Akademie věd České republiky
Ústav teorie informace a automatizace, v.v.i.

Academy of Sciences of the Czech Republic
Institute of Information Theory and Automation

RESEARCH REPORT

Miroslav Pištěk

Approximate Dynamic Programming based on High Dimensional Model Representation

2310 October 2011

ÚTIA AVČR, v.v.i., P.O.Box 18, 182 08 Prague,
Czech Republic
Fax: (+420)286890378, http://www.utia.cas.cz,
E-mail: utia@utia.cas.cz

Abstract

In this article, an efficient algorithm for an optimal decision strategy approximation is introduced. The proposed approximation of the Bellman equation is based on HDMR technique. This non-parametric function approximation is used not only to reduce memory demands necessary to store Bellman function, but also to allow its fast approximate minimization. On that account, a clear connection between HDMR minimization and discrete optimization is newly established. In each time step of the backward evaluation of the Bellman function, we relax the parameterized discrete minimization subproblem to obtain parameterized trust region problem. We observe that the involved matrix is the same for all parameters owning to the structure of HDMR approximation. We find eigenvalue decomposition of this matrix to solve all trust region problems effectively. The achieved estimates of minima are immediately stored in HDMR approximation to avoid a full-domain representation of Bellman function. We assume that the newly developed approximate minimzation of HDMR can be beneficial also in other applications.

1 Motivation

The main focus of this article is to develop an approximative tool suitable for enlarging the class of computationally feasible decision-making problems. This work copes with the principal problem within the stochastic dynamic programming, which is known as *curse of dimensionality*, see [4]. In the contemporary state of arts, there is a lack of approximative techniques capable of encompassing problems with a larger decision-making horizon.

The aim of this work is to reduce both computational and memory demands necessary to compute and store the optimal strategy. To this end, properties of an approximation tool called High Dimensional Model Representations (thereinafter "HDMR") are promising. It was stimulated by applications in chemistry, see Ref. [5], which focused on reducing enormous memory demands of the involved models. In its background, there stands a simple observation: only low-order correlations amongst the input variables have a significant impact upon the outputs of a typical model.

We shall use the second order HDMR approximation throughout this artice. For general function $g(x)$ it reads

$$\begin{aligned} g(x) \quad &\approx \quad \tilde{g}(x) \equiv \tilde{g}(x^1, x^2, \ldots, x^d) = \\ &\tilde{g}_\emptyset + \sum_{m=1}^{d} \tilde{g}_m(x^m) + \sum_{m=1}^{d-1} \sum_{n=m+1}^{d} \tilde{g}_{mn}(x^m, x^n) \end{aligned} \tag{1}$$

Here, a zero-order component $\tilde{g}_\emptyset$ denotes a constant scalar value over the domain of $g(x)$; the first-order components $\tilde{g}_m(x^m)$ describe an independent effect

of each variable x_m and the second-order component $\tilde{g}_{mn}(x^m, x^n)$ represents the joint effect of the variables x^m and x^n. Experience shows that even this low-order case of HDMR often provides a sufficient description of $g(x)$.

Such a function approximation (representation) yields two main advantages. The first one is the data reduction. The memory space necessary to store all the values of the original function $g(x)$ grows exponentially with the dimension d, whereas the size growth of decomposition components is just quadratic in d. This property helps us to cope with high-dimensional problems of the real world. It was already shown that HDMR is capable to represent strategy on large control horizon [3], however, the linearization of the Bellman equation was necessary.

In this article, we show another advantage of HDMR representation, which is the reduction of computational complexity. The key observation is the fact that the second order HDMR decomposition allows fast upper estimate on minima, which is especially important in the context of the decision making theory. It suggests to develop new algorithm for approximate dynamic programming based directly on the second order HDMR approximation.

The outline of this work is as follows. Section 2 deals with the current state of art in the decision making theory. A central point here is the presentation of the Bellman equation with its notorious difficulties, mainly the problem of a rapidly growing domain of the Bellman function. To cope with this inconvenience, an approximate technique of HDMR is introduced in detail within Section 3. Also, a system of linear equation determining an optimal function approximation is derived here. Its linearity does not match well with the non-linear Bellman equation. On that account, an algorithm for approximate minimization of HDMR function is introduced in Section 4. Combining these approaches, a viable technique for approximate decision making is obtained, see Section 5. Section 6 is devoted to the conclusion.

