DISCRETE DYNAMIC PROGRAMMING WITH RECURSIVE ADDITIVE SYSTEM

Iwamoto, Seiichi
Department of Mathematics, Kyushu University

https://doi.org/10.5109/13082

出版情報：統計数理研究. 16 (1/2), pp.49-66, 1974-03. Research Association of Statistical Sciences
バージョン：
権利関係：

DISCRETE DYNAMIC PROGRAMMING WITH RECURSIVE ADDITIVE SYSTEM

By

Seiichi IWAMOTO*

(Received July 15, 1973)

1. Introduction.

In the paper [5], N. Furukawa and S. Iwamoto have defined Markovian decision processes with a new broad class of reward systems, that is, recursive reward functions, and have studied the existence and properties of optimal policies. Under some conditions on the reward functions, they have proved that there exists a (p, ε)-optimal stationary policy and that in the case of a finite action space there exists an optimal stationary policy. These are some generalizations of results by D. Blackwell [3].

In this paper the author defines a dynamic programming problem with a recursive additive system which is referred to one type of Markovian decision processes with recursive reward functions defined by the previous authors [5]. This paper gives an algorithm for finding optimal stationary policies in the dynamic programming with the recursive additive system in the case of finite state and action spaces. Furthermore, we give several interesting examples with numerical computations to obtain optimal policies.

The motivation to consider the dynamic programming problem with the recursive additive system is the following: If we restrict the "reward" in narrow sense, for instance, the money in economic systems or the loss in statistical decision problems, it will be appropriate for us to accept the total sum of stage-wise rewards as a performance index. That is so-called additive reward system. But many practical problems in the field of engineerings enable us to interpret the "reward" in wider sense. In those problems we often encounter much complicated reward systems that are more than so-called additive. We have an interesting class of such complicated reward systems in which we can find a common feature named "recursive additive". By talking about various reward systems belonging to this class at the same time, we can make clear, as a dynamic programming problem, an important common property within the class, Our proofs are partially owing to Blackwell [2].

2. Notations and optimization problem.

A dynamic programming problem with a recursive additive system is, in general, defined by six-tuple $\{S, A, p, r, \beta, t\}$, where S is a set of *states* labeled by the integers

* Department of Mathematics, Kyushu University, Fukuoka.

$s=1,2,\cdots,N$, that is, $S=\{1,2,\cdots,N\}$, A is a set of *actions* labeled by the integers $a=1,2,\cdots,K$, that is, $A=\{1,2,\cdots,K\}$, p is a *transition law* p_{ij}^k, that is,

$$\sum_{j=1}^{N} p_{ij}^k = 1\,, \quad p_{ij}^k \geqq 0 \qquad \text{for} \quad i, j \in S,\ k \in A\,,$$

$r=(r_{ij}^k;\ i,j\in S,\ k\in A)$ is a set of *stage-wise rewards*, $\beta=(\beta_{ij}^k;\ i,j\in S,\ k\in A)$ is a *generalized accumulator* whose value β_{ij}^k is a discount factor depending on transition (i, k, j), and t is a *translator* from R^1 to R^1.

Throughout this paper we call the dynamic programming problem with recursive additive system "recursive additive dynamic programming" or simply "recursive additive DP". We sometimes use the convenient notations $\beta(i, k, j)$, $r(i, k, j)$ and $p(i, k, j)$ in stead of β_{ij}^k, r_{ij}^k and p_{ij}^k respectively.

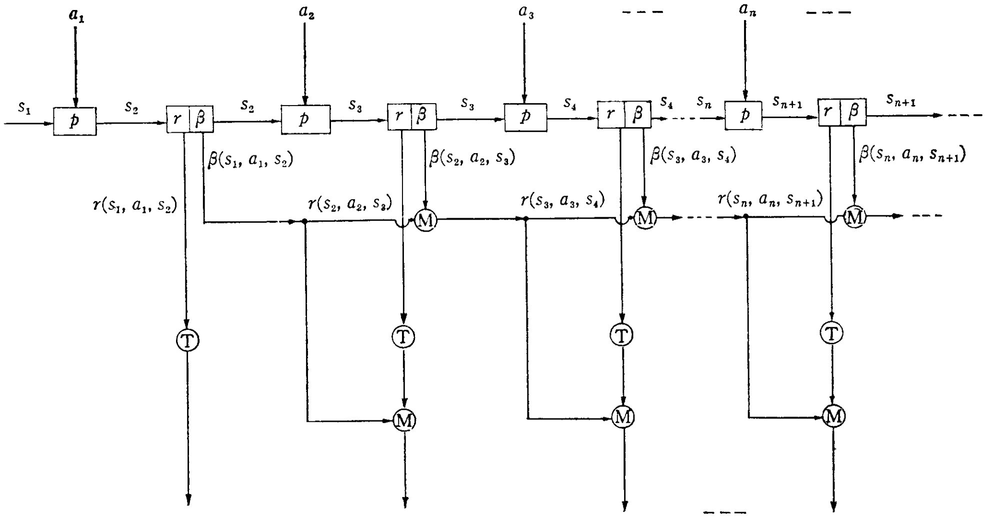
When the system starts from initial state $s_1\in S$ at 1-st stage and the decision maker takes an action $a_1\in A$ on this state s_1, the system moves to next state $s_2\in S$ with probability $p(s_1, a_1, s_2)$ at 2-nd stage and the system yields a stage-wise reward $r(s_1, a_1, s_2)$ and a discount factor $\beta(s_1, a_1, s_2)$.

