Estimating of Bellman function via suboptimal strategies

Jan Zeman
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Prague, Czech Republic
janzeman3@seznam.cz

***Abstract*—The paper concerns approximate dynamic decision making. It deals with solving Bellman equation to obtain the Bellman function via so-called suboptimal strategies. The suboptimal strategies are strategies reflecting the revision of the realized past decisions. The revision is conditioned on availability of future knowledge. Suboptimal strategies help to transform the estimation of Bellman function to solving system of algebraic equations, when parametrized form of Bellman function is used. The presented approach is applied to futures trading task.**

***Index Terms*—Bellman function, optimization, dynamic decision making**

I. Introduction

Decision making is present in any area of the human interest since the majority of real-life problems can be regarded as a selection among several alternatives under uncertainty. Though decision making has been deeply investigated by many sciences (games, social sciences, etc.), there is no universal ready-to-use approach. The main difficulty is to find a sequence of decisions optimizing some criterion while ensuring the best long-term performance under incomplete knowledge and uncertainty.

Dynamic programming (DP) is known to be an effective approach to solve this complex task in a recursive manner [1], [2]. However DP suffers from the computational complexity known as curse-of-dimensionality [3], which makes conventional DP solution infeasible. A number of approximate solutions (mostly suggested within computer science) has been developed, however none of them can be considered for decision making.

The majority of approximation techniques came from machine learning and artificial intelligence fields. This has been partially motivated by a conceptual similarity of reinforcement learning [4], [5] and dynamic programming. The approximation approaches to DP can be formally divided into:
Model-based approaches, which assume knowledge of the system model and focus on efficient solving Bellman equation. These methods include for example indirect approximation of Bellman function [6]; real-time dynamic programming [7].
Model-free methods, which do not assume availability of the system model. They include for example temporal difference method (TDM) [8], Q-learning [9].

The mentioned approximate DP techniques have been applied to different application areas. The most notable success has been achieved in the applications, where environment does not change (or changes very slowly) and there is a possibility to collect rich amount of data [10]–[12]. To guarantee the reliable results, the convergence properties of some of the suggested approaches (e.g. TDM) should be investigated in advance. Besides, the approximation techniques can also suffer from of the curse-of-dimensionality and the risk should be eliminated. To summarize, there is no universal, ready-to-use approximate DP approach, which can cope with large state- and action-spaces, incomplete knowledge and uncertainty.

The objective of the research is to find an approximate DP technique that satisfies the above-mentioned requirements and can be applied to decision making. The proposed approximation has been motivated by DM task arisen in economic analysis and futures contract trading but the solution developed is of general nature and can be of interest for other applications.

Section II recalls the decision-making task using dynamic programming. Section III introduces the suboptimal strategies and details of their usage for the approximation. Main algorithmic aspects of the proposed approach are described by Section IV. The preliminary results of the approach obtained on the future trading data are described in Section V. Section VI summarizes the main features of the approach and open questions remained.

II. Decision-making task

The decision-making task assumes a *decision maker* and a *system*. The system is a part of the world, which is of interest for the decision maker.

The decision maker has own goal with respect to the system expressed in the form of a *gain function* G_τ^T, which quantifies the degree of reaching the goal on the time interval $\{\tau, \dots, T\}$. The data available to decision maker consists of observations made on the system y_t and its decisions u_t, where t is discrete time, $t \in t^* = \{1, 2, \dots, T\}$. Using the *knowledge* $\mathcal{P}_t = (y_1, \dots, y_t, u_1, \dots, u_{t-1})$ of the past system outputs and previous decisions, the decision maker designs a decision $u_t \in u_t^*$, where u_t^* is a set of admissible decisions.

At time t, the decision maker designs the strategy to maximize the gain G_t^T, which depends on the system output over the whole decision period, even on the unknown output $(y_{t+1}, \dots, y_T)$. The information available to the decision maker at time t is $\mathcal{P}_t$. Therefore the decision maker is forced to optimize the *expected value* $\mathcal{E}$ of the gain:

$$\mathcal{V}_t(\mathcal{P}_t) = \max_{u_t,\dots,u_T} \mathcal{E}[G_t^T|\mathcal{P}_t],$$

which defines Bellman function $\mathcal{V}_t(\mathcal{P}_t)$.

The assumption of an additive gain function

$$G_{t_1}^{t_2} = G_{t_1}^{t} + G_{t+1}^{t_2} \quad \text{for any } t, t_1, t_2 \in \{1, 2, \dots, T\}$$

and the optimality principle [1] allow us rewrite Bellman function in the recursive shape:

$$\mathcal{V}_t(\mathcal{P}_t) \;=\; \max_{u_t,\dots,u_{t+h}} \mathcal{E}[G_t^{t+h} + \mathcal{V}_{t+h+1}(\mathcal{P}_{t+h+1})|\mathcal{P}_t], \quad (1)$$

where the maximizing arguments $u_t, \dots, u_{t+h}$ are the proposed decision rules and h is a constant, which allows the design of multi-step decision.

