Bull. Kyushu Inst. Tech.
Pure Appl. Math. No. 55, 2008, pp. 15–24

INFINITE STAGE NONDETERMINISTIC STOPPED DECISION PROCESSES AND MAXIMUM LINEAR EQUATION

Toshiharu FUJITA

Received November 30, 2007

Abstract

In this paper, we are concerned with two types of infinite stage nondeterministic stopped decision processes. Our purposes are to derive the optimal equations for stopped decision processes and to give a method to solve some maximum linear equations by using the technique of dynamic programming.

1 Introduction

This paper considers stopped decision processes in the framework of infinite stage nondeterministic dynamic programming. Dynamic programming has been originated by Bellman [1] and developed and applied by Howard [6], Nemhauser [8], Bertsekas [3], White [11], Bellman and Zadeh [2], Sniedovich [10], Puterman [9], Iwamoto and Fujita [7] and others. Dynamic programming models can be classified under transition systems. Deterministic transition system and stochastic one are most popular and also fuzzy one often appears in problems with uncertainty. Moreover we have recently proposed a dynamic programming with nondeterministic transition system ([4], [5]). It is called nondeterministic dynamic programming. Under the nondeterministic system, a single state yields more than one state with nonnegative weight in the next stage. This characteristic shows that nondeterministic dynamic programming covers traditional ones. It also has a strong possibility for applying the idea of dynamic programming to various problems.

In this study, we apply infinite stage nondeterministic dynamic programming to two types of stopped decision processes. One has a stopping region. If a state moves into it, the decision process is stopped. The other has decision 'stop'. If decision-maker takes 'stop', the decision process is stopped. Furthermore we apply the result for stopped decision processes to a system of maximized linear equations. We show that an algorithm based on dynamic programming still works for solving the system.

2 Infinite stage nondeterministic dynamic programming

2.1 Notations and definitions

An infinite-stage nondeterministic dynamic programming is defined by four-tuple:

$$\mathcal{N} = (X, \{U, U(\cdot)\}, T, \{r, \beta\}),$$

where the definitions of each component are as follows.

1. X is a nonempty finite set which denotes a *state space*. Its elements $x_n \in X$ are called nth states. x_0 is an initial state.
2. U is a nonempty finite set which denotes a *decision space*. Furthermore we also denote by U a mapping from X to 2^U and $U(x)$ is the set of all feasible decisions for a state $x \in X$, where 2^Y denotes the following power set:

$$2^Y = \{A \mid A \subset Y, A \neq \emptyset\}.$$

After this, let $G_r(U)$ denote the graph of a mapping $U(\cdot)$:

$$G_r(U) := \{(x, u) \mid u \in U(x), x \in X\} \subset X \times U,$$

and $\mathbf{R}$ denotes the real number system.

3. $T : G_r(U) \to 2^X$ is a *nondeterministic transition law*. For each pair of a state and a decision $(x, u) \in G_r(U)$, $T(x, u)$ means the set of all states appeared in the next stage. If a decision u_n is chosen for a current state x_n, all $x_{n+1} \in T(x, u)$ will become a next state at the same time.
4. $r : G_r(U) \to \mathbf{R}$ is a *reward function* and $\beta : G_r(T) \to [0, \infty)$ is a *weight function*. If a decision u_n is chosen for a current state x_n, we get a reward $r(x_n, u_n)$ and each next state $x_{n+1} \in T(x_n, u_n)$ will be appeared with a corresponding weight $\beta(x_n, u_n, x_{n+1})$.

Next a mapping $f : X \to U$ is called *decision function* if $f(x) \in U(x)$ for any $x \in X$. A sequence of decision functions:

$$\pi = \{f_0, f_1, \ldots, f_n, \ldots\}$$

is called a *Markov policy*. Let Π denotes the set of all Markov policies, which is called *Markov policy class*. If a decision-maker takes a Markov policy $\pi = \{f_0, f_1, \ldots\}$, he chooses $f_n(x_n)(\in U)$ for state x_n at nth stage.