However, at the end of 1-st stage the decision maker indeed gets the translated reward $t(r(s_1, a_1, s_2))$. The system is then repeated from the new state $s_2\in S$ at 2-nd stage. If he chooses an action $a_2\in A$ on state s_2, it moves to state s_3 with probability $p(s_2, a_2, s_3)$ at 3-rd stage. Then the system also yields a stage-wise reward $r(s_2, a_2, s_3)$ and a discount factor $\beta(s_2, a_2, s_3)$ at the end of 2-nd stage, and he really receives the discounted reward $\beta(s_1, a_1, s_2)t(r(s_2, a_2, s_3))$ of the translated one $t(r(s_2, a_2, s_3))$ multiplied by a discount factor $\beta(s_1, a_1, s_2)$ which was swept at the end of 1-st stage. Similarly at the end of 3-rd stage he gets a reward $\beta(s_1, a_1, s_2)\beta(s_2, a_2, s_3)t(r(s_3, a_3, s_4))$ which is discounted one of $t(r(s_3, a_3, s_4))$ multiplied by $\beta(s_1, a_1, s_2)\beta(s_2, a_2, s_3)$. In general when he undergoes the history $(s_1, a_1, s_2, a_2, \cdots, s_n, a_n, s_{n+1})$ of the system up to n-th stage, he comes to receive a reward $\beta(s_1, a_1, s_2)\beta(s_2, a_2, s_3)\cdots\beta(s_{n-1}, a_{n-1}, s_n)t(r(s_n, a_n, s_{n+1}))$ at the end of n-th stage. Furthermore, the process goes on $(n+1)$-st stage, $(n+2)$-nd stage and so on.

Since we consider a sequential nonterminating decision process, the decision maker continues to take actions infinitely. Consequently if he undergoes the history $h=(s_1, a_1, s_2, a_2, \cdots)$, he comes to receive the total reward

$$\begin{aligned} V(h) = {} & t(r(s_1, a_1, s_2)) + \beta(s_1, a_1, s_2)t(r(s_2, a_2, s_3)) \\ & + \beta(s_1, a_1, s_2)\beta(s_2, a_2, s_3)t(r(s_3, a_3, s_4)) \\ & + \cdots + \beta(s_1, a_1, s_3)\beta(s_2, a_2, s_3) \\ & \cdots \beta(s_{n-1}, a_{n-1}, s_n)t(r(s_n, a_n, s_{n+1})) + \cdots\,. \end{aligned}$$

This reward system is illustrated as follows:



$$t(r(s_1, a_1, s_2))+\beta(s_1, a_1, s_2)t(r(s_2, a_2, s_3))$$
$$+\beta(s_1, a_1, s_2)\beta(s_2, a_2, s_3)t(r(s_3, a_3, s_4))+\cdots$$
$$+\beta(s_1, a_1, s_2)\beta(s_2, a_2, s_3)\cdots\beta(s_{n-1}, a_{n-1}, s_n)t(r(s_n, a_n, s_{n+1}))$$
$$+\cdots$$

Fig. 1. Reward system of recursive additive DP.

It should be noted that Ⓣ, · and Ⓜ mean

$$\begin{tikzcd} r \arrow[d] & & & & & b \arrow[d] \\ \textcircled{T} \arrow[d] & r \arrow[r] & \cdot \arrow[r] \arrow[d] & r & a \arrow[r] & \textcircled{M} \arrow[d] \\ t(r) & & r & & & a\cdot b \end{tikzcd}$$

respectively in Fig. 1. The decision maker wishes to maximize his total expected reward over the infinite future.

We assume that he has a complete information on his history consisted of states and actions up to date and that he knows not only the stage-wise reward $r=(r^k_{ij})$, its translator $t=t(\cdot)$ and the generalized accumulator $\beta=(\beta^k_{ij})$ but also the recursive additivity of reward system in Fig. 1.

