Ideal and non-ideal predictors in estimation of Bellman function

Jan Zeman
Institute of Information Theory and Automation
Pod Vodarenskou vezi 4
CZ-182 08 Prague 8
Czech Republic
zeman@utia.cas.cz

Abstract

The paper considers estimation of Bellman function using revision of the past decisions. The original approach is further extended by employing predictions coming from an imperfect predictor. The resulting algorithm speeds up the convergence of Bellman function estimation and improves the results quality. The potential of the approach is demonstrated on a futures market data.

1 Introduction

The dynamic programming (DP) is clever and effective framework in many problems [1, 2]. Unfortunately, it also suffers by many issues such as curse of dimensionality [6]. Moreover, the incomplete knowledge and uncertainty makes the dynamic programming task hardly solvable, although the analytical solution of DP task is known [5]. The approximate dynamic programming (ADP) tries to solve the DP tasks and fight with all the technical issues. There are many ways to approximate the dynamic programming, but all of them assume that the approximation is precise enough to solve the problem. There is only a few approaches implicitly assuming the non-ideal or imperfect approximation. This paper presents such a approach related to solution of Bellman equation[1], i.e. estimation of Bellman function.[1]

The value iteration algorithm [1, 2, 4] makes the theoretical ground for estimation of Bellman function. It also suffers by dimensionality [4], which is often solved by approximate methods (see [3] for a review). There is an duality relation between optimal decision rule and Bellman function [1] and value iteration uses the idea of the convergent improving of Bellman function and decision rule - both together. Unfortunately, the value iteration is difficult to solve for tasks with continuous variables. We try to approximate Bellman function in value iteration and to speed up the convergence by searching the samples of the optimal decision rule [7]. The present approach can be extended using the prediction. This extension can bring only restricted impact, therefore must be considered a non-ideal predictor and its properties. Hence, the paper presents the imagination of the ideal and non-ideal predictor and their influence to estimation of Bellman function. These imaginations are compared and we point out the break, where the non-ideal one stops working. Respecting such a property, we can improve the estimation of Bellman function.

The paper briefly introduces the dynamic programming and estimation of Bellman function in Sec. 2. Then introduces the revisions and the related optimality criterion (Sec. 3) and considers the extension of the approach by additional usage of ideal and non-ideal prediction (Sec. 4). Finally, tests the presented idea on the task of trading commodity futures (Sec. 5).

[1] also called *value* function or *cost-to-go* function

2 Dynamic programming and revisions

We consider discrete time $t \in \{1, 2, \ldots, T\} = t^*$, where T is *horizon*. We consider *decision maker*, which is human or machine with particular aims to a part of world, so-called *system*. The decision maker observes the system and obtains *observation* y_t, then designs *decision* u_t and applies it to the system, this process is repeated at each time $t \in t^*$. The information available to design decision u_t is called knowledge P_t and consists of past observation and past decisions, $P_t = \{y_1, y_2, \ldots, y_t\} \cup \{u_1, u_2, \ldots, u_{t-1}\}$. The decision maker designs a decision rule, $u_t = \pi_t(P_t)$. The aim of the dynamic programming is to design the sequence of decision rules $\pi_1, \pi_2, \ldots, \pi_T$, so-called *strategy*, in order to maximize the sum of the gain functions $\sum_{t=1}^{T} g_t$ under the conditions above.

We consider the following task properties: (i) The decision maker and its environment work in open loop, i.e. decisions have influence neither on the environment behavior nor future observation; (ii) gain function has form $g_t(P_t, u_t)$ and it depends at last n observations and decisions, i.e. $g_t(y_{t-n}, \ldots, y_t, u_{t-n}, \ldots, u_t)$, where n is finite number.

2.1 Finite and infinite horizon

The optimal rule for finite horizon T can be constructed value-wise (see [5])

$$\pi_t^o(P_t) = \arg\max_{u_t} \mathbb{E}\left[g_t + \mathcal{V}_{t+1}(P_{t+1})|u_t, P_t\right], \tag{1}$$

where function $\mathcal{V}_{t+1}(.)$ is *Bellman function* and it is given by recurrence

$$\mathcal{V}_t(P_t) = \max_{u_t} \mathbb{E}\left[g_t + \mathcal{V}_{t+1}(P_{t+1})|u_t, P_t\right] \tag{2}$$

with the terminal condition

$$\mathcal{V}_{T+1}(P_{T+1}) = 0. \tag{3}$$

The equations (1, 2, 3) form the algorithm of dynamic programing with finite horizon T. This algorithm is important for revisions.