For the horizon T growing to infinity, Bellman function converges to stationary form [2]:

$$\mathcal{V}(\mathcal{P}_t) = \max_{u_t,\dots,u_{t+h}} \mathcal{E}\left[G_t^{t+h} + \mathcal{V}(\mathcal{P}_{t+h+1})|\mathcal{P}_t\right]. \quad (2)$$

The decision u_t is viewed as a decision rule dependent on the available knowledge $\mathcal{P}_t$, $u_t = u_t(\mathcal{P}_t)$. We differ the notation from here onward: A decision is denoted by letter u_t, a decision based on decision rule is denoted as a function with the argument $u_t(\mathcal{P}_t)$ and a decision rule is denoted as a function without an argument $u_t(.)$.

The maximization over rules (2) can be approximated by maximization over values of decisions (see [13] for details):

$$\mathcal{V}(\mathcal{P}_t) = \max_{u_t,\dots,u_{t+h}} \mathcal{E}\left[G_t^{t+h} + \mathcal{V}(\mathcal{P}_{t+h+1})|\mathcal{P}_t, u_t, \dots, u_{t+h}\right]. \quad (3)$$

III. Estimating Bellman function

Bellman equation (3) characterizes the approach for designing the decision u_t. The right-hand side of equation consists of two terms: the h-step gain and the Bellman function $\mathcal{V}(\mathcal{P}_{t+h+1})$. The suitable interpretation is that h-step gain characterizes reaching the short-term aims, whereas Bellman function characterizes the long-term aims. Hence, it is important to consider the expected value of both terms. This paper does not deal with the expected value of h-step gain, but the paper focuses on expressing the expected value of Bellman function.

Most of approaches estimates Bellman function using following update rule (so-called value iteration):

$$\hat{\mathcal{V}}_t(\mathcal{P}_t) \;=\; \max_{u_t} \mathcal{E}[G_t^t + \hat{\mathcal{V}}_{t-1}(\mathcal{P}_{t+1})|\mathcal{P}_t, u_t],$$

where $\hat{\mathcal{V}}_t(.)$ is estimation of Bellman function obtained at time t. The convergence of the approach is proved in [1], hence most of the approaches continue and expand this branch.

We try to extract Bellman function directly from Bellman equation by solving the following system of equations:

$$\begin{aligned}
\mathcal{V}(\mathcal{P}_1) &= \max_{u_1,\dots,u_{1+h}} \mathcal{E}[G_1^{1+h} + \mathcal{V}(\mathcal{P}_{h+2}) \\
&\qquad\qquad |\mathcal{P}_1, u_1, \dots, u_{1+h}], \\
\mathcal{V}(\mathcal{P}_2) &= \max_{u_2,\dots,u_{2+h}} \mathcal{E}[G_2^{2+h} + \mathcal{V}(\mathcal{P}_{h+3}) \\
&\qquad\qquad |\mathcal{P}_2, u_2, \dots, u_{2+h}], \quad (4) \\
&\vdots \\
\mathcal{V}(\mathcal{P}_t) &= \max_{u_t,\dots,u_{t+h}} \mathcal{E}[G_t^{t+h} + \mathcal{V}(\mathcal{P}_{t+h+1}) \\
&\qquad\qquad |\mathcal{P}_t, u_t, \dots, u_{t+h}].
\end{aligned}$$

The system consists of Bellman equation (3) indexed by $1, \dots, t$. On the authors best knowledge, there is no branch of research trying find the Bellman function by solving the system (4).

The system has two main disadvantages: It is the *functional* system. Each equation contains *maximum* function, which makes computation complex.

The following sections show how to transform the system to system of algebraic equation. The main contribution of this paper is in work off the maximum.

A. Suboptimal strategies

This section defines the suboptimal strategies and prepares the notation for the solving the system (4). The suboptimal strategies are important for the work off the maximum from (4).

The decision maker designs and applies a sequence of *admissible decisions*, where each decision u_t is based on the maximal knowledge $\mathcal{P}_t$ available at time t. A sequence of admissible decisions is called *admissible strategy* and is denoted:

$$U_t^a = (u_1(\mathcal{P}_1), u_2(\mathcal{P}_2), \dots, u_t(\mathcal{P}_t)). \quad (5)$$

Let us consider an unrealistic strategy focused on decisions designed as if we had known the future consequences of these decisions. Let us consider, at time t, the *suboptimal strategy* $U_t(\mathcal{P}_t)$ as a revision of the past decisions $u_1, u_2, \dots, u_t$ based on the knowledge $\mathcal{P}_t$ available at time t, i.e.

$$U_t(\mathcal{P}_t) = (u_1(\mathcal{P}_t), u_2(\mathcal{P}_t), \dots u_t(\mathcal{P}_t)). \quad (6)$$

This revision is done at time t and reflects the changes in decisions caused by growing knowledge. An analogy with the human revising is: 'If I had known the today's information, I would have do anything else yesterday.'