Let F denote the (finite) set of all functions from S to A. By a *policy*, we mean a sequence $\{f_n,\ n=1,2,\cdots\}$ of functions $f_n\in F$. We usually write a policy $\pi=\{f_1,f_2,\cdots\}$. Using policy π means that, if we find the system in state s on the n-th day, the action chosen that day is $f_n(s)\in A$. A policy $\{g,f_1,f_2,\cdots\}$ is denoted by $\{g,\pi\}$. By $\{g^n,\pi\}$, we mean the policy $\{g,g,\cdots,g,f_1,f_2,\cdots\}$. For $\pi=\{f_1,f_2,\cdots\}$, we define ${}^n\pi$ for each n by ${}^n\pi=\{f_{n+1},f_{n+2},\cdots\}$. In particular ${}^0\pi=\pi$. A policy $\{f,f,\cdots\}$, denoted by $f^{(\infty)}$, is called a *stationary policy.*

We associate with each $f\in F$ the $N\times 1$ column vector $\bar{r}(f)$ whose i-th element $\bar{r}(f)(i)$ is

$$\bar{r}(f)(i)=\sum_{j=1}^{N}p^{f(i)}_{ij}t(r^{f(i)}_{ij})\qquad i=1,2,\cdots,N$$

and (ii) the $N\times N$ Markov matrix $\bar{P}(f)$ whose (i,j) element $\bar{P}(f)(i,j)$ is

$$\bar{P}(f)(i,j)=p_{ij}^{f(i)}\times\beta_{ij}^{f(i)} \qquad i,j=1,2,\cdots,N.$$

If the decision maker uses policy $\pi=\{f_1,f_2,\cdots\}$ and the system is initially in state $i\in S$, his total expected return from π is the column vector

$$V(\pi)=\sum_{n=0}^{\infty}\bar{P}_n(\pi)\bar{r}(f_{n+1}),$$

where $\bar{P}_0(\pi)=I$, the $N\times N$ identity matrix, and for $n\geqq 1$

$$\bar{P}_n(\pi)=\bar{P}(f_1)\bar{P}(f_2)\cdots\bar{P}(f_n).$$

That is, i-th element of $V(\pi)$ is

$$V(\pi)(i)=\bar{r}(f_1)(i)+\sum_{j=1}^{N}p_{ij}^{f_1(i)}\beta_{ij}^{f_1(i)}\bar{r}(f_2)(j)+\sum_{j,k=1}^{N}p_{ij}^{f_1(i)}p_{jk}^{f_2(j)}\beta_{ij}^{f_1(i)}\beta_{jk}^{f_2(j)}\bar{r}(f_3)(k)$$
$$+\cdots+\sum_{j,k,\cdots,h,l=1}^{N}p_{ij}^{f_1(i)}p_{jk}^{f_2(j)}\cdots p_{hl}^{f_{n-1}(h)}\beta_{ij}^{f_1(i)}\beta_{jk}^{f_2(j)}\cdots\beta_{hl}^{f_{n-1}(h)}\bar{r}(f_n)(l)+\cdots.$$

We associate with each $f\in F$ the operator $L(f)$ which maps the $N\times 1$ column vector u into $L(f)u=\bar{r}(f)+\bar{P}(f)u$. For any two column vectors u, v we write $u\geqq v$ if every coordinate of u is at least as large as the corresponding coordinate of v, and $u>v$ if $u\geqq v$ and $u\neq v$. For $N\times 1$ column vector $u=(u_1,\cdots,u_N)'$ and $N\times N$ matrix $A=(a_{ij})$, we use the norms $\|u\|$, $\|A\|$ defined by $\|u\|=\max_{1\leqq i\leqq N}|u_i|$, $\|A\|=\max_{1\leqq i\leqq N}\sum_{j=1}^{N}|a_{ij}|$ respectively.

A policy π^* is called *optimal* if $V(\pi^*)\geqq V(\pi)$ for all π.

3. Optimal policies.

We now set

ASSUMPTION (I). It follows that $0\leqq\beta_{ij}^k<1$ for any $(i,k,j)\in S\times A\times S$.

We remark that if Assumption (I) is satisfied, we have for any policy π and any $i\in S$

$$\frac{r_*}{1-K_*}\leqq V(\pi)(i)\leqq\frac{r^*}{1-K^*},$$

where $r_*=\min_{i,k,j}t(r_{ij}^k)$, $r^*=\max_{i,k,j}t(r_{ij}^k)$, $K_*=\min_{i,k,j}\beta_{ij}^k$ and $K^*=\max_{i,k,j}\beta_{ij}^k$.

Hence Assumption (I) assures the finiteness of the vector $V(\pi)$ for any policy. Furthermore note that Assumption (I) implies the hypotheses in the following Lemmas 3.2 and 3.3.

Under Assumption (I) Furukawa and Iwamoto (Theorem 6.6 (ii) in [5]) have proved that there exists an optimal policy which is stationary in the finite state-and-action space dynamic programming problem with recursive additive functions.

In this section we shall give an algorithm for finding optimal policies which is stationary.

LEMMA 3.1. (i) *For any $f\in F$ and any policy π we have*

$$V(f, \pi) = L(f)V(\pi).$$

(ii) *For each* $f_n \in F$, $n \geqq 1$ *and any policy* π *we have*

$$V(f_1, f_2, \cdots, f_n, \pi) = L(f_1)L(f_2) \cdots L(f_n)V(\pi).$$

PROOF. The proof of this lemma is easy.