Throughout this work, a few general conventions are followed. The domain of the variable x is denoted X, $x \in X$, $|X|$ denotes the count of elements of finite set X. Next, x_t is the quantity x at the discrete time instant labeled by $t \in T$. The letter "f" is reserved for conditioned probability density functions, arguments in condition are separated by "|" in the argument list. Knowing $f(x|y)$, we introduce the expected value of the variable x conditioned by y

$$\mathcal{E}\,[\,x\,|\,y\,] \equiv \int_X x\, f(x|y)\, dx$$

For the vector $x \in X$, $X \subset \mathbb{R}^d$, and $m \in M \equiv \{1, \ldots, d\}$, x^m denotes its m-th coordinate. Therefore, it reads $x = (x^1, \ldots, x^d)$. Taking some $N \subset M$, we denote set of complementary indices by $N^c \equiv M \setminus N$. A HDMR approximation of the function $h(x)$ is marked by $\tilde{h}(x)$. For the domain of $h(x)$, $\mathrm{dom}(h)$ is used.

2 Decision Making Theory

Within this section, the classical results are briefly summarized together with their classical troubles. A detailed discussion is to be found in Ref. [2], for example.

The decision-making task consists in selecting the decision-maker's strategy in order to reach decision-maker's aim with respect to the part of the world (so-called system). The decision maker observes or influences the system over the finite decision making horizon $\tau < \infty$. Value $y_t \in Y_t$ provides the decision maker with all knowledge influencing the future behavior of system, thus y_t evolves with time $t \in T \equiv \{1, \ldots, \tau\}$. It includes the current state of system together with other external data observed at time instance t. Furhter, we shall reference y_t simply as state of the system. Analogously, the decisions (actions) are denoted as $a_t \in A_t$. It is the value that can be directly chosen by the decision maker for reaching decision-maker's aims. The strategy is a collection of mappings of the current state $y_{t-1} \in Y_{t-1}$ into the choice of the next decision $a_t \in A_t$, for the optimal strategy we use $\{a_t^{opt}(y_{t-1})\}_{t\in T}$. To formalize a degree of achievements of the decision-maker's aims, a loss function Z is given assigning a loss to each possible system trajectory $(a_1, y_1, \ldots, a_\tau, y_\tau)$. It respect just one rule: the more suitable the trajectory is, the lower loss value it possesses. Here, a less general concept of the additive loss function is used when the losses accumulate with time as

$$Z(a_1, y_1, \ldots, a_\tau, y_\tau) = \sum_{t=1}^{\tau} z_t\,(a_t, y_t) \tag{2}$$

The involved system is completely described in a probabilistic manner by the following collection of pdfs called the outer model of a system

$$\{f(y_t|a_t, y_{t-1})\}_{t\in T} \tag{3}$$

There are many ways how to find these formulae.

Knowing the loss function (2) altogether with the system model (3), the optimal strategy is determined by the Bellman theorem. If we recursively evaluate the Bellman function

$$V_{t-1}(y_{t-1}) = \min_{a_t\in A_t} \mathcal{E}\,[\,z_t(a_t, y_t) + V_t(y_t)\,|\,a_t, y_{t-1}\,] \tag{4}$$

at all times $t \in T$ with the boundary condition $V_\tau \equiv 0$, then the optimal strategy that minimizes the value of the overall expected loss $Z(a_1, y_1, \ldots, a_\tau, y_\tau)$ is defined by simple formula

$$a_t^{opt}(y_{t-1}) \equiv \operatorname*{argmin}_{a_t\in A_t} \mathcal{E}\,[\,z_t(a_t, y_t) + V_t(y_t)\,|\,a_t, y_{t-1}\,] \tag{5}$$

However, the exact calculation of the Bellman function is computationally infeasible in the majority of practical applications [4].