The suboptimal strategy can be designed at each time instant. Therefore the following sequence of suboptimal strategies can be available at time t:

$$\begin{aligned}
U_1(\mathcal{P}_1) &= (u_1(\mathcal{P}_1)), \\
U_2(\mathcal{P}_2) &= (u_1(\mathcal{P}_2), u_2(\mathcal{P}_2)), \\
U_3(\mathcal{P}_3) &= (u_1(\mathcal{P}_3), u_2(\mathcal{P}_3), u_3(\mathcal{P}_3)), \\
&\vdots \\
U_t(\mathcal{P}_t) &= (u_1(\mathcal{P}_t), u_2(\mathcal{P}_t), u_3(\mathcal{P}_t) \dots u_t(\mathcal{P}_t)).
\end{aligned}$$

The admissible strategies converges to the *optimal strategy* with growing t, i.e. each decision rule $u_i(.)$ tends to optimal decision rule $u^O(.)$ (details can be find in [2]). Consequently, the suboptimal strategies tends also to the optimal strategy:

$$\begin{array}{rcl} U_t(\mathcal{P}_t) &=& (u_1(\mathcal{P}_t), u_2(\mathcal{P}_t), u_3(\mathcal{P}_t), \ldots), \\ &\downarrow& \text{for } t \to +\infty \\ U^O(\mathcal{P}_\infty) &=& (u_1^O, u_2^O, u_3^O, \ldots), \end{array}$$

where $u_1^O, u_2^O, u_3^O, \ldots$ denote the realization of optimal strategy.

B. Assumption on convergence

We assume that there exists time t_0 such that for each $t \geq t_0$, there exists $i \in \mathcal{N}, i < t$ such that the first i decisions have reached their optimal values, i.e.

$$\begin{array}{rcl} u_1(\mathcal{P}_t) &=& u_1^O, \\ u_2(\mathcal{P}_t) &=& u_2^O, \\ &\vdots& \\ u_i(\mathcal{P}_t) &=& u_i^O. \end{array}$$

An intuitive motivation for this special convergence is that there exists time t_0 such that any knowledge gained later cannot improve the previous part of the strategy.

This convergence is a generalization of the suboptimal strategies behavior found on the trading task (see Section V). The assumption is crucial for further steps.

C. Similarity indexes

This section defines two indexes, characterizing the degree of the convergence in a sense defined in Sec. III-B. The indexes are obtained by comparing a subsequence $U_t(\mathcal{P}_t)$ and the optimal one $U_t^O(\mathcal{P}_\infty)$.

Similarity index S_t is a number of identical elements in $U_t(\mathcal{P}_t)$ and $U^O(\mathcal{P}_\infty)$.

$$S_t = \sum_{i=1}^{t} \delta(u_i(\mathcal{P}_t), u_i^O), \qquad (7)$$

where $\delta(x, y) = 1$ for $x = y$ and $\delta(x, y) = 0$ for $x \neq y$.

Strict similarity index s_t is the maximal length of the identical part of strategies starting from the first element:

$$s_t = \max_i \{i; \forall j \leq i, u_j(\mathcal{P}_t) = u_j^O\}. \qquad (8)$$

The definitions of S_t and s_t imply $s_t \leq S_t \leq t$. Both similarity indexes show a degree of convergence $U_t(\mathcal{P}_t)$ to the optimal one $U_t^O(\mathcal{P}_\infty)$. Similarity index S_t shows the total degree, whereas strict one s_t shows the degree of converged continuous part.

Example: To illustrate the introduced notions, let us consider the following suboptimal $U_t(\mathcal{P}_t)$ and the optimal $U_t^O(\mathcal{P}_\infty)$ strategies:

$$\begin{array}{ll} U_t(\mathcal{P}_t) & = \{\, 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \,\}, \\ U_t^O(\mathcal{P}_\infty) & = \{\, 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \,\}. \end{array}$$

The strategies have 4 elements identical, the fifth element differs, the sixth and seventh elements are identical and then strategies differ. The similarity index $S_t = 6$, because there are 6 identical elements in the strategies. The strict similarity index $s_t = 4$, because the fourth element is the last element, before the first difference occurs.

D. Bellman equation and similarity indexes

Let us focus back to the solving the system (4), which can be written in short form:

$$\begin{array}{rcl} \mathcal{V}(\mathcal{P}_k) &=& \displaystyle\max_{u_k, \ldots, u_{k+h}} \mathcal{E}[G_k^{k+h} + \mathcal{V}(\mathcal{P}_{k+h+1}) \\ && \qquad\qquad |\mathcal{P}_k, u_k, \ldots, u_{k+h}] \\ && \text{for } k \in \{1, \ldots, t\}. \end{array}$$

Let us extend the used knowledge $\mathcal{P}_t$ by optimal strategy $U^O(\mathcal{P}_\infty) = (u_1^O, \ldots, u_{t+h}^O, \ldots)$ to obtain extended knowledge: $\mathcal{P}_t^e = \mathcal{P}_t \cup U^O$, and search Bellman function for the extended knowledge, i.e. Bellman function is searched within the class of function dependent on optimal actions.