LEMMA 3.2. (i) *If* $\beta_{ij}^k \geqq 0$ *for each* $(i, k, j) \in S \times A \times S$, *then* $L(f)$ *is monotone, that is,* $u \geqq v$ *implies* $L(f)u \geqq L(f)v$ *for each* $f \in F$.

(ii) *If* $\max_{(i,k,j) \in SAS} |\beta_{ij}^k| < 1$, *then for each* $f \in F$ *the operator* $L(f)$ *is a contraction mapping on* R^N *with contraction coefficient less than* 1, *that is, there exists* $C = \max_{i,k,j} |\beta_{ij}^k|$ *such that* $\|L(f)u - L(f)v\| \leqq C\|u-v\|$ *for any* $N \times 1$ *vectors* u, v.

PROOF. The first result follows directly from the equality

$$L(f)u - L(f)v = \bar{P}(f)(u-v).$$

Then, the second follows by the inequalities

$$\|L(f)u - L(f)v\| \leqq \|\bar{P}(f)\| \times \|u-v\|,$$

$$\|\bar{P}(f)\| \leqq \max_{(i,k,j) \in SAS} |\beta_{ij}^k| \, \|P(f)\|$$

and

$$\|P(f)\| = 1,$$

where $\{P(f)(i,j)\} = \{P_{ij}^{f(i)}\}$ is the Markov matrix for each $f \in F$.

LEMMA 3.3. *Let* $\max_{i,k,j} |\beta_{ij}^k| < 1$ *be satisfied. Then it follows that* $\lim_{n \to \infty} L(f_1)L(f_2) \cdots L(f_n)w = V(\pi)$ *for any policy* $\pi = \{f_1, f_2, \cdots, f_n, \cdots\}$ *and any* $w \in R^N$.

PROOF. The proof is straightforward and we omit it.

Throughout the reminder of this section, we shall discuss analogously according to Blackwell [2].

THEOREM 3.1. *Let Assumption (I) be satisfied. If* $V(\pi^*) \geqq V(f, \pi^*)$ *for any* $f \in F$, *then* π^* *is optimal.*

PROOF. By Lemma 3.1 (i), our hypothesis is that

$$V(\pi^*) \geqq L(f)V(\pi^*) \qquad \text{for any } f \in F.$$

By Lemma 3.2 (i), we have for any policy $\pi = \{f_1, f_2, \cdots, f_n, \cdots\}$

$$V(\pi^*) \geqq L(f_1)L(f_2) \cdots L(f_n)V(\pi^*) \qquad n \geqq 1.$$

Consequently Lemma 3.3, with letting $n \to \infty$, implies that

$$V(\pi^*) \geqq V(\pi).$$

Since π is arbitrary policy, the proof is complete.

THEOREM 3.2. *Let Assumption (I) be satisfied. If* $V(f, \pi) > V(\pi)$, *then* $V(f^{(\infty)}) > V(\pi)$.

PROOF. By Lemma 3.1 (i), our hypothesis is that

$$L(f)V(\pi) > V(\pi).$$

Since Lemma 3.2 (i) holds, that is, operator $L(f)$ is monotone, applying the monotone operator $L^{n-1}(f)$ yields

$$L^n(f)V(\pi) \geqq L^{n-1}(f)V(\pi),$$

so that

$$V(f^n, \pi) \geqq V(f, \pi) > V(\pi) \qquad \text{for all} \quad n \geqq 1.$$

Then letting $n \to \infty$, we have by virtue of Lemma 3.3

$$V(f^{(\infty)}) > V(\pi).$$

Now we have our main results which are analogues of the Howard's policy improvement algorithm for the case where β depends on $(i, k, j) \in S \times A \times S$.

Theorem 3.3. *Let Assumption (I) be satisfied. Take any $f \in F$. For each $i \in S$, denote $G(i, f)$ the set of all $k \in A$ for which*

$$\bar{r}_i^k + \sum_{j=1}^{N} p_{ij}^k \beta_{ij}^k V(f^{(\infty)})(j) > V(f^{(\infty)})(i),$$

where $\bar{r}_i^k = \sum_{j=1}^{N} p_{ij}^k t(r_{ij}^k)$. If $G(i, f)$ is empty for all $i \in S$, then $f^{(\infty)}$ is optimal. For any $g \in F$ such that

(a) *$g(i) \in G(i, f)$ for some i and*

(b) *$g(i) = f(i)$ whenever $g(i) \notin G(i, f)$,*

we have $V(g^{(\infty)}) > V(f^{(\infty)})$.