For our purposes, this standard form of Bellman equation is not convenient. It is the whole expresion on the right hand side of (4) which we need to aproximate using HDMR. Thus, we rewrite Bellman equation into the following equivalent form

$$E_t(a_t, y_{t-1}) = \mathcal{E}\left[z_t(y_t, a_t) + \min_{a_{t+1}\in A_{t+1}} E_{t+1}(a_{t+1}, y_t)) \,\middle|\, a_t, y_{t-1} \right] \tag{6}$$

with optimal strategy composed of actions satisfying

$$a_t^{opt}(y_{t-1}) = \operatorname*{argmin}_{a_t\in A_t} E_t(a_t, y_{t-1}) \tag{7}$$

In that setting, the value of $E_t(a_t, y_{t-1})$ is the expected loss of choosing action a_t provided the current state of system is y_{t-1}.

3 High Dimensional Model Representation

This section is to prepare a HDMR approximation technique to reduce memory demands necessary to represent function $E_t(a_t, \sigma_{t-1})$ defined by (6). There are many ways how to construct the decomposition like (1), see Ref. [5]. To reduce this ambiguity, it is necessary to formalize the desired properties of the decomposition.

The function Hilbert space $L^2(X)$ is an useful concept for the function approximation. Generally, it is a space of real functions defined over X with the finite norm $\|g\| \equiv \sqrt{\langle g\,,\, g\rangle}$ inducted by the following scalar product

$$\langle g\,,\, h\rangle_X \equiv \int_X g(x)\, h(x)\, dx \tag{8}$$

Then, the optimal HDMR decomposition $\tilde{g}$ of the function $g \in L^2(X)$ is defined as minimizer of an approximation error evaluated in this norm, i.e., it is a function minimizing $\|g - \tilde{g}\|$. The uniqness of projection on closed subspaces of $L^2(X)$ implies the uniqness of minimizing function $\tilde{g}(x)$ matching this form

$$\tilde{g}(x) \equiv \tilde{g}_\emptyset + \sum_{m=1}^{d} \tilde{g}_m(x^m) + \frac{1}{2}\sum_{m,n=1}^{d} \tilde{g}_{mn}(x^m, x^n) \tag{9}$$

where we introduce symmetrized components $\tilde{g}_{mn}(x^m, x^n) \equiv \tilde{g}_{nm}(x^n, x^m)$ for $n < m$ and zero components $\tilde{g}_{mm} \equiv 0$ into the game to get rid of uncomfortable sumation over $m < n$, recall (1). We shall stick to this convention throughout the whole article.

Let X be d-dimensional product of finite sets X_i

$$X \equiv \prod_{i=1}^{d} X_i \tag{10}$$

and let the integration in (8) be summation over X. Next, for any subset of indices $M \subset \{1, \ldots, d\}$, we denote its complement by $M^c \equiv \{1, \ldots, d\} \setminus M$ and define

$$X_M^{\perp} \equiv \prod_{i \in M^c} X_i \tag{11}$$

Then, the minimizer $\tilde{g}(x)$ has especialy convenient form and the formulae for its decomposition components read

$$\begin{aligned}
\tilde{g}_{\emptyset} &\equiv \frac{1}{|X|} \sum_{y \in X} g(y^1, \ldots, y^d) \qquad (12)\\
\tilde{g}_m(x^m) &\equiv \frac{1}{|X_m^{\perp}|} \sum_{y \in X_m^{\perp}} g(y^1, \ldots, y^{m-1}, x^m, y^{m+1}, \ldots, y^d) - \tilde{g}_{\emptyset}\\
\tilde{g}_{mn}(x^m, x^n) &\equiv \frac{1}{|X_{mn}^{\perp}|} \sum_{y \in X_{mn}^{\perp}} g(y^1, ., x^m, ., x^n, ., y^d) - \tilde{g}_m(x^m) - \tilde{g}_n(x^n) - \tilde{g}_{\emptyset}
\end{aligned}$$

We note that this variant directly corresponds to ANOVA-HDMR in Ref. [5].

The key observation in our aplication of HDMR is its simple construction. Even if $dom(g)$ is too large to operate with all the function values at once, the decomposition components can be constructed point-wise, evaluating and adding values $g(x)$ to proper sums in (12) point by point. This proces is also highly paralelizable.

4 Fast Minimization of HDMR

In the previous section, we show how to find HDMR approximation of $E_t(a_t, y_{t-1})$ at one time step $t \in \{1, \ldots, T\}$. However, to fully approximate equation (6) using HDMR, it is inevitable to find a way of minimizing functions in HDMR decomposition. Owning to the regular structure of this approximation, minimization of approximated function could be even easier then minimization of the original function.