Then, the maximum in equations of (4) is reached for the optimal strategy $U^O(\mathcal{P}_\infty) = (u_1^O, \ldots, u_{t+h}^O, \ldots)$. The inserting the optimal strategy to system (4), we obtain:

$$\begin{array}{rcl} \mathcal{V}(\mathcal{P}_k^e) &=& \mathcal{E}[G_k^{k+h} + \mathcal{V}(\mathcal{P}_{k+h+1}^e) | \mathcal{P}_k, u_k^O, \ldots, u_{k+h}^O] \\ && \text{for } k \in \{1, \ldots, t\}. \end{array} \qquad (9)$$

Under the assumption (Sec. III-B), s_t elements of the optimal strategy $U^O(\mathcal{P}_\infty)$ are known at the time t and equal to the realization of suboptimal strategy $(u_1^O, \ldots, u_{s_t}^O) = (u_1(\mathcal{P}_t), \ldots, u_{s_t}(\mathcal{P}_t))$. Hence, the suboptimal decisions can be inserted into (9):

$$\begin{array}{rcl} \mathcal{V}(\mathcal{P}_k^e) &=& \mathcal{E}[G_k^{k+h} + \mathcal{V}(\mathcal{P}_{k+h+1}^e) \\ && \qquad |\mathcal{P}_k, u_k(\mathcal{P}_t), \ldots, u_{k+h}(\mathcal{P}_t)] \\ && \text{for } k \in \{1, \ldots, s_t - h\}. \end{array} \qquad (10)$$

The step from (9) to (10) explains the importance of index s_t, because the number of equations available to solve the system (4) descends from t to $s_t - h$. Hence, the applicability of the designed approach is closely related to behavior s_t according the time t (possible relations are described in further Sec. III-F).

We worked off the maximum from (4) and obtained the system of functional equations (10), which should be solved to obtain Bellman function. The optimization task is replaced by solution of the system of functional equations. The following section describes its transforming to system of algebraic equations.

The previous steps generalize the approach in [14]. The remaining open question is difference between $\mathcal{V}(\mathcal{P}_k^e)$ and $\mathcal{V}(\mathcal{P}_k)$.

E. Parametrized form of Bellman function

A parametrized form of Bellman function should be chosen to obtain the suitable approximation of Bellman equation (10), otherwise we cannot solve it. The parametrized form typically characterizes a class of functions, which should be form invariant relative to recursion (10), i.e. the inserting the form into the right hand side of (10) reproduces the form on the left one.

Let us select a finite-parametrized form of Bellman function:

$$\mathcal{V}(\mathcal{P}_t^e) \approx V(\mathcal{P}_t^e; \Theta), \tag{11}$$

where $\Theta \in \Theta^*$ is a vector of unknown parameters. Then, inserting (11) into the system (10), one can write:

$$\begin{aligned} V(\mathcal{P}_k^e; \Theta) + \kappa_k &= \mathcal{E}[G_k^{k+h} + V(\mathcal{P}_{k+h+1}^e; \Theta) \\ &\qquad |u_k(\mathcal{P}_t), \ldots, u_{k+h}(\mathcal{P}_t)] \\ &\text{for } k \in \{1, \ldots, s_t - h\} \end{aligned} \tag{12}$$

where κ_k is an error caused by approximation (11).

The system of functional equations (10) is reduced to the system of algebraic equations (12). Thus, the solution of Bellman equation converges to an estimation of the parameters Θ.

F. Limitation of the approach

The success of the presented approach depends on the behavior of the strict similarity index s_t, which influences the number of equations in the systems (10) and (12). The number of equations available in (10) and (12) indicates the available knowledge about Bellman function and its growing leads to improvement of the estimation. The number of equations at the time t is equal to $(s_t - h)$, hence the increase of s_t means growing of the knowledge.

According to behavior of the similarity indexes, we can expect the following types of task:

- Task with a strong similarity - is a task, when s_t and S_t grow with the time t and indexes are close to time t: the number of equations in (10), (12) also grows with t. This type of task yields the best condition for the estimation of Bellman function via solution (10), (12).
- General task without a similarity - is opposite to the previous type, when s_t and S_t are small constants independent of t. In this case, the system (10) has a small number of equations and the number does not grow. This are the worst conditions to estimate Bellman function by solving (10), (12), because there could not be enough equations to find a solution.
  In this case, it is better to use different design of Bellman function. However even the available "poor" system of equations can be used as a prior information about Bellman function.
- Task with a weak similarity - is between the previous two extreme types: the strict similarity index s_t is a small constant or growing only by jumps, but S_t grows with t and is close to t. In this case, Bellman function can be estimated by solving (10), (12), but the similarity index S_t should be used instead of s_t in definition of equation systems, i.e. $k \in \{1, \ldots, S_t - h\}$. This redefinition causes that the systems (10) or (12) contain the invalid equations based on non-optimal decisions. It depends on the solved problem, whether the solution of systems (10) and (12) provide satisfactory estimation of Bellman function or not.

IV. Algorithmic aspects

A lot of issues emerges, when the theoretical solutions are applied in the practice. This section introduces the most important aspects and shows the algorithms solving them.

The algorithms presented here as well as the introduced assumptions are partially derived from the properties and behavior observed on the futures trading task (see Sec. V).