We consider task with infinite horizon $T = +\infty$. Consequently, the solution has stationary form:

$$\pi^o(P_t) = \arg\max_{u_t} \mathbb{E}\left[g_t + \mathcal{V}(P_{t+1})|u_t, P_t\right], \tag{4}$$

where function $\mathcal{V}$ is stationary Bellman function and it is given by recurrence

$$\mathcal{V}(P_t) = \max_{u_t} \mathbb{E}\left[g_t + \mathcal{V}(P_{t+1})|u_t, P_t\right]. \tag{5}$$

The equation (4) contains two terms at right-hand side. In general, both therms in (4) can be calculated difficultly due to uncertainty, incomplete knowledge, demanded prediction etc. Hence, ADP considers approaches how to calculate the terms, or to approximate them adequately. We focus on the approximation of Bellman function.

2.2 Estimation of Bellman function

Let us consider the infinite horizon task. The equation (4) contains two terms at right-hand side. First term is gain function g_t, which can be evaluated under knowledge P_t for the considered shape of g_t. Second term is Bellman function $\mathcal{V}(.)$ in stationary form applied on unavailable knowledge P_{t+1}. Under knowledge P_t the decision maker must predict further knowledge P_{t+1} and estimate Bellman function.

Let us assume that we have ideal predictor $\mathcal{M}^I(.)$ such as it can predict $P_{t+1} = \mathcal{M}^I(P_t)$. Equation (5) can be written for each time index $i \in \{1, \ldots, t\}$

$$\mathcal{V}(P_i) = \max_{u_i} \mathbb{E}\left[g_i + \mathcal{V}(\mathcal{M}^I(P_i))|u_i, P_i\right]. \tag{6}$$

The obtained t-equations system contains the main information about Bellman function $\mathcal{V}(.)$. Assuming the knowledge of the optimal decisions $u_1^o, \ldots, u_t^o$, the system is transformed to final form:

$$\mathcal{V}(P_i) = \mathbb{E}\left[g_i + \mathcal{V}(\mathcal{M}^I(P_i))|u_i^o, P_i\right], \quad \text{for } i \in \{1, \ldots, t\}. \tag{7}$$

This equations system contains only unknown function $\mathcal{V}(.)$ and can be used to estimation Bellman function [7]. The information contained in (7) is not full, therefore this system can bring only an approximate solution. This approximate solution can be found considering approximation of Bellman function in parametrized form $\mathcal{V}(.) \approx V(., \Theta)$, where Θ is finite dimensional unknown parameter. The system (7) characterizes points of Bellman function and inserting the approximation $V(., \Theta)$ the system is transformed to system for unknown variable Θ. Typically, the number of equations in (7) is bigger than dimension of parameter Θ. Hence, the best estimation of parameter $\hat{\Theta}$ is searched by regression methods.

This approach originates from value iteration [4] and the system (7) can be interpreted as subsystem of the full system:

$$\mathcal{V}(P) = \mathbb{E}\left[g_i + \mathcal{V}(\mathcal{M}^I(P))|\pi^o(P), P\right], \quad \text{for } P \in P^*. \tag{8}$$

Bellman function $\mathcal{V}(.)$ is a solution of system (8). The formal difference between system (7) and (8) is in the used knowledge. While system (8) contains all possible values of P, the system (7) contains only the realizations passed during the decision process $P_1, P_2, \ldots, P_t$. We have assumed the knowledge of the optimal decisions related to these realizations, therefore the term $\pi^o(P)$ is known only for these realizations $u_i^o = \pi^o(P_i)$ for $i \in \{1, 2, \ldots, t\}$. All in all, the system (8) contains a full information about Bellman function, whereas the system (7) contains only t points of Bellman function.

3 Revisions

The previous approach depends on the possibility to find the optimal decisions $u_1^o, \ldots, u_t^o$ for the given knowledge P_t. This is possible to obtain by the revisions.