2.2 Formulation

For an initial state $x_0 \in X$ and Markov policy $\pi \in \Pi$ we introduce *total weighted value* is given by

$$V(x_0;\pi) := r_0 + \sum_{x_1 \in X(1)} \beta_0 r_1 + \sum\sum_{(x_1,x_2) \in X(2)} \beta_0\beta_1 r_2$$
$$+ \cdots + \sum\sum\cdots\sum_{(x_1,\dots,x_n) \in X(n)} \beta_0\beta_1 \dots \beta_{n-1} r_n + \cdots$$
$$x_0 \in X,\ \pi \in \Pi,$$

where

$$r_n = r(x_n, f_n(x_n)), \qquad \beta_n = \beta(x_n, f_n(x_n), x_{n+1}),$$
$$X(n) = \{(x_1,\dots,x_n) \in X \times \cdots \times X \mid x_{m+1} \in T(x_m, f_m(x_m))\ \ 0 \le m \le n-1\}.$$

Then our purpose is to maximize the total weighted value over Markov policy class. Thus the *nondeterministic dynamic programming problem* is formulated as the following maximization problem:

$$\mathrm{P}(x_0) \qquad \text{Maximize} \quad V(x_0;\pi) \quad \text{subject to} \quad \pi \in \Pi.$$

The problem $\mathrm{P}(x_0)$ means an infinite-stage decision process starting at 0th stage with an initial state x_0.

A policy π^* is called *optimal* if

$$V(x_0;\pi^*) \ge V(x_0;\pi) \qquad \forall \pi \in \Pi,\ \forall x_0 \in X.$$

2.3 Optimal equation

The norm of the weight function β is defined as follows

$$\beta_1 := \|\beta\|_1 = \max_{(x,u) \in G_r(U)} \sum_{y \in T(x,u)} |\beta(x,u,y)|.$$

Let $v(x_0)$ be the maximum value of $\mathrm{P}(x_0)$. Then we have the following optimal equation.

THEOREM 2.1 ([4]). *Under the assumption $\beta_1 < 1$, the maximum value function v satisfies the following optimal equation:*

$$v(x) = \max_{u \in U(x)} \left[r(x,u) + \sum_{y \in T(x,u)} \beta(x,u,y) v(y) \right] \qquad x \in X.$$

Note that the solution of this equation is unique.

Let $f^*(x) \in U(x)$ be a point which attains $v(x)$. Then we get the optimal stationary Markov policy $\pi^* = \{f^*, f^*, \dots\}$ in Markov class Π.

3 Stopped decision processes

Now we introduce two types of infinite stage nondeterministic stopped decision processes.

3.1 Stopping region

We define the infinite stage nondeterministic stopped decision process with stopping region:

$$\mathcal{NS}_1 = (\{\mathcal{S}, \mathcal{T}\}, \{U, U(\cdot)\}, T, \{r, g, \beta\}).$$

Additional definitions of new components are given below.

1^+. $\{\mathcal{S}, \mathcal{T}\}$ is a division of state space X. $\mathcal{T}$ is called a *stopping region*. If a state moves into $\mathcal{T}$, the process is terminated.

4^+. $g : \mathcal{T} \to \mathbf{R}$ is a *stop-reward function*. If the process is terminated, we get stop-reward instead of stage-reward given by r.

The other components are defined in section 2.

Then the optimal equation for the nondeterministic stopped decision process $\mathcal{NS}_1$ is obtained as follows.

COROLLARY 3.1.

$$v(x) = g(x) \qquad x \in \mathcal{T},$$

$$v(x) = \max_{u \in U(x)} \left[r(x, u) + \sum_{y \in T(x, u)} \beta(x, u, y) v(y) \right] \qquad x \in \mathcal{S}.$$

This corollary immediately follows from Theorem 2.1.

3.2 Decision 'Stop'

Next we introduce the decision 'Stop'. We define the infinite stage nondeterministic stopped decision process

$$\mathcal{NS}_2 = (X, \{U, U(\cdot)\}, T, \{r, g, \beta\}).$$

Additional definitions of new components are given below.

2^+. U includes a decision 'Stop', which is denoted by S. If we take $S \in U$, the process is terminated.

4^+. $g : X \to \mathbf{R}$ is a *stop-reward function*.

The other components are defined in section 2.

Then we have the following optimal equation for the nondeterministic stopped decision process $\mathcal{NS}_2$ by using Theorem 2.1.