A. Design of suboptimal strategies

The suboptimal strategies are designed using already known data. Hence, the the expected value can be omitted and the planning horizon is set to zero $h = 0$:

$$\begin{aligned} \mathcal{V}_k(\mathcal{P}_k) &= \max_{u_k}(G_k^k + \mathcal{V}_{k+1}(\mathcal{P}_{k+1})), \\ &\text{for } k \in \{t, t-1, \ldots, 1\}. \end{aligned} \tag{13}$$

The recursion starts from:

$$\mathcal{V}_{t+1}(\mathcal{P}_{t+1}) = 0. \tag{14}$$

The hidden problem of the approach based on (13) and (14) is the dependence of the decision u_i, $i \in \{1, 2, \ldots, t\}$, on the previous decisions i.e. $u_i = f(u_1, u_2, \ldots, u_{i-1})$. Hence, the design of the suboptimal strategy consists of two steps:

1) Design of Bellman functions: is done using the backward recursion (13) starting from the initial condition (14). Due to the mentioned dependence of a decision on the previous one, Bellman function cannot be calculated value-wise (i.e. inserting all known values from $\mathcal{P}_t$). Moreover, it must be distinguished, which values from $\mathcal{P}_t = (y_1, \ldots, y_t, u_1, \ldots, u_{t-1})$ can be simply inserted and which must be considered and served as variables. Due to the fact that $u_1, u_2, \ldots, u_t$ are designed, all decisions must be considered as variables, whereas the system output $y_1, \ldots, y_t$ can be simply inserted. Hence, the following sequence is generated:

$$\mathcal{V}_t(\mathcal{P}_t) = b_t(u_1, u_2, \ldots, u_{t-1}), \tag{15}$$

$$\vdots \tag{16}$$

$$\mathcal{V}_3(\mathcal{P}_t) = b_3(u_1, u_2), \tag{17}$$

$$\mathcal{V}_2(\mathcal{P}_t) = b_2(u_1), \tag{18}$$

$$\mathcal{V}_1(\mathcal{P}_t) = b_1, \tag{19}$$

where the functions $b_1(.), b_2(.), \ldots, b_t(.)$ are functions of decisions u_i, $i \in \{1, 2, \ldots, t-1\}$ for the given y_i originating from Bellman function.

This fact can be interpreted as demonstration of curse of dimensionality (see [3]), when the dimension of problem (i.e. memory requirements) grows with the time t.

2) Inserting: of decisions follows the design of Bellman function. When the backward recursion from the previous step reaches $\mathcal{V}_1(\mathcal{P}_1) = b_1$ (19), the first decision u_1 can be generated. Then, u_1 is inserted into the stored function $\mathcal{V}_2(\mathcal{P}_t) = b_2(u_1)$ (18) and decision u_2 is generated, and so on.

The two-step algorithm is given by general case and can be omitted or easier in special problems such the considered trading task.

Assumption: The realization of the algorithm is possible only under the assumption of the open loop, i.e. the decisions cannot influence the system.

B. Estimating similarity indexes

The presented design of systems (10) and (12) relies on similarity indexes s_t and S_t based on the optimal strategy U^O. The strategy is however not available at time t, when is necessary to calculate the similarity indexes. Hence, the similarity indexes should be estimated from the available information.

This fact turn us back to the suboptimal strategies, where the convergence of the ith decisions with the growing t has been analyzed:

$$u_i(\mathcal{P}_i), u_i(\mathcal{P}_{i+1}), u_i(\mathcal{P}_{i+2}), \ldots, u_i(\mathcal{P}_t). \tag{20}$$

Let an element of the suboptimal strategy $u_i(\mathcal{P}_t)$ is called *d-optimal* if it does not change with the growing time for at least the last d steps:

$$u_i(\mathcal{P}_{t-d+1}) = u_i(\mathcal{P}_{t-d+2}) = \ldots = u_i(\mathcal{P}_t), \tag{21}$$

where $d \in \mathcal{N}$ is a chosen constant. Then, the strict similarity index can be estimated as:

$$\hat{s}_t = \max_i\{i; \text{if } u_j(.) \text{ for } j \in \{1, \ldots, i\} \text{ are d-optimal}\}, \tag{22}$$

and similarity index $\hat{S}_t$ can be estimated as a count of $u_i(.)$ rated as d-optimal:

$$\hat{S}_t = |\{i; \text{if } u_i(.) \text{ is d-optimal}\}|. \tag{23}$$

Instead of indexes s_t and S_t, it can be advantageous to use time independent values relating the indexes and the time t. Let us introduce the values:

$$c_1 = \max_t(t - s_t), \tag{24}$$

$$c_2 = \max_t(t - S_t), \tag{25}$$

where values c_1 and c_2 corresponds with maximal number of equations lost by step from (9) to (10). Now, we can replace the set $\{1, \ldots, s_t - h\}$ characterizing the systems (10) and (12) by the corresponding subset $\{1, \ldots, t - (c_1 + h)\}$.

Instead of the values of the constants c_1 and c_2 at the time t, their estimates $\hat{c}_{1,t}$ and $\hat{c}_{2,t}$ are used:

$$\hat{c}_{1,t} = \max_{i \in \{1,\ldots,t\}}(i - \hat{s}_i), \tag{26}$$

$$\hat{c}_{2,t} = \max_{i \in \{1,\ldots,t\}}(i - \hat{S}_i), \tag{27}$$

The values $\hat{c}_{1,t}$ and $\hat{c}_{2,t}$ increase with the time t and it can be shown that they converge to a small constant.