Proof. It follows that

$$V(g, f^{(\infty)})(i) = \bar{r}(g)(i) + \bar{P}(g)V(f^{(\infty)})(i).$$

Furthermore, $V(g, f^{(\infty)})(i) > V(f^{(\infty)})(i)$ if and only if $g(i) \in G(i, f)$, and $V(g, f^{(\infty)})(i) = V(f^{(\infty)})(i)$ if $g(i) = f(i)$. Thus if $G(i, f)$ is empty for all $i \in S$, $V(f^{(\infty)}) \geqq V(g, f^{(\infty)})$ for all g so that, from Theorem 3.1, $f^{(\infty)}$ is optimal. On the other hand, for any g satisfying (a) and (b), we have

$$V(g, f^{(\infty)}) > V(f^{(\infty)}),$$

so that, from Theorem 3.2, $V(g^{(\infty)}) > V(f^{(\infty)})$.

Corollary 1. *Under Assumption (I), there is an optimal policy which is stationary.*

Proof. According to Theorem 3.3, if we take any stationary policy $f^{(\infty)}$, either it is optimal, that is, $G(i, f)$ is empty for all $i \in S$, or it has a stationary improvement $g^{(\infty)}$, that is, $G(i, f)$ is nonempty for some $i \in S$. Since F is finite, there is one which has no stationary improvement with finite interations, so that it must be optimal.

Corollary 2. *Let Assumption (I) be satisfied. Take any $f_1 \in F$. Then solve the linear equation*

$$v_i = \bar{r}(f_1)(i) + \sum_{j=1}^{N} \bar{P}(f_1)(i, j)v_j \qquad j = 1, 2, \cdots, N$$

for v_i $(i \in S)$. Using these values v_i $(i \in S)$, find the element of $G(i, f_1)$ for each $i \in S$. If $G(i, f_1)$ is empty for all $i \in S$, $f_1^{(\infty)}$ is optimal, and $V(f_1^{(\infty)})(i) = v_i$ for all $i \in S$. If at least $f_2(i) \in G(i, f_1)$ for some $i \in S$, we obtain an improved policy $f_2^{(\infty)}$ such that $f_2(i) \in G(i, f_1)$ for some i and $f_2(i) = f_1(i)$ for $G(i, f_1)$ empty. We then return to solving the linear equation for f_2. And again we find the element of $G(i, f_2)$ for each $i \in S$. If we repeat this procedure, then for some n we have

$$f_n(i) = f_{n+1}(i) \qquad \text{for all} \quad i \in S$$

and then $f_n^{(\infty)}$ is an optimal policy which is stationary.

PROOF. The proof follows directly from Theorem 3.3 and Corollary 1. Note that, as an example, we can take an initial $f_1 \in F$ such that $\bar{r}(f_1)(i) = \max_{1 \leqq k \leqq N} \sum_{j=1}^{N} p_{ij}^k t(r_{ij}^k)$.

4. Application to some additive systems.

We now illustrate some interesting examples with numerical computation which follow from the discussion in Section 3.

EXAMPLE 1. (General Additive System)

We say that a dynamic programming problem $\{S, A, p, r, \beta\}$ has a *general additive system* if $t(r) = r$ for any $r \in R^1$ in the recursive additive DP$\{S, A, p, r, \beta, t\}$ discussed at Section 2. We call this dynamic programming " general additive DP ". In this case we have an optimization problem where our aim is to maximize the expected reward of the function

$$V(h) = r_1 + \beta_1 r_1 + \beta_1 \beta_2 r_3 + \cdots + \beta_1 \beta_2 \cdots \beta_{n-1} r_n + \cdots$$

over the infinite future and to find the optimal policy among the all ones, where $h = (s_1, a_1, s_2, a_2, \cdots)$, $r_k = r_{s_k s_{k+1}}^{a_k} = r(s_k, a_k, s_{k+1})$, $\beta_k = \beta_{s_k s_{k+1}}^{a_k} = \beta(s_k, a_k, s_{k+1})$, $s_k \in S$ and $a_k \in A$ for $k \geqq 1$.

In many financial decision problems, the discount factor β_{ij}^k may depend on selected action $k \in A$. Then we have K different discount factors $\beta_1, \beta_2, \cdots, \beta_K$. These problems can be formulated into the general additive DP.

In particular, if $0 \leqq \beta_{ij}^k \equiv \beta < 1$, then we have a dynamic programming problem $\{S, A, p, r, \beta\}$ whose objective function is

$$V(h) = r_1 + \beta r_2 + \beta^2 r_3 + \cdots + \beta^{n-1} r_n + \cdots .$$

We call this dynamic programming " discounted DP ". Furthermore, we can make the modified discounted DP's by assigning the translator $t(\cdot)$ the specified function such as $t(r) = (1-r)e^r$ or $t(r) = \log r$.

We have a following data from the Taxicab Problem due to Howard [6].