The revision is the reconsideration of the decision under another knowledge than was used to design it. To design decision the maximal available knowledge is used, but we can redesign the decision under higher knowledge; let us denote the rules and the decisions by superscript, which characterizes the knowledge used to design the rule, e.g. $u_t^{t+i} = \pi^{t+i}(P_t)$ is redesign of the tth rule/decision under knowledge P_{t+i}. But we omit the superscript, when the rule/decision is designed under natural conditions $u_t = u_t^t = \pi_t^t(P_t) = \pi_t(P_t)$ is rule/decision designed under the knowledge available to design it. This differs the revision and the decisions. One clever way of this redesign is solving the same task, but with the finite horizon $T \equiv t$. We can reconsider all decision using the equations (1, 2, 3). And obtain the revision based on the knowledge P_t:

$$U_t^t = \{u_1^t, u_2^t, u_3^t, \ldots, u_t^t\}, \tag{9}$$

where $u_i^t = \pi_i^t(P_t)$ for $i \in \{1, 2, \ldots, t\}$. The sequence U_t^t is called t-revision. Due to asymptotic properties of DP [5], the revisions tends to optimal values, i.e. $u_i^t = \pi_i^t(P_i) \to \pi_i^o(P_i) = u_i^o$.

3.1 Optimality of revision

For our special shape of the gain function, the convergence can be interpreted as weighting of the influence of the terminal condition (3) and information contained in data, inserted into g_t. The algorithm of searching of t-revision goes backwards and from design of $\pi_t^t(.)$ to $\pi_1^t(.)$. The terminal condition (3) influences a decision rule $\pi_i^t(.)$ via Bellman equation (2). But the information-rich data can quickly decrease the influence, such as $\pi_t^t(.)$ is influenced, but further $\pi_{t-1}^t(.), \pi_{t-2}^t(.), \ldots$ are influenced less and less. When the influence of terminal condition is lost for $l \in \{1, 2, \ldots, t\}$ and $\pi_l^t(.)$ is independent on terminal condition (3), then the decision rule $\pi_i^t(.)$ maps the knowledge to optimal decision, where $i \in \{l, l-1, l-2, \ldots, 1\}$. The optimality is given by the independence on the terminal condition and the absolute dependence on the data.

The issue is to recognize, whether the optimality was reached. This factor can negative influence the potential of estimation of Bellman function. The bad recognized optimality can lead in: learning from non-optimal decisions, i.e. adding the non-valid equations to system (7); or redundant omission of some optimal decisions, i.e. omission available equations of system (7). Hence, the preciseness of optimality recognition is required.

The possible way how to recognize the optimal decision lies in independence on the terminal condition (3). Let us consider the terminal condition in form:

$$\mathcal{V}_{T+1}(P_{T+1}) = f(P_{T+1}), \tag{10}$$

where $f(.)$ is general function of P_{T+1}. Let us denote the class of all those functions $\mathfrak{F}$. The revision algorithm (1, 2, 3) can be generalized by usage the terminal condition (10) instead of (3). Then, the revision can be written as function of knowledge and terminal condition $u_i^t = \pi_i^t(P_i, f)$.

Finally, *the revision of decision u_i^t equals optimal decision u_i^o, if the revision does not depend on terminal condition (10), i.e.*

$$\exists \tilde{u}_i \in u^* \quad \forall f \in \mathfrak{F} \quad \pi_i^t(P_i, f) = \tilde{u}_i. \tag{11}$$

and the constant $\tilde{u}_i$ is the optimal decision, $u_i^o = \tilde{u}_i$.

The impact of this proposition is great, because it represents the inter-connection between the finite and infinite horizon task. The proposition gives algorithm how to use the generalized finite horizon task (1, 2, 10) to find some optimal decisions of infinite horizon task (4, 5). Unfortunately, the proposition does not guarantee that any optimal decisions will be found. Typically, there exists an index t^o such that revisions $u_1^t, u_2^t, \ldots, u_{t^o}^t$ are optimal and independent on f; and revisions $u_{t^o+1}^t, \ldots, u_t^t$ cannot be decided, whether are optimal because of the dependence on f.

The proposition uses simply idea that the interconnection between two consequent decisions is done via Bellman equation and the connection term is Bellman function. Using the right Bellman function $\mathcal{V}_{t+1}(.)$ we could connect the finite horizon task and infinite horizon task easily. Unfortunately, Bellman function is unknown therefore the proposition must go over all possible candidates $f \in \mathfrak{F}$, i.e. over all possible interconnections. Having a bit information about Bellman function, it is possible to exclude the impossible candidates and use the proposition over subset $\mathfrak{F}' \subset \mathfrak{F}$ containing only the possible ones. This idea can be reached by usage of predictions.