The speed of convergence is high, and the time of the last change $t_{ch;1}$ and $t_{ch;2}$ show the time required to learn the final values of $\hat{c}_{1,t}$ and $\hat{c}_{2,t}$, which is important for the implementation.

V. Example: Futures Trading

The presented approach has been motivated by futures trading task [15]. The original aim was to design an automatic trading system, which analyzes the prices and other market statistics and generates the recommendation to a trader.

Futures trading task is a task typically solved by exchange speculator, who takes the available information and decides, whether to buy or sell the commodity. He can profit by reselling the commodity, when the price is changing. A profit is made, when the speculator guesses the direction of the price's evolution, otherwise the speculator loses.

A. Futures trading as a game

From our point of view, the futures trading task can be interpreted as turn based game: the player obtains a price y_t at the beginning of each turn $t \in \{1, 2, \ldots, T\}$. He chooses his decision u_t, his decision partially depends on his guess, whether the price should increase $u_t = 1$ or decrease $u_t = -1$. The player can also decide not to play for the turn $u_t = 0$. At the beginning of the next turn $t + 1$, the player makes profit of $(y_{t+1} - y_t)u_t$. But the player pays a transaction cost $C|u_{t-1} - u_t|$ for each change of decision and the player starts with $u_0 = 0$.

The complexity of the task is in the existence of transaction cost, because the obtained profit can be smaller than cost paid for the entering and leaving the game. The second complex property is in the leaving the game, because the player can hold his decision for a lot of turns with paid only the first enter, waiting for the price increase over the acceptable threshold. In this scope, the player does not speculate only on the increase (or decrease), he speculates on increase (or decrease) of the price over the required transaction cost in the acceptable horizon.

The player aims to maximize his profit up to the horizon T:

$$G_1^T = \sum_{t=1}^{T}(y_t - y_{t-1})u_{t-1} - C|u_{t-1} - u_t|$$

with the initial condition $u_0 = 0$.

The described game is a typical optimization problem of dynamic decision making and all previous ideas can be applied.

B. Available data

We have 35 price sequences available for the off-line experiments. The data were collected once a day, when the exchange was closing. Each data set contains data from 1990 to 2005, which makes about 3900 samples all together. Five price sequences were chosen as a representative for the further experiments:

- Cocoa - CSCE (CC),

- Petroleum-Crude Oil Light - NMX (CL),
- 5-Year U.S. Treasury Note - CBOT (FV2),
- Japanese Yen - CME (JY),
- Wheat - CBOT (W).

C. The validity of assumption (Sec. III-B)

In the case of futures trading, the optimal strategy can be found. It is an subsequence of the largest suboptimal strategy $U(\mathcal{P}_T)$ designed at the whole sequence. It is possible to say, which part of subsequence is optimal and to prove this. Hence, the similarity indexes s_t and S_t (respectively constants c_1 and c_2) can be directly estimated.

To test of estimation of $\hat{s}_t$ and $\hat{S}_t$ we used algorithm presented in Sec. IV-B with $d = 2$. The similarity indexes and constants $\hat{c}_1$ and $\hat{c}_2$ were estimated for the available data (see Tab. I).

The table shows optimistic results. The existence of values c_1, c_2 shows that number equations in (10) and (12) will grow. Thus, the new information about Bellman function will be added to estimation algorithm in each time t. The small values of $c_1, c_2 \ll T$ show that the suboptimal strategy converges to the optimal one quickly.

Moreover, the detailed analysis shows that s_t is equal to S_t, and that the $\hat{c}_{1,t}$ and $\hat{c}_{2,t}$ stop changing relatively early and the learning part of the data can be relative small. The estimation of $\hat{c}_{1,t}$ and $\hat{c}_{2,t}$ gives satisfactory results close to real values. All these facts led to a conclusion that futures trading is the task with a strong similarity (Sec. III-F). An example of similarity indexes behavior for a task with a strong similarity is depicted on Fig. 1.

An exception, possessing a weak similarity, is the market with ticker CL. The obtained similarity indexes are depicted in Fig. 2 and Fig. 3. The difference between s_t and t is noticeable but it has only a local character, therefore the approach can be used - with the expectation of worse results on the intervals with a weak similarity.

D. Estimation of Bellman function parameters

This section express the concrete form of system (12) for futures trading task, as was obtained in Sec. III-E.

Let the parametrized form of Bellman function be:

$$\mathcal{V}(\mathcal{P}_t) \approx F(u_{t-1})\Psi_t, \tag{28}$$

where $\Psi_t = (y_t, y_{t-1}, \dots, y_{t-n}, 1)^T$ is called regressor and $F(.)$ is a row vector function $F(.) = (f_1(.), f_2(.), \dots f_{n+2}(.))$. The reason of the choice and the related proof can be found in [14]. It is important to mention here that the proof operates with optimal decisions. The estimating of regressor length n for futures trading is analyzed in [16].