TABLE 4.1.1.
Data for general additive DP.

state	action	transition probability			stage-wise reward			generalized accumulator		
i	k	p_{i1}^k	p_{i2}^k	p_{i3}^k	r_{i1}^k	r_{i2}^k	r_{i3}^k	β_{i1}^k	β_{i2}^k	β_{i3}^k
1	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	10	4	8	$\frac{95}{100}$	$\frac{98}{100}$	$\frac{98}{100}$
	2	$\frac{1}{16}$	$\frac{3}{4}$	$\frac{3}{16}$	8	2	4	$\frac{90}{100}$	$\frac{90}{100}$	$\frac{93}{100}$
	3	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{5}{8}$	4	6	4	$\frac{98}{100}$	$\frac{96}{100}$	$\frac{98}{100}$
2	1	$\frac{1}{2}$	0	$\frac{1}{2}$	14	0	18	$\frac{85}{100}$	$\frac{90}{100}$	$\frac{95}{100}$
	2	$\frac{1}{16}$	$\frac{7}{8}$	$\frac{1}{16}$	6	16	8	$\frac{80}{100}$	$\frac{80}{100}$	$\frac{95}{100}$
	3	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	-5	-5	-5	$\frac{95}{100}$	$\frac{95}{100}$	$\frac{95}{100}$
3	1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	10	2	8	$\frac{75}{100}$	$\frac{90}{100}$	$\frac{95}{100}$
	2	$\frac{1}{8}$	$\frac{3}{4}$	$\frac{1}{8}$	6	4	2	$\frac{95}{100}$	$\frac{70}{100}$	$\frac{80}{100}$
	3	$\frac{3}{4}$	$\frac{1}{16}$	$\frac{3}{16}$	4	0	8	$\frac{95}{100}$	$\frac{95}{100}$	$\frac{95}{100}$

Then the policy iteration algorithm (abbreviated, hereafter by PIA) described below gives an optimal stationary policy

$$f=\begin{pmatrix}1\\1\\3\end{pmatrix}.$$

TABLE 4.1.2.
Solution by the PIA for general additive DP.

v_1		119.660		169.490		
v_2		117.384		166.129		
v_3		106.376		164.411		
	VDO ↗		PIR ↘ VDO ↗		PIR ↘	
	1		1		1	
f	1		1		1	STOP
	1		3		3	

VDO ↗ is the value determination operation.
PIR ↘ is the policy improvement routine.

However the PIA for discounted DP with same transition probability and same stage-wise reward as general additive DP yields an optimal stationary policy

$$g=\begin{bmatrix}2\\2\\2\end{bmatrix}.$$

TABLE 4.1.3.
Solution by the PIA for discounted DP with $\beta^k_{ij}\equiv 0.90$.

v_1		91.257		119.439		121.653		
v_2		97.551		134.479		135.306		
v_3		89.967		121.927		122.837		
	1		1		2		2	
g	1		2		2		2	STOP
	1		2		2		2	

EXAMPLE 2. (Multiplicative Additive System)

We say that a dynamic programming $\{S, A, p, r\}$ has a *multiplicative additive system* if $\beta^k_{ij}=r^k_{ij}$ for any $(i, k, j)\in SAS$ and $t(r)=r$ in the recursive additive DP$\{S, A, p, r, \beta, t\}$. We call this dynamic programming "multiplicative additive DP". Then, the reward system is given as follows:

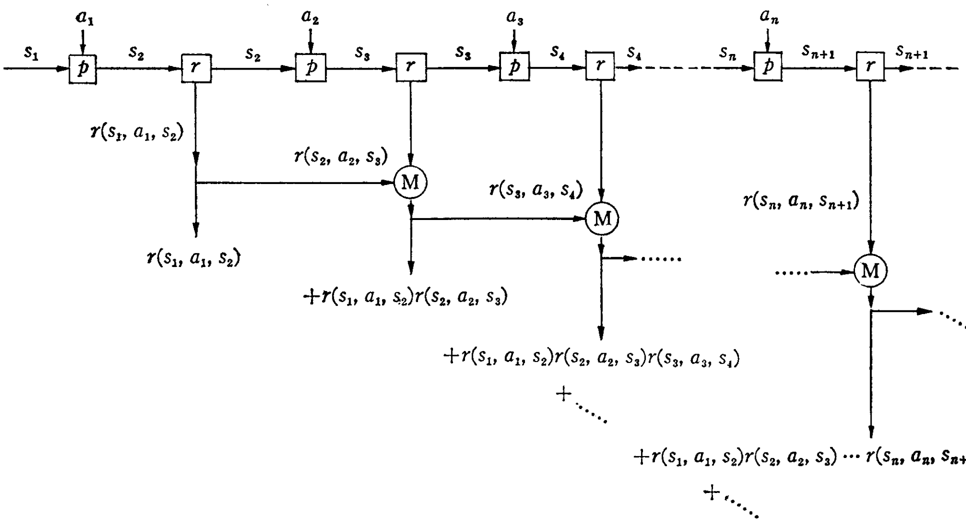


Fig. 2. Reward system of multiplicative additive DP.