For set of indices $M \subset \{1, \ldots, d\}$ we are interested in point-wise minima of function $\tilde{g}(x)$ in HDMR approximation

$$h(y) \equiv \min_{z \in X_M^{\perp}} \tilde{g}(y, z) \tag{13}$$

for all $y \in \prod_{i \in M} X_i$. The arguments y, z of the minimized function should be reordered to correspond to the original variable x. For the sake of simplicity, however, we shall omit this permutation in the notation. Next, we rewrite the previous equation in a more detailed view recalling (9). Thus, we obtain

$$h(y) = \tilde{g}_\emptyset + \sum_{m \in M} \tilde{g}_m(y^m) + \frac{1}{2} \sum_{m,n \in M} \tilde{g}_{mn}(y^m, y^n) + \min_{z \in X_M^\perp} w_y(z) \tag{14}$$

where we separated the interesting part of calculation by defining function

$$w_y(z) \equiv \sum_{m \in M^c} \tilde{g}_m(z^m) + \frac{1}{2} \sum_{m,n \in M^c} \tilde{g}_{mn}(z^m, z^n) + \sum_{m \in M^c, n \in M} \tilde{g}_{mn}(z^m, y^n) \tag{15}$$

Regardless of a specific choice of y, minimization of this function is equivallent to search for the clicque of minimal weight in complete multipartite edge-weighted graph. To show it, identify different X_m as partities of graph, $z^m \in X_m$ as vertices in particular partite set X_m and $g_{mn}(z^m, z^n)$ as weight of edge between two vertices $z^m \in X_m$, $z^n \in X_n$ taken from distinct partities ($g_{mm} \equiv 0$ as we claimed). Weights of vertices $\tilde{g}_m(x^m)$ together with those inducted by parameter y can be added to weights of corresponding edges. This problem is known to be NP-hard [1]. As it plays a role of repeatedly solved subproblem in our case, it is obvious that we should search only for approximate solution of (13). We introduce an approximative algorithm based on the exact solution of the associated trust region problem which exploits the repetetive nature of this minimization.

4.1 Problem Reformulation

At the moment, however, it is fruitful to rewrite function $w_y(z)$ in a more convenient form. For $i \in \{1, \ldots, |X_m|\}$ we denote $X_m[i]$ the i-th element of finite set X_m, $m \in \{1, \ldots, d\}$. Then, for all $m, n \in M^c$ we define matrices F^{mn} in this way

$$F_{ij}^{mn} \equiv \tilde{g}_{mn}(X_m[i], X_n[j]) \tag{16}$$

In the same manner, we define matrices G^{mn}

$$G_{ij}^{mn} \equiv \tilde{g}_{mn}(X_m[i], X_n[j]) \tag{17}$$

for all $m \in M$ and $n \in M^c$ and vectors $\vec{p}^m$

$$p_i^m \equiv \tilde{g}_m(X_m[i]) \tag{18}$$

for all $m \in M^c$. Then, we represent variables $z^m \in X_m$ by binary vectors $\vec{z}^m \in \{0,1\}^{|X_m|}$ in this way

$$z_i^m = 1 \Longleftrightarrow z^m = X_m[i], \qquad z_i^m = 0 \quad \text{otherwise} \tag{19}$$

Finally, we compose all matrices F^{mn} into one matrix F, matrix G out of matrices G^{mn} and construct vectors $\vec{p}$ and $\vec{z}$ similary. Thus, we obtain a concise reformulation of $w_y(z)$

$$w_{\vec{y}}(\vec{z}) = \frac{1}{2}\,\vec{z}^T F \vec{z} + \vec{y}^T G \vec{z} + \vec{p}^T \vec{z} \tag{20}$$

parameterized by binary vector $\vec{y}$ related to $y \in X_M$ in the same manner as $\vec{z}$ is related to $z \in X_M^{\perp}$. In what follows, we refer to particular element of $\vec{z}$ as z_i^m to stress its meta-vector structure. Then, we implicitly expect $m \in M^c$ and $i \in \{1, \ldots, |X_m|\}$.