COROLLARY 3.2.

$$v(x) = \max\left[g(x), \max_{u \in U(x)\setminus\{S\}}\left[r(x,u) + \sum_{y \in T(x,u)} \beta(x,u,y)v(y)\right]\right] \qquad x \in X.$$

4 Maximum linear equation

In this section, we use the following notations. For two real values a and b, their maxima is denoted by

$$a \vee b = \max\{a, b\}$$

and for the set of real values $\{a_1, a_2, \ldots, a_n\}$, their maxima by

$$\bigvee_{i=1}^{n} a_i = \max\{a_1, a_2, \ldots, a_n\}.$$

EXAMPLE 4.1. We show that to solve the following nondeterministic stopped decision process:

$$\mathcal{NS}_1 = (\{\mathcal{S}, \mathcal{T}\}, \{U, U(\cdot)\}, T, \{r, g, \beta\}),$$

where

$$\mathcal{S} = \{s_1\}, \qquad \mathcal{T} = \{s_2\}, \qquad U = \{1, 2\},$$
$$U(u) = U \qquad \forall u \in U,$$
$$T(x,u) = \{s_1, s_2\} \qquad \forall (x,u) \in U \times U,$$

implies to solve the following relation of a system of maximized linear equations:

$$\begin{aligned} x &= (b^1 + a_1^1 x + a_2^1 y) \vee (b^2 + a_1^2 x + a_2^2 y), \\ y &= c, \end{aligned} \tag{1}$$

where $a_j^k, b^k, c \in \mathbf{R}$ $(j, k = 1, 2)$.

Indeed, the optimal equations for $\mathcal{NS}_1$ becomes

$$\begin{aligned} v(s_1) &= \left[r(s_1,1) + \sum_{y=s_1,s_2} \beta(s_1,1,y)v(y)\right] \vee \left[r(s_1,2) + \sum_{y=s_1,s_2} \beta(s_1,2,y)v(y)\right], \\ v(s_2) &= g(s_2). \end{aligned} \tag{2}$$

We put

$$r(s_1,k) = b^k, \qquad g(s_2) = c, \qquad \beta(s_1,k,s_j) = a_j^k \qquad j,k = 1,2,$$

then Eqs. (2) are the same with Eqs. (1). In this case, variable x in Eqs. (1) plays the role of the value function $v(s_1)$ and y plays the role of $v(s_2)$. □

EXAMPLE 4.2. We consider the following nondeterministic stopped decision process:

$$\mathcal{N}\mathcal{S}_2 = (X, \{U, U(\cdot)\}, T, \{r, g, \beta\}),$$

where

$$X = \{s_1, s_2\}, \qquad U = \{1, 2, S\},$$
$$U(u) = U \qquad \forall u \in U,$$
$$T(x,u) = X \qquad \forall (x,u) \in U \times U.$$

Then to solve $\mathcal{N}\mathcal{S}_2$ implies to solve the following relation of a system of maximized linear equations:

$$\begin{aligned} x &= c_1 \vee (b_1^1 + a_{11}^1 x + a_{12}^1 y) \vee (b_1^2 + a_{11}^2 x + a_{12}^2 y), \\ y &= c_2 \vee (b_2^1 + a_{21}^1 x + a_{22}^1 y) \vee (b_2^2 + a_{21}^2 x + a_{22}^2 y), \end{aligned} \tag{3}$$

where $a_{ij}^k, b_i^k, c_i \in \mathbf{R}$ $(i,j,k = 1,2)$. Indeed, the optimal equation for $\mathcal{N}\mathcal{S}_2$ becomes

$$\begin{aligned} v(s_1) &= g(s_1) \vee \left[r(s_1,1) + \sum_{y=s_1,s_2} \beta(s_1,1,y)v(y) \right] \vee \left[r(s_1,2) + \sum_{y=s_1,s_2} \beta(s_1,2,y)v(y) \right], \\ v(s_2) &= g(s_2) \vee \left[r(s_2,1) + \sum_{y=s_1,s_2} \beta(s_2,1,y)v(y) \right] \vee \left[r(s_2,2) + \sum_{y=s_1,s_2} \beta(s_2,2,y)v(y) \right]. \end{aligned} \tag{4}$$

We put

$$r(s_i,k) = b_i^k, \qquad g(s_i) = c_i, \qquad \beta(s_i,k,s_j) = b_{ij}^k \qquad i,j,k = 1,2,$$

then Eqs. (4) are the same with Eqs. (3). □

Generally, these types of system of maximized linear equations give the optimal equations for infinite horizon optimal stopping problems under nondeterministic transition system.