4 Revision and prediction

As was mentioned above, we can operate with ideal predictor $\mathcal{M}^I(.)$ such as $P_{t+1} = \mathcal{M}^I(P_t)$. Having the ideal predictor, we can use it recursively to predict P_{t+i} for any $i > 0$ and use the P_{t+i} as information for revision and searching its optimality. Such a approach can help us to increase the value t^o and use all available equations of system (7).

Let us consider the revision algorithm. For P_t, the algorithm starts with terminal condition (10). For one-step prediction P_{t+1}, the algorithm has one more step and due to back recursion in (2) obtain $\mathcal{V}_{t+1}$, i.e. the restricted analogy of condition (10), after one step:

$$\mathcal{V}_{t+2}(P_{t+2}) = f(P_{t+2}), \tag{12}$$

$$\mathcal{V}_{t+1}(P_{t+1}) = \max_{u_{t+1}} \mathbb{E}\left[g_{t+1} + \mathcal{V}_{t+2}(P_{t+2}) | u_{t+1}, P_{t+1}\right], \tag{13}$$

where we expect that $f \in \mathfrak{F}$, and $\mathcal{V}_{t+1}(P_{t+1}) \in \mathfrak{F}_1 \subseteq \mathfrak{F}$.

This expectation originates from properties of Bellman equation, which can be viewed as operator on class $\mathfrak{F}$, i.e. $\mathcal{V}_i = \mathcal{T}(\mathcal{V}_{i+1})$, see [1, 2]. The recursion (2) converges for each terminal condition. Consequently, the operator has following property $\lim_{n\to+\infty} \mathcal{T}^n(f) = \mathcal{V}$, where $\mathcal{T}^n(.)$ is operator $\mathcal{T}(.)$ n-times recursively applied onto $f \in \mathfrak{F}$ and $\mathcal{V}$ is Bellman function. Hence, we can expect that the operator $\mathcal{T}(.)$ applied on all functions in $\mathfrak{F}$ produces the subset of $\mathfrak{F}$:

$$\mathfrak{F}_1 = \mathcal{T}(\mathfrak{F}) \quad \text{and} \quad \mathfrak{F}_1 \subseteq \mathfrak{F}. \tag{14}$$

Furthermore, each prediction step can be used as one more application of operator $\mathcal{T}(.)$, which reduces the set of possible candidates to terminal $\mathcal{V}_{t+1}$. The h-step prediction generates h subsets of $\mathfrak{F}$ as is depicted at Fig. 1. The usage of prediction can be interpreted as starting the optimality criterion from less set $\mathfrak{F}_i$ instead of $\mathfrak{F}$, which can result in earlier recognizing the optimal decisions, i.e. obtaining higher value t^o.

Of course, we do not have the ideal predictor $\mathcal{M}^I(.)$, but often we can use an predictor $\hat{P}_{t+1} = \mathcal{M}(P_t)$. We assume that the predictor $\mathcal{M}(.)$ has some restricted preciseness and degenerates the

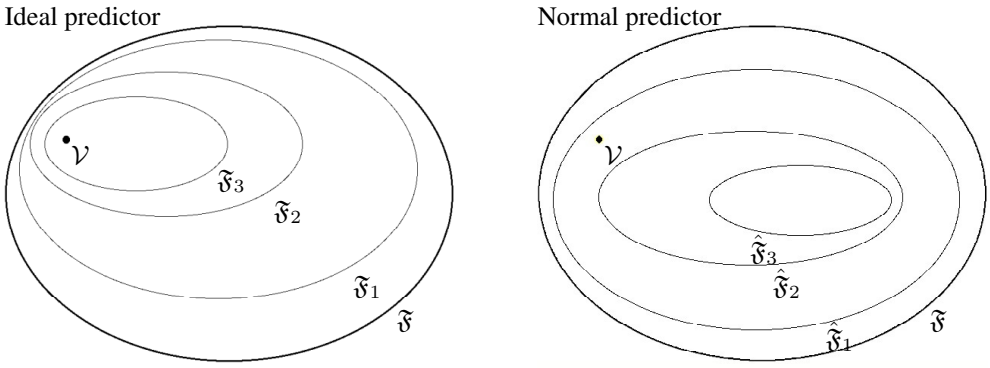

Figure 1: The convergence of operator J in space of possible Bellman functions: $\mathcal{V}$ is optimal Bellman function, $\mathfrak{F}$ is space of possible Bellman functions.