This case leads us to maximization the expected value of the function

$$V(h)=r_1+r_1r_2+r_1r_2r_3+\cdots+r_1r_2\cdots r_{n-1}r_n+\cdots,$$

where r_k and h have the same meanings as Example 1. Then, Assumption (I) means $0\leqq r^k_{ij}<1$ for any $(i, k, j)\in SAS$.

The stage-wise reward r^k_{ij} in the following data satisfies Assumption (I).

TABLE 4.2.1.
Data for multiplicative additive DP.

state	action	transition probability			stage-wise reward			generalized accumulator		
i	k	p_{i1}^k	p_{i2}^k	p_{i3}^k	r_{i1}^k	r_{i2}^k	r_{i3}^k	β_{i1}^k	β_{i2}^k	β_{i3}^k
1	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{5}$	$\frac{2}{5}$	$\beta_{ij}^k=r_{ij}^k$ for $(i,k,j)\in SAS$		
	2	$\frac{1}{16}$	$\frac{3}{4}$	$\frac{3}{16}$	$\frac{2}{5}$	$\frac{1}{10}$	$\frac{1}{5}$			
	3	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{5}{8}$	$\frac{1}{5}$	$\frac{3}{10}$	$\frac{1}{5}$			
2	1	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{7}{10}$	$\frac{1}{20}$	$\frac{9}{10}$			
	2	$\frac{1}{16}$	$\frac{7}{8}$	$\frac{1}{16}$	$\frac{2}{5}$	$\frac{4}{5}$	$\frac{2}{5}$			
	3	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{20}$	$\frac{1}{20}$	$\frac{1}{20}$			
3	1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{10}$	$\frac{2}{5}$			
	2	$\frac{1}{8}$	$\frac{3}{4}$	$\frac{1}{8}$	$\frac{3}{10}$	$\frac{1}{5}$	$\frac{1}{10}$			
	3	$\frac{3}{4}$	$\frac{1}{16}$	$\frac{3}{16}$	$\frac{1}{5}$	$\frac{1}{20}$	$\frac{2}{5}$			

The calculations by PIA are shown in the following:

TABLE 4.2.2.
Solution by the PIA for multiplicative additive DP.

v_1		0.6990		0.7938		
v_2		1.3091		2.6198		
v_3		0.5876		0.6434		
		↗ ↘		↗ ↘		
	1		1		1	
f	1		2		2	STOP
	1		1		1	

Then we get an optimal stationary policy

$$f=\begin{bmatrix}1\\2\\1\end{bmatrix}$$

for this multiplicative additive DP. But the optimal one for discounted DP with $\beta=0.95$ is

$$g=\begin{bmatrix}2\\2\\2\end{bmatrix}.$$

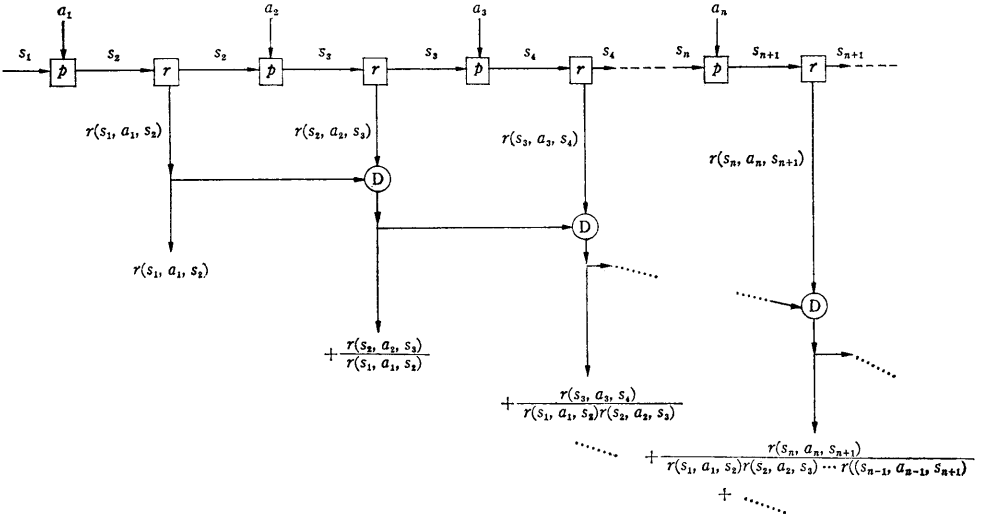
TABLE 4.2.3.