TABLE I
DOMINATING CONSTANTS c_1 AND c_2

Market	c_1	c_2	$\hat{c}_{1,T}$	$\hat{c}_{2,T}$	$t_{ch;1}$	$t_{ch;2}$	T
CC	6	6	7	6	342	342	3822
CL	444	6	446	6	847	2205	3863
FV2	8	8	9	8	383	383	3766
JY	4	4	5	4	50	50	3871
W	7	7	8	7	2452	2452	3822

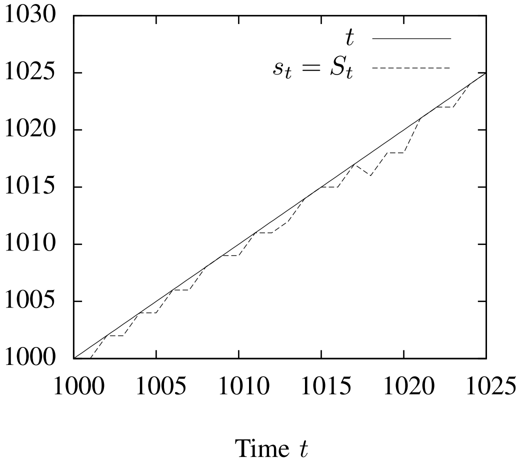

Fig. 1. Detail of similarity indexes S_t and s_t according the time t (Cocoa - CC)

We consider the form (28) as given and consequently derive the concrete form of system (12) in this section. The admissible values of u_t are $u^* = \{-1, 0, 1\}$, and the function $f_i(u_{t-1})$ is fully characterized by three values. Thus, the vector function $F(u_t)$ is fully characterized by $3(n + 2)$ parameters:

$$\Theta = (f_1(-1), f_1(0), f_1(1), \dots, f_{n+2}(-1), f_{n+2}(0), f_{n+2}(1)),$$

which is parameters vector introduced in Section III-E.

To obtain the realization of system (12), we insert (28) into (12):

$$\begin{aligned} G_k^{k+h} - \kappa_k \quad &= \quad F(u_{k-1}^O)\Psi_k - F(u_{k+h}^O)\Psi_{k+h+1} \\ &\quad\quad \text{for } k \in \{1, \dots, t - c_1 - h\}, \end{aligned} \tag{29}$$

where κ_k is error caused by approximation (12). The obtained system (29) can be rewritten as a system of linear equations:

$$b - \mathcal{K} = \mathbf{A}\Theta, \tag{30}$$

where $\mathbf{A}$ is matrix $(t - c_1 - h) \times (3n + 2)$ with elements from $\{-1, 0, 1\}$,

$$b = (G_1^{1+h}, G_2^{2+h}, \dots, G_{t-c_1-h}^{t-c_1})$$

and

$$\mathcal{K} = (\kappa_1, \kappa_2, \dots, \kappa_{t-c_1-h}).$$

The solution of the system (30) is documented in [17]. The obtained parameters determine Bellman function, which is used in optimization algorithm presented in [14].

E. Tuning the parameters

The aim of this experiment is to show the influence of the similarity indexes on the quality of the design. Hence, the sequence of four generic experiments was designed. The experiments differ in value of c_1 used in design (see Tab. II).

The values were chosen to show basic influence of the c_1 on the final gain. We expect that results should go worse with a growing distance from the correct value of c_1.

Experiment No. 1 uses the smaller value of $c_1 = 2$. This should cause taking non-optimal invalid equations into the system (12) and consequently worse results.

Experiment No. 2 uses the best values of c_1 and we expected the best results on it.

Experiments No. 3 and No. 4 take bigger values $c_1 = 10$ and $c_1 = 20$. These should cause the late adding of the equation into the system (12), hence the estimation does not use full available knowledge and may give worse results.

The results of experiments are compared via the gain function, which characterizes the profit from the trading (see Tab. II). The gain is summed over all markets. The obtained results correspond with our expectations, i.e. No. 2 gave the best profit.

F. Comparison with MPC

The aim of the experiment was to compare the obtained results with results given by the model predictive control (MPC). MPC can be viewed as special case of the dynamic decision task (1), where Bellman function is set to zero $\mathcal{V}(\mathcal{P}_t) = 0$. Hence, the experiment can be almost identical and the only difference is in Bellman function.

The results of the experiment are in Tab. II. The total profit of the presented approach is bigger than MPC. But taking the score market by market, the MPC gives better results at three markets of five, whereas the presented approach wins only twice.

VI. Conclusions

The design of Bellman function is considered in the paper. The approach is based on the idea to solve the system of Bellman equations. The so-called suboptimal strategies are defined and used to work off the maximization from the obtained system. Bellman function is then computed as a solution of the functional equations system, which can be transferred into the system of algebraic equation by choosing the parametrized shape of Bellman function.

