS DO

$x \leftarrow$ setBaseRoute($x_{\tau_{k-1}}^d, d_{\tau_k m}$)

$x \leftarrow$ ConstructGreedyRandomizeSolution($x, d_{\tau_k c}$)

$x \leftarrow$ LocalSearch(x)

if $sl_{\tau_k}(x) \geq \varepsilon$ then

$x_{\tau_k}^d \leftarrow x$

$\varepsilon \leftarrow sl_{\tau_k}(x^d{}_{\tau_k})$

END

END

Hypothesis 3: Let dtc be a candidate set of confirmations at time t. Then, in general the most important tradeoff subject to a confirmation of dtc is whether the vehicle should wait at its current location nt or move on to another geographical location.

Against the background of the preceding discussion, Hypothesis 3 may be consulted for approximating the set of feasible decisions $D_t(s_t)$. Both d_t^w and d_t^m are promising candidate decisions that focus on different characteristics of the successor state. Considering only those two movement operations per candidate set d_{tc}leads to an approximate set $\tilde{D}_t(s_{t)}$ of feasible decisions. This set can be specified as

$\tilde{D}_t(s_t) = \cup_{d_{tc}} \in \Im(R_t)\tilde{D}(d_{tc}, x_t^d)$

with

a=dist (n_t,*(2))*

$$\tilde{D}(d_{tc}, x_t^d) = \begin{cases} \{d_t \mid d_{tm} \in \{n_t, x_t^d(2)\}\} : sl_t(x_t^d) \geq 0 \\ \{d_t \mid d_{tm} = x_t^d(2)\} : -a \leq sl_t(x_t^d) \leq -1 \\ \varnothing \quad : \quad sl_t(x_t^d) \leq -a \end{cases} \quad \textbf{(12)}$$

As a result, an optimization problem resulting from Eq. 8 may be replaced by a problem of type

$$P_t^{"} = (\tilde{D}_t(s_t), \mid d_{tc} \mid + \tilde{V}_t^d(r_t, g_t(s_t^d)) \quad \textbf{(13)}$$

CONCLUSIONS

However, adhering to the actor-critic framework does not necessarily lead tomore effective decisions. Possibly, the positive effect of the actor-critic principleis blurred by the hypothetical character of the downgrade from perfect anticipationto approximate anticipation. This article exclusively focuses on perfect anticipation. It develops three consecutive approaches to deriving perfect anticipatory decisions, each of which pursues the principle of dynamic programming. The first approach is given in terms of the elementary methods of dynamic programming. These provide the foundation of the second approach, which additionally consults simulation techniques and is known as forward dynamic programming. Forward dynamic programming implies a quite large family of methods that this article summarizes under the actor-critic framework. This

framework serves as a basis of model free dynamic programming representingthe third of the approaches to perfect anticipation. Concerning model free dynamic programming, the use of post-decision states proves to be of particular relevance, because this type of state may lead to a significant increase of the effectiveness of perfect anticipation.

REFERENCES

[1] E.Aarts, J.Korst; Simulated annealing and boltzmann machines, John Wiley & Sons, Chichester, UK, **(1989)**.

[2] Abbiw R.Jackson, B.Golden, S.Raghavan, E.Wasil; A divide–and–conquer local search heuristic for data visualization, Comput Oper.Res., **33(11)**, 3070–3087 **(2006)**.

[3] E.Agafonov, A.Bargiela, E.Burke, E.Peytchev; Mathematical justification of a heuristic for statistical correlation of real-life time series, Eur.J.Oper.Res., **198(1)**, 275–286 **(2009)**.

[4] R.Bellman; The theory of dynamic programming, Bull Am.Math.Soc, **60(6)**, 503–516 **(1954)**.

[5] R.Bellman; Dynamic programming, Princeton University Press, Princeton, NJ, **(1957a)**.

[6] R.Bellman; A markovian decision porcess, J.Math.Mech., **6(5)**, 679–684 **(1957b)**.

[7] J.Branke, M.Middendorf, G.Noeth, M.Dessouky; Waiting strategies for dynamic vehicle routing, Transp Sci., **39(3)**, 298–312 **(2005)**.

[8] Z.Chen, H.Xu; Dynamic column generation for dynamic vehicle routing with time windows, Transp Sci., **40(1)**,74–88 **(2006)**.

[9] A.George, W.Powell; Adaptive stepsizes for recursive estimation with applications in approximate dynamic programming, Mach Learn, **65(1)**, 167–198 **(2006)**.

[10] H.Robbins, S.Monro; A stochastic approximation method, Ann Math Stat., **22(3)**, 400–407 **(1951)**.

[11] C.Romanowski, R.Nagi; A data mining approach to forming generic bills of materials in support of variant design activities, J Comput Inf.Sci.Eng., **4(4)**, 316–328 **(2004)**.

[12] C.Romanowski, R.Nagi, M.Sudit; Data mining in engineering design environment: OR applications from graph matching, Comput Oper Res., **33(11)**, 3150–3160 **(2006)**.

[13] R.Rosen; Anticipatory systems: Philosophical, Mathematical and methodological foundations, Pergamon Press, Oxford, UK, **(1985)**.