operator $\mathcal{T} \to \hat{\mathcal{T}}$. Let us denote: $\hat{\mathfrak{F}}_1 = \hat{\mathcal{T}}(\mathfrak{F})$ and $\hat{\mathfrak{F}}_{i+1} = \hat{\mathcal{T}}(\hat{\mathfrak{F}}_i)$. Typically, the non-ideal predictor predicts quite good first one or two steps and then the predictions go worse. The expected influence to Bellman function is depicted at Fig. 1, where the 'Normal predictor' gets lost in second step and the operator $\hat{\mathcal{T}}$ converges to the other function. Despite of this fact, the non-ideal predictor can be successfully used, when the number of prediction steps is restricted. The restriction equals to index of the last set including Bellman function; e.g. it can be used only as one-step predictor in case depicted at Fig. 1. We expect that this phenomenon would be observable as relation between length of used predictions and results quality. We expect slowly increase followed by rapid decrease of results quality according to the growing prediction length.

5 Experiment

The following experiment should demonstrate the expected properties. We compare: (i) the original method to estimation of Bellman function, where the optimal revisions were searched at available data, i.e. the terminal condition (10) was taken over whole set $\mathfrak{F}$; and (ii) the presented method, where the optimal revisions were searched at available data extended by prediction, i.e. the terminal condition (10) was taken over the subset $\hat{\mathfrak{F}}_i \subseteq \mathfrak{F}$. The set of experiments contains 11 experiments per data sequence. Each experiment is related with one prediction length 0-10, where zero length is the original task using $\mathfrak{F}$ and the other lengths $l \in \{1, 2, \ldots, 10\}$ corresponds with the systems of sets $\hat{\mathfrak{F}}_1, \ldots, \hat{\mathfrak{F}}_{10}$ (see previous section).

We expect that results quality will grow with length of prediction to some break value. Then, the result quality will decrease. This expectation is caused by the imagination of the non-ideal predictor analogical to Fig. 1, where Bellman function is in $\hat{\mathfrak{F}}_1$, but it is not in $\hat{\mathfrak{F}}_2$. Thus, we expect that each data set should have some length of prediction l, where Bellman function is in $\hat{\mathfrak{F}}_l$, but it is not in $\hat{\mathfrak{F}}_{l+1}$. The approach to estimate of Bellman function should work most effective, when starts with the smallest subset $\hat{\mathfrak{F}}_l$ containing the Bellman function $\mathcal{V}(.)$. Otherwise, when it starts with subset $\hat{\mathfrak{F}}_{l+1}$, it need more information to find Bellman function, because it got lost by irrelevant set $\hat{\mathfrak{F}}_{l+1}$ and the convergence is delayed. We expect that this phenomenon should be observable as results quality increase for prediction length $1, 2, \ldots, l$, followed by quality decrease for prediction length $l+1, \ldots, 10$.

The experiment was done on trend prediction task based on the trading with commodities. The task is classical price speculation, where decision maker tries to predict future price trend and chose the decision to follow the trend. The gain function has shape:

$$g_t = (y_t - y_{t-1})u_{t-1} + C|u_{t-1} - u_t|, \tag{15}$$

where y_{t-1}, y_t are samples of price sequence, u_{t-1}, u_t are decisions and C is transaction cost. The decision can be chosen from two-values set $u_t \in \{-1, 1\}$, where $u_t = 1$ characterizes the future price increase and $u_t = -1$ characterizes decrease.

The data used for experiment are day samples of price, so-called close price. The used time series are related to following five commodities: Cocoa - CSCE (CC), Petroleum-Crude Oil Light - NMX (CL), 5-Year U.S. Treasury Note - CBT (FV2), Japanese Yen CME (JY), Wheat - CBT (W). The used data were collected between January 1990 and September 2005, which is about 4000 trading days.

The experiment designs the decisions via approximated (4). The predictor $\mathcal{M}(.)$ is based on the autoregressive model, $y_{t+1} = \alpha y_t + \beta y_{t-1} + e_t$, where α, β are model parameters and e_t is noise, $e_t \approx N(\mu, \sigma)$. The model parameters are estimated via Bayesian estimation [5]. The prediction is calculated recursively $\hat{P}_{t+1} = \mathcal{M}(P_t)$, and $\hat{P}_{t+i+1} = \mathcal{M}(\hat{P}_{t+i})$ for $i \in \{1, \ldots, 9\}$. And Bellman function is approximated in parametrized form:

$$\mathcal{V}(P_t) \approx V(P_t, \Theta) = \Theta' \Psi_t(P_t), \tag{16}$$

where Θ is vector of $n+1$ parameters and $\Psi_t(P_t) = (y_t, y_{t-1}, \ldots, y_{t-n+1}, 1)'$, which is 1st order Taylor expansion of Bellman function $\mathcal{V}(.)$. The parameters Θ are estimated via system (7) and the count of equation t^o in system (7) is estimated via revisions and the optimality proposition.