Now we solve maximized linear equations by using dynamic programming approach. Let us consider the following relation of a system of equations:

$$x_i = \bigvee_{k=1}^{K_i} \left(\sum_{j=1}^{N} a_{ij}^k x_j + b_i^k \right) \qquad i = 1,2,\dots,N, \tag{5}$$

where $a_{ij}^k, b_i^k \in \mathbf{R}$ $(1 \le k \le K_i, 1 \le i, j \le N)$. We call the system (5) *maximum linear equation*. For the set $A = \{a_{ij}^k \in \mathbf{R} \mid 1 \le k \le K_i, 1 \le i, j \le N\}$, we use the following notations.

$$\|A\| = \max_{1 \le k \le K_i, 1 \le i \le N} \sum_{j=1}^{N} |a_{ij}^k|,$$

$$A \ge O \Leftrightarrow a_{ij}^k \ge 0 \qquad \text{for } 1 \le k \le K_i,\ 1 \le i, j \le N.$$

Note that under the assumption

$$\|A\| < 1$$

there exists a unique solution of Eq. (5). Further under the additional assumption

$$A \ge O,$$

we have the following algorithm for finding the unique solution.

ALGORITHM. (Howard's Policy Iteration Algorithm [6]).

Step 1 (initial selection)

Let $n = 0$. Take any feasible selection (decision function) f_0. The term feasible selection f means that f is a function satisfying

$$1 \le f(i) \le K_i \qquad i = 1, 2, \ldots, N.$$

Step 2 (value determination)

Solve the system of linear equations given by the maximum linear equation with the selection f_n and set the solution to $x^n = (x_1^n, x_2^n, \ldots, x_N^n)$, that is, the determined x^n satisfies

$$x_i^n = \sum_{j=1}^{N} a_{ij}^{f_n(i)} x_j^n + b_i^{f_n(i)} \qquad i = 1, 2, \ldots, N.$$

Step 3 (optimality test)

If x^n satisfies

$$x_i^n = \bigvee_{k=1}^{K_i} \left(\sum_{j=1}^{N} a_{ij}^k x_j^n + b_i^k \right) \qquad i = 1, 2, \ldots, N,$$

then go to step 6. Otherwise, go to step 4.

Step 4 (selection improvement)

Choose a feasible selection f_{n+1} satisfying

$$\bigvee_{k=1}^{K_i}\left(\sum_{j=1}^{N} a_{ij}^k x_j^n + b_i^k\right) = \sum_{j=1}^{N} a_{ij}^{f_{n+1}(i)} x_j^n + b_i^{f_{n+1}(i)} \qquad i = 1, 2, \ldots, N.$$

Step 5 (next step)
Let $n = n + 1$. Go to step 2.

Step 6 (optimal solution)
The selection f_n is optimal and (x^n, y^n) is the desired solution.

EXAMPLE 4.3. We solve the following maximum linear equations:

$$\begin{array}{lcccc}
x_1 = & 24 & \vee \left(4 + \frac{1}{4}x_1 + \frac{1}{3}x_2\right) & \vee & \left(6 + \frac{1}{3}x_1 + \frac{1}{3}x_2\right), \\
x_2 = & 18 & \vee \left(8 + \frac{1}{2}x_1 + \frac{1}{6}x_2\right) & \vee & \left(5 + \frac{1}{3}x_1 + \frac{1}{2}x_2\right). \\
 & u = S \text{ (stop)} & u = 1 & & u = 2
\end{array}$$

<u>**1st iteration**</u>

Step 1. Let $n = 0$ and $f_0 = (S, S)$, where $f_n = (k_1, k_2)$ means $f_n(1) = k_1$, $f_n(2) = k_2$.

Step 2. The linear equations

$$\begin{cases} x_1 = 24, \\ x_2 = 18 \end{cases}$$

have the solution $(x_1^0, x_2^0) = (24, 18)$.

Step 3. We check if (x_1^0, x_2^0) satisfies the maximum linear equations:

$$\begin{cases} 24 \vee \left(4 + \frac{1}{4}x_1^0 + \frac{1}{3}x_2^0\right) \vee \left(6 + \frac{1}{3}x_1^0 + \frac{1}{3}x_2^0\right) = 24 \vee 16 \vee 20 = 24 = x_1^0, \\ 18 \vee \left(8 + \frac{1}{2}x_1^0 + \frac{1}{6}x_2^0\right) \vee \left(5 + \frac{1}{3}x_1^0 + \frac{1}{2}x_2^0\right) = 18 \vee 23 \vee 22 = 23 \neq x_2^0. \end{cases}$$

Since they are not satisfied go to step 4.