TABLE II

Results overview: relation of c_1-values and the gain, comparison with results obtained by MPC method

Ticker	Var.	No. 1	No. 2	No. 3	No. 4	MPC
CC	gain	6880	25040	6380	-4320	34730
	c_1	2	6	10	20	
CL	gain	13770	27350	-20530	21490	-20300
	c_1	2	6	10	20	
FV2	gain	-42171	25269	33325	-52144	-38546
	c_1	2	8	10	20	
JY	gain	-2493	7488	33538	-5008	52097
	c_1	2	4	10	20	
W	gain	25833	23448	22908	20018	29125
	c_1	2	7	10	20	
Total	gain	1819	108595	75621	-19964	57106

The paper contains the classification of the tasks implied by the different behavior of the suboptimal strategies and relates the behavior of suboptimal strategies to applicability of presented approach.

The approach is applied and demonstrated on an example of futures trading, which is a typical economic decision making task. The behavior of suboptimal strategies is tested. Then, the new design of Bellman function is applied for various settings. Results of experiments are presented and compared. The comparison obtained result with MPC method is presented. The results have proved the potential applicability of the approach in the trading task.

A. Open questions

A questions remain open for the further research:

- The assumption on convergence (Sec. III-B) has the key importance on applicability of the whole approach. The assumption is valid on futures trading task, but there is no evidence whether it is valid for other problems.
- The presented results were obtained on delegates of available data (CC, CL, FV2, JY, and W). But only the application on all available data can verify the method and show the real advantages and disadvantages of the approach.
- The trading problem is sufficient to find the main properties of the approach. But it is necessary to verify the approach by another task with.

Acknowledgment

This work was supported by grants GA ČR 102/08/0567, MŠMT 1M0572.

References

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

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

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

[4] R. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[5] B. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientic, 1996.

[6] P. Werbos, *Handbook of intelligent control: neural, fuzzy and adaptive approaches*. Van Nostrant Reinhold, 1992, ch. Approximate dynamic programming for real-time control and neural modelling.

[7] A. G. Barto, S. J. Bradtke, and S. P. Singh, "Learning to act using real-time dynamic programming," *Artificial Intelligence*, vol. 72, no. 1-2, pp. 81–138, 1995.

[8] R. S. Sutton, "Learning to predict by the method of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.

[9] Watkins C., D. P., and and, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[10] S. Schaal, "Learning from demonstration," in *Advances in Neural Information Processing Systems 9*. MIT Press, 1997.

[11] R. H. Crites and A. G. Barto, "Elevator group control using multiple reinforcement learning agents," *Machine learning*, vol. 33, no. 2-3, pp. 235–262, 1998.

[12] J. M. Lee and J. H. Lee, "Approximate dynamic programming-based approaches for input–output data-driven control of nonlinear processes," *Automatica*, vol. 41, no. 7, pp. 1281–1288, 2005.

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

[14] M. Kárný, J. Šindelář, Š. Pírko, and J. Zeman, "Adaptively optimized trading with futures," ÚTIA AV ČR, v.v.i., Tech. Rep., 2010.

[15] J. Hull, *Options, futures, and other derivatives*. Pearson/Prentice Hall, 2006.

[16] O. Křivánek and J. Zeman, "Experiment: Setting the length of the regeressor," ÚTIA AV ČR, v.v.i., Tech. Rep. 2262, 2009.

[17] O. Křivánek, *Extended-Iterations-Spread-in-Time Strategy in Fully Probabilistic Design (Master's degree thesis)*. Czech Technical University, 2010.

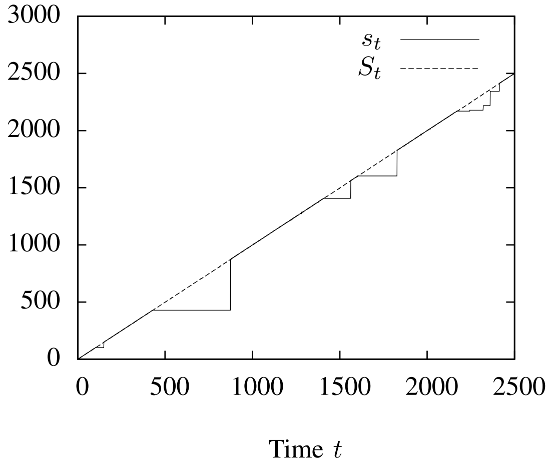

Fig. 2. Similarity indexes S_t and s_t according the time t (Petroleum-Crude Oil Light - CL)

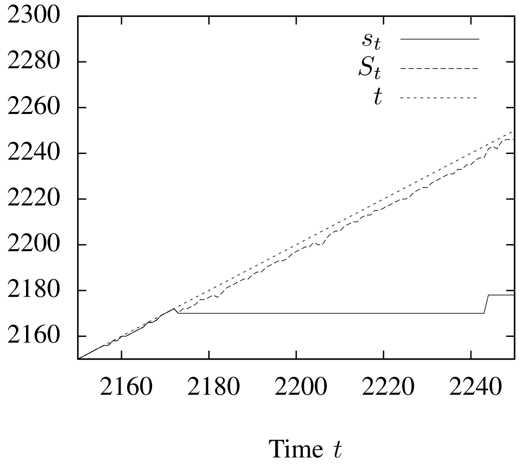

Fig. 3. Detail of similarity indexes S_t and s_t according the time t (Petroleum-Crude Oil Light - CL)