Now we abbreviate

$$s \equiv \sum_{m \in M^c} |X_m| \tag{21}$$

and find a better way of formalizing a one-to-one correspondance (19) between $z \in X_M^{\perp}$ and $\vec{z} \in \{0,1\}^s$. We introduce vectors $\vec{e}^m$

$$e_i^m \equiv 1 \tag{22}$$

and rewrite conditions (19) as constraints

$$\vec{e}^m \vec{z}^m = 1 \tag{23}$$

for all $m \in M^c$. These constraints ensure proper minimization domain as we already restricted elements $z_i^m \in \{0,1\}$ for all m and i. Furhter, we define vector $\vec{\epsilon}$

$$\vec{\epsilon} \equiv (1, \ldots, 1) \in \mathbb{R}^{|M^c|} \tag{24}$$

and introduce matrix E composed of vectors $\vec{e}^m$ extended by zeros in such a way that it permits to rearrange constraints (23) as

$$E\vec{z} = \vec{\epsilon} \tag{25}$$

Then we rewrite the mimimization task in (14) into an equivalent form

$$\min_{\vec{z} \in \{0,1\}^s, E\vec{z} = \vec{\epsilon}} \left\{ \frac{1}{2}\,\vec{z}^T F \vec{z} + (G\vec{y} + \vec{p})^T \vec{z} \right\} \tag{26}$$

which is still NP-hard problem. On that account, we introduce a computationally feasible algorithm to get estimates on minima in (26) for all relevant values of parameter $\vec{y}$.

4.2 Trust Region Based Minimization

To prosper out of this repetitive nature of our problem, we observe that the involved matrix F does not depend on the value of parameter and thus we can afford some intiensive matrix preprocessing. This fact suggests to turn our attention to the trust region problem. For symmetric matrix A and vectors $\vec{x}$, $\vec{b}$ of proper dimension it can be formulated as follows

$$\min_{\|\vec{x}\|^2=1} \left\{ \frac{1}{2}\, \vec{x}^T A \vec{x} + b^T \vec{x} \right\} \tag{27}$$

This classical optimization problem is amenable to fast solution even though it is non-convex in general. From vide spectra of developed methods, we choose the one which is computetionaly expensive for an one step minimization, but which will pay out in our repetetive setting. If we find ortoghonal matrix U such that $A = U^T D U$ holds with D diagonal, the minimizer $\vec{x}^\star$ of (27) reads

$$\vec{x}^\star = -U^T (D - \lambda\, \mathbb{I})^{-1} U \vec{b} \tag{28}$$

where $\lambda \in \mathbb{R}$ is a solution of one-dimensional equation

$$\sum_i \left(\frac{q_i}{D_{ii} - \lambda} \right)^2 = 1 \tag{29}$$

where $\vec{q} \equiv U\vec{b}$ and $\lambda < \min_i(D_{ii})$. It can be shown that precisely one such λ exists and it can be found by Newtons' method, for instance, see Ref. [6].

Therefore, we introduce relaxation of the minimization subproblem (26) matching the form of the trust region problem (27). We hold constraint $E\vec{z} = \vec{\epsilon}$ and search for relaxation of the other constraint $\vec{z} \in \{0,1\}^s$. Not to violate the first constraint, we project the minimized criteria in (26) onto the subspace ortoghonal to E. The projection matrix reads

$$P = \mathbb{I} - E^T (E E^T)^{-1} E \tag{30}$$

where the involved inversion exist as EE^T is diagonal matrix with positive diagonal. It appears here because we did not normalize vectors $\vec{e}^m$ in (23). We denote $\vec{c}$ the center of minimization domain composed of subvectors $\vec{c}^m$

$$c_i^m \equiv \frac{1}{|X_m|} \tag{31}$$

and we decompose vector $\vec{z}$ into two parts,

$$\vec{z} = \vec{c} + \vec{u} \tag{32}$$

where $P\vec{u} = \vec{u}$, $P\vec{c} = 0$ and $E\vec{c} = \vec{\epsilon}$ by definition. Next, we substitute these identites into (26) and omit terms constant in $\vec{u}$ as we are interested only in the argument of minima. Thus, we obtain the following minimization problem

$$\min_{\|\vec{u}\|^2=R^2} \left\{ \frac{1}{2}\vec{u}^T P^T F P \vec{u} + \vec{c}^T F P \vec{u} + (G\vec{y} + \vec{p})^T P \vec{u} \right\} \tag{33}$$