Step 4. Set $f_1 = (S, 1)$.

Step 5. Go to step 2.

<u>**2nd iteration**</u>

Step 2. The linear equations

$$\begin{cases} x_1 = 24, \\ x_2 = 8 + \frac{1}{2}x_1 + \frac{1}{6}x_2 \end{cases}$$

yield the solution $(x_1^1, x_2^1) = (24, 24)$.

Step 3. We check if (x_1^1, x_2^1) satisfies

$$\begin{cases} 24 \vee \left(4 + \frac{1}{4}x_1^1 + \frac{1}{3}x_2^1\right) \vee \left(6 + \frac{1}{3}x_1^1 + \frac{1}{3}x_2^1\right) = 24 \vee 18 \vee 22 = 24 = x_1^1, \\ 18 \vee \left(8 + \frac{1}{2}x_1^1 + \frac{1}{6}x_2^1\right) \vee \left(5 + \frac{1}{3}x_1^1 + \frac{1}{2}x_2^1\right) = 18 \vee 24 \vee 25 = 25 \neq x_2^1. \end{cases}$$

Since they are not satisfied, go to step 4.

Step 4. Set $f_1 = (S, 2)$.

Step 5. Go to step 2.

3rd iteration

Step 2. The linear equations

$$\begin{cases} x_1 = 24, \\ x_2 = 5 + \frac{1}{3}x_1 + \frac{1}{2}x_2 \end{cases}$$

yield the solution $(x_1^2, x_2^2) = (24, 26)$.

Step 3. We check if (x_1^2, x_2^2) satisfies

$$\begin{cases} 24 \vee \left(4 + \frac{1}{4}x_1^2 + \frac{1}{3}x_2^2\right) \vee \left(6 + \frac{1}{3}x_1^2 + \frac{1}{3}x_2^2\right) = 24 \vee \frac{56}{3} \vee 22 = 24 = x_1^2, \\ 18 \vee \left(8 + \frac{1}{2}x_1^2 + \frac{1}{6}x_2^2\right) \vee \left(5 + \frac{1}{3}x_1^2 + \frac{1}{2}x_2^2\right) = 18 \vee \frac{73}{3} \vee 26 = 26 = x_2^2. \end{cases}$$

Since the third solution (x_1^2, x_2^2) satisfies the original maximum linear equation, go to step 6.

Step 6. Thus we get the optimal selection $f_2 = (S, 2)$ and the desired unique solution:

$$(x_1^*, x_2^*) = (x_1^2, x_2^2) = (24, 26). \qquad \square$$

Acknowledgments. The author wishes to thank Professor S. Iwamoto for valuable advice on this investigation.

References

[1] R. E. Bellman, Dynamic Programming, NJ: Princeton Univ. Press, 1957.

[2] R. E. Bellman and L. A. Zadeh, Decision-making in a fuzzy environment, Management Science, **17** (1970), B141–B164.

[3] D. P. Bertsekas, Dynamic Programming and Stochastic Control, Academic Press, New York, 1976.

[4] T. Fujita, On Nondeterministic Dynamic Programming, RIMS Kokyuroku (Kyoto Univ.), **1488** (2006), 15–24.

[5] T. Fujita, T. Ueno, and S. Iwamoto, A nondeterministic dynamic programming model, Lecture Notes in Artificial Intelligence **3214** (2004), 1208–1214.

[6] R. A. Howard, Dynamic Programming and Markov Processes, MIT Press, Cambridge, Mass., 1960.
[7] S. Iwamoto and T. Fujita, Stochastic decision-making in a fuzzy environment, J. Oper. Res. Soc. Japan, **38** (1995), 467–482.
[8] G. L. Nemhauser, Introduction to Dynamic Programming, Wiley, New York, 1966.
[9] M. L. Puterman, Markov Decision Processes: discrete stochastic dynamic programming, Wiley & Sons, New York, 1994.
[10] M. Sniedovich, Dynamic Programming, Marcel Dekker, Inc. NY, 1992.
[11] D. J. White, Dynamic Programming, Holden-Day, San Francisco, Calif., 1969.

Department of Mathematics
Kyushu Institute of Technology
Tobata, Kitakyushu 804-8550, *Japan*
E-mail address: fujita@mns.kyutech.ac.jp