Table 1 contains experiment results. According to our expectation, the results written by bold font have the expected growing quality. As can be seen, the prediction improved the results quality in all datasets according to non-prediction experiment for $l = 0$. The length of growing trend is related with feasibility of the predictor to the dataset, and it was expected that the 1- or 2-step prediction can improve the results. Hence, the results of 3-step prediction at CC and JY can be viewed as unexpected success. A little surprising fact is the quality of results after the increase. We have expected the rapid decrease due to worse initial conditions, but a few experiments reached comparatively results or better results. The expected behavior can be demonstrated at CL dataset, where $l \in \{0, 1, 2\}$ the results quality grows and then fall down and stay under the value for $l = 0$. An representative of the surprising is JY dataset, which grows for $l \in \{0, 1, 2\}$, then it decrease, but then it increases and reaches better results than for $l \in \{0, 1, 2\}$. The mentioned facts lead to conclusion that the prediction improve the results for a few steps. But after these steps, there cannot be expected any property or trend related to the prediction length.

6 Conclusion

The paper presents the approach to estimation of Bellman function via revisions. The revisions are originally calculated from the knowledge available to design the decision. The paper considers extension of this approach by the usage of predictions. It is expected the better convergence to Bellman function. The idea is considered for ideal predictor and non-ideal predictor. The ideal predictor can simply improve the algorithm, but it is unavailable, whereas all available predictors can be classified as non-ideal. The imagination of the non-ideal predictor leads to expectation that the prediction can improve the approach, when is used a restricted number of prediction steps.

The idea is experimentally tested on trend prediction task, where works quite well. The results have verified the idea that the improvement is related to restricted number of prediction steps. But surprising was the fact that after these few steps, the improvement can be reached, but randomly. This opens the question of the better analysis of the problem: the paper describes only a raw imagination of the problem and the convergence in set $\mathfrak{F}$, and relations between sets $\hat{\mathfrak{F}}_i$ and $\hat{\mathfrak{F}}_{i+1}$ can be more complex than was presented. This fact is topic of the further consideration.

Moreover, the paper presents that the number of prediction steps should be restricted, but it does not give any guidelines how to estimate the right length of the prediction. The right guidelines can make the approach suitable for applications and should be also considered in future.

Ex.	0	1	2	3	4	5	6	7	8	9	10
CC	-13,0	**-13,0**	**-10,7**	**-3,5**	-10,9	-6,6	-12,8	-15,2	-15,2	-8,2	-6,3
CL	-14,2	**-9,6**	**-6,8**	-16,0	-23,8	-21,4	-14,9	-16,7	-21,7	-25,1	-24,8
FV2	2,6	**24,2**	23,1	19,2	22,1	24,8	24,7	27,1	27,4	23,7	16,5
JY	8,3	**20,4**	**22,5**	**40,6**	28,3	30,9	30,8	44,6	39,9	15,5	1,7
W	2,2	**16,0**	**17,5**	12,9	11,0	13,9	10,7	7,6	8,1	13,3	12,6

Table 1: Results of experiments Ap1-Ap10 in $1000 USD.

Acknowledgments

This work was supported by grants MŠMT 1M0572 and GAČR 102/08/0567.

References

[1] R. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, 1957.

[2] D.P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, Nashua, US, 2001. 2nd edition.

[3] M. Hauskrecht. Value-function approximations for partially observable markov decision processes. *Journal of Artificial Intelligence Research*, 13:33–94, 2000.

[4] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1-2):99–134, 1998.

[5] M. Kárný, Böhm J., T. V. Guy, L. Jirsa, I. Nagy, P. Nedoma, and L. Tesař. *Optimized Bayesian Dynamic Advising: Theory and Algorithms*. Springer, London, 2005.

[6] W. B. Powell. *Approximate Dynamic Programming*. Wiley-Interscience, 2007.

[7] J. Zeman. Estimating of Bellman function via suboptimal strategies. In *2010 IEEE Int. Conference on Systems, Man and Cybernetics*. IEEE, 2010.