where the only question left is to adjust the diameter R properly. By the symmetry of our problem, we can simply choose any vector $\vec{u} \in \{0,1\}^s$ satisfying $E\vec{u} = \vec{\epsilon}$, and prescribe

$$R^2 = \|\vec{c} - \vec{u}\|^2 = \sum_{m \in M^c} \left\{ \left(1 - \frac{1}{|X_m|}\right)^2 + \sum_{i=2}^{|X_m|} \frac{1}{|X_m|^2} \right\} = \sum_{m \in M^c} \left(1 - \frac{1}{|X_m|}\right) \tag{34}$$

Owning to this choice of R, the optimization problem (33) is a proper relaxation of our original problem (26). What is more, it is amenable to fast solution in direct analogy to (27). In a detail, we diagonalize the involved symmetric matrix $P^T F P$, and for a particular choice of $\vec{y}$ we find exact minimizer $\vec{u}^\star$ according to equations (28) and (29). Once solved, we have $\vec{z}^\star = \vec{c} + P\vec{u}^\star$ which implicitly satisfy the constraint $E\vec{z} = \vec{\epsilon}$.

In general, such $\vec{z}^\star$ does not correspond to any feasible solution of (26). Nontheless, it is quite informative - the higher the value of $\vec{z}_i^{\star,m}$ is, the lower cirteria we should expect when adjusting m-th coordinate of z to $X_m[i] \in X_m$. One can came up with many ways of "rounding" of $\vec{z}^\star$ to some feasible solution of unrelaxed problem, however, there is not any guarantee that a particular heuristics is the best possible. Here, we reduce the size of minimization domain in (14) using the following heuristics. At first, we find k highest elements of each $\vec{z}^{\star,m}$ and denote their indices by n_i^m where $i \in \{1, \ldots, k\}$. Then, we define

$$Z_m \equiv \{X_m[n_i^m] | i \in \{1, \ldots, k\}\} \subset X_m \tag{35}$$

for each $m \in M^c$, abbreviate $Z_M^\perp \equiv \prod_{m \in M^c} Z_m$ and see that $Z_M^\perp \subset X_M^\perp$ and $|Z_M^\perp| = k^{|M^c|}$. Next, we substitute $Z_M^\perp$ into (14) instead of $X_M^\perp$ and solve this minimization directly by enumeration as the domain is much smaller then in the original case. This way we obtain an upper bound on minima in (14) for each particular choice of $y \in X_{M^c}^\perp$. We admit that we employed here the first idea we had, we feel that wide spectra of better heuristics can by applied to "round" $\vec{z}^\star$. We postpone this question to future research.

4.3 Numerical Experiment

We prepared a numerical experiment with simulated data. At first, we solved problem (26) exactly for 20 different values of parameter $\vec{y}$ with randomly generated matrices F, G of a proper structure and zero vector $\vec{p} = 0$. Elements

of F, respective G, were chosen from a uniform distribution on the interval $[0, 100]$, respective $[0, 1]$, and the dimensions of F and G correspond to minimization of 6 dimensional function $w_y(z)$ over $z \in X^2$ where $|X| = 200$ parameterized by $y \in Y^4$ where $|Y| = 20$. Then, we utilize the method introduced in the previous subsection to find an estimate on minima. We examinated the quality of estimates for four different choices of parameter k. The results are summarized in Figure 1. They correspond to our expectation. The higher

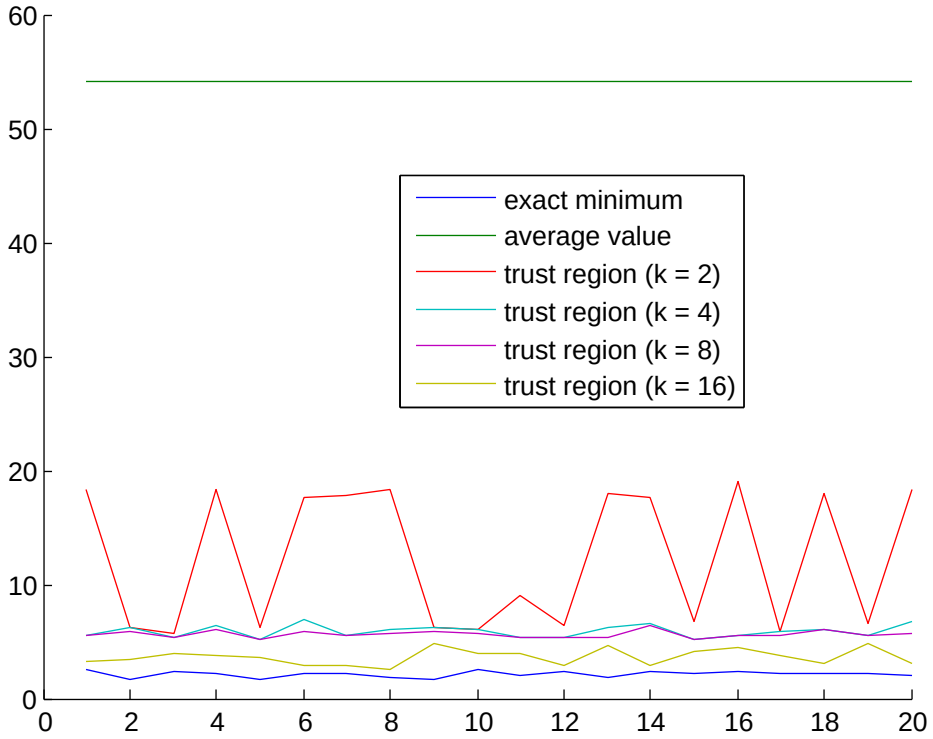

Figure 1: Upper bounds on minima obtained by approximative minimization for 20 different values of y and various values of parameter k. They are compared to the average values of the minimized functions and the values of their exact minima.

the value of k is, the better approximation we observe. On the other hand, it obviously takes more computational time to obtain these more accurate upper estimates. To see this relationship more clearly, we propose Table 1 where the average computational times of all these methods are compared to the average computational time of the exact minimization.

average results	trust region			
	k=2	k=4	k=8	k=16
computational time (in %)	3.3%	3.3%	3.5%	4.0%
distance from minima	0.19	0.07	0.06	0.03

Table 1: The average computational times and average distances from the exact minima for various values of k. Computational times are stated relatively to the computational time of the exact minimization and the distances from minima are also normalized. They are rescaled and shifted in such a way that exact minima correspond to 0 whereas the avreage value of the minimized criteria correspod to 1. The averages were taken over 20 different values of y.

5 Approximate DP based on HDMR

At the moment, we are prepared to apply both HDMR approximation developed in Section 3 and fast approximative minimization of HDMR function from Section 4 to effectively approximate equation (6). The proposed algorithm is recursive and runs in the backward manner as the evaluation of the exact Bellman equation. We denote the approximated function as $\tilde{E}_t$ even though it is not, in fact, the HDMR approximation of E_t (with an exception of $t = T$).

For the first step, $t = T$, we have

$$E_T(a_T, y_{T-1}) = \mathcal{E}\left[\, z_T(y_T, a_T) \,|\, a_T, y_{T-1} \,\right] \tag{36}$$

To obtain all components $\tilde{E}_{T,\emptyset}, \tilde{E}_{T,m}, \tilde{E}_{T,mn}$ of its HDMR approximation, we take all pairs $(a_T, y_{T-1}) \in A_T \times Y_{T-1}$, evaluate $E_T(a_T, y_{T-1})$ for each pair and add the result to proper sums in (12). Next, supose we know all $\tilde{E}_{t+1,\emptyset}$, $\tilde{E}_{t+1,m}$, $\tilde{E}_{t+1,mn}$ and we want to find the HDMR approximation of E_t. Substituting $\tilde{E}_{t+1}$ into (6) we have

$$E_t(a_t, y_{t-1}) \approx \mathcal{E}\left[z_t(y_t, a_t) + \min_{a_{t+1} \in A_{t+1}} \tilde{E}_{t+1}(a_{t+1}, y_t)) \,\middle|\, a_t, y_{t-1} \right] \tag{37}$$

and thus we define $\tilde{E}_{t+1}$ as the HDMR approximation of the expression on the right-hand side.

To find it, we have to evaluate this function at all points again. If we denote

$$h_t(y_t) \equiv \min_{a_{t+1} \in A_{t+1}} \tilde{E}_{t+1}(a_{t+1}, y_t) \tag{38}$$

we can find its upper bound following the instructions of Section 4. Looking at (13), we indentify $\tilde{g} = \tilde{E}_{t+1}$ and $X_M^{\perp} = A_{t+1}$. Based on the knowledge of $\tilde{E}_{t+1,\emptyset}$, $\tilde{E}_{t+1,m}$ and $\tilde{E}_{t+1,mn}$, we construct matrices F_t, G_t and vector $\vec{p}_t$ according to

(16), (17) and (18). Now we have (38) in the form of (26) and we search for the upper bound on minima according to Section 4.2. We relax the problem into the form of (33) and find its exact minimizer $\vec{h}_t^\star(y_t)$ for all $y_t \in Y_t$ in a direct analogy to (28) and (29). We stress again that the involved diagonalization of the matrix $P_t^T F_t P_t$ is carried out just once for each time step t and then used for all $y_t \in Y_t$. Based on the knowledge of $\vec{h}_t^\star(y_t)$, we find an upper bound $\overline{h}_t(y_t)$ on unrelaxed problem (38) following the last paragraph of Section 4.2. Knowing $\overline{h}_t(y_t)$, we continue similarly to the procedure applied for $t = T$. We sequentialy take all pairs $(a_t, y_{t-1}) \in A_T \times Y_{T-1}$, evaluate function

$$\mathcal{E}\left[\, z_t(y_t, a_t) + \overline{h}_t(y_t) \,\middle|\, a_t, y_{t-1} \,\right] \tag{39}$$

for each pair and add the result to proper sums according to (12).

This way, we construct all components $\tilde{E}_{t,\emptyset}$, $\tilde{E}_{t,m}$ and $\tilde{E}_{t,mn}$ in a point-wise manner avoiding the full dimensional representation of $\tilde{E}_t$. We repeat this step to recursively compute $\tilde{E}_t$ for all $t \in \{1, \ldots, T\}$ and finally, we follow equation (7) to derive the approximated optimal strategy.

6 Conclusion

The aim of this work was to cope with both computational and memory demands necessary to find and represent the optimal decision making strategy. The proposed variant of approximate dynamic programming based on HDMR approximation is appealing for two reasons. At first, this approximation reduces memory demands considerably. There are, however, many other possible approximations of the Bellman equation. Thus, the crucial advantage of this particular one is that HDMR permits a fast approximative minimization, an inevitable ingredient of an algorithm attempting to well approximate the Bellman equation over longer time horizon.

The bottle-neck of this approximation technique is the fact that it still needs to pass through the whole decision tree. Either it can be parallelized easily, or some reinforcement learnign algorithm that aims at this problem can be included. The fact that HDMR approximation permits a fast approximative minimization would still be worthwhile.

What is definitely missing in this article is an experimental verification of the proposed algorithm. It is left to future research, however, the algorithm should be more precise then the one developed in [3]. Therein, the average value of an approximated function was used as an upper bound on minimum. If we look at Figure 1, we see that the upper bound on minimum developed here is more precise then the mere average of the criteria. Thus we may conclude that this algorithm should do better even in a real-world application.

The author would like to express his acknowledgement to RNDr. Ondřej Pangrác, Ph.D., for inspiring discussion about the link between HDMR minimization and discrete optimization and also to the project GAČR P 102/11/0437 for support.

References

[1] Richard M. Karp. Reducibility among combinatorial problems. *Miller, R. E.; Thatcher, J. W., Complexity of Computer Computations, New York: Plenum*, 1972.

[2] H. Kushner. *Introduction to Stochastic Control*. Holt, Rinehart and Winston, New York, 1970.

[3] Miroslav Pistek. On implicit approximation of the bellman equation. *15th IFAC Symposium on System Identification, Saint-Malo, France*, 2009.

[4] Warren B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Wiley-Interscience, 2007.

[5] H.J. Rabitz and O.F. Alis. General foundations of high-dimensional model representations. *Journal of Mathematical Chemistry*, 25:197–233, 1999.

[6] D. C. Sorensen. Newton's method with a model trust region modification. *SIAM J. Numer. Anal.*, 19(2), 1982.