Solution by the PIA for discounted DP with $\beta^k_{ij} \equiv 0.95$.

v_1		9.1641		12.5129		12.7511		
v_2		9.4747		13.3053		13.4382		
v_3		9.0985		12.6705		12.8101		
	1		1		2		2	
g	1		2		2		2	STOP
	1		1		2		2	

EXAMPLE 3. (Divided Additive System)

We say that a dynamic programming $\{S, A, p, r\}$ has a *divided additive system* if $\beta^k_{ij} = 1/r^k_{ij}$ for any $(i, k, j) \in SAS$ and $t(r) = r$ in the recursive additive DP$\{S, A, p, r, \beta, t\}$. We call this dynamic programming " divided additive DP ". Then, we have the following reward system:



a_1
a_2
a_3
a_n
s_1
s_2
s_2
s_3
s_3
s_4
s_4
s_n
s_{n+1}
s_{n+1}
p
r
$r(s_1, a_1, s_2)$
$r(s_2, a_2, s_3)$
$r(s_3, a_3, s_4)$
$r(s_n, a_n, s_{n+1})$
D
$r(s_1, a_1, s_2)$
$+\frac{r(s_2, a_2, s_3)}{r(s_1, a_1, s_2)}$
$+\frac{r(s_3, a_3, s_4)}{r(s_1, a_1, s_2)r(s_2, a_2, s_3)}$
$+\frac{r(s_n, a_n, s_{n+1})}{r(s_1, a_1, s_2)r(s_2, a_2, s_3) \cdots r((s_{n-1}, a_{n-1}, s_{n+1})}$
$+ \cdots$

Fig. 3. Reward system of divided additive DP.

Here $\textcircled{D}$ means
$$\begin{tikzcd} & w \arrow[d] \\ \frac{v}{u} \arrow[r] & \textcircled{D} \arrow[d] \\ & \frac{w}{uv} \end{tikzcd}.$$

We have an optimization problem whose objective function is the expected return of the function

$$V(h) = r_1 + \frac{r_2}{r_1} + \frac{r_3}{r_1 r_2} + \cdots + \frac{r_n}{r_1 r_2 \cdots r_{n-1}} + \cdots,$$

where $h = (s_1, a_1, s_2, a_2, \cdots)$, $r_k = r^{a_k}_{s_k s_{k+1}}$ for $k \geqq 1$.

A dynamic programming problem $\{S, A, p, r, t\}$ is called "modified divided additive DP" if $\beta_{ij}^k = 1/r_{ij}^k$ for $(i, k, j) \in SAS$ in recursive additive DP$\{S, A, p, r, \beta, t\}$. Then in the modified divided additive DP

$$V(h) = r_1 + \frac{t(r_2)}{r_1} + \frac{t(r_3)}{r_1 r_2} + \cdots + \frac{t(r_n)}{r_1 r_2 \cdots r_{n-1}} + \cdots .$$

We can see, in Bellman's book ([1], Chap. I), an example of modified additive DP, which is restricted to the finite horizon and deterministic case.

If $r_{ij}^k > 1$ for any $(i, k, j) \in SAS$, then Assumption (I) is statisfied. The stage-wise rewards in the following satisfy Assumption (I).

Table 4.3.1.
Data for divided additive DP.

state	action	transition probability			stage-wise reward			generalized accumulator		
i	k	p_{i1}^k	p_{i2}^k	p_{i3}^k	r_{i1}^k	r_{i2}^k	r_{i3}^k	β_{i1}^k	β_{i2}^k	β_{i3}^k
1	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{3}{2}$	$\frac{6}{5}$	$\frac{7}{5}$	$\beta_{ij}^k = \frac{1}{r_{ij}^k}$ for (i,j,k)		
	2	$\frac{1}{16}$	$\frac{3}{4}$	$\frac{3}{16}$	$\frac{7}{5}$	$\frac{11}{10}$	$\frac{6}{5}$			
	3	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{5}{8}$	$\frac{6}{5}$	$\frac{13}{10}$	$\frac{6}{5}$			
2	1	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{17}{10}$	$\frac{21}{20}$	$\frac{19}{10}$			
	2	$\frac{1}{16}$	$\frac{7}{8}$	$\frac{1}{16}$	$\frac{7}{5}$	$\frac{9}{5}$	$\frac{26}{25}$			
	3	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{21}{20}$	$\frac{21}{20}$	$\frac{21}{20}$			
3	1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	$\frac{3}{2}$	$\frac{11}{10}$	$\frac{7}{5}$			
	2	$\frac{1}{8}$	$\frac{3}{4}$	$\frac{1}{8}$	$\frac{13}{10}$	$\frac{6}{5}$	$\frac{11}{10}$			
	3	$\frac{3}{4}$	$\frac{1}{16}$	$\frac{3}{16}$	$\frac{6}{5}$	$\frac{21}{20}$	$\frac{7}{5}$			

The calculation by PIA for this divided additive DP is given as follows:

Table 4.3.2.
Solution by the PIA for divided additive DP.

v_1		4.8429		7.3393		11.8020		
v_2		4.5288		8.3417		12.2804		
v_3		4.9567		7.2878		11.2934		
	1	↗ ↘	3	↗ ↘	2	↗ ↘	2	
f	1		3		3		3	STOP
	1		3		2		2	

We have an optimal stationary policy

$$f=\begin{pmatrix}2\\3\\2\end{pmatrix}.$$

On the other hand, if transition probability p_{ij}^k and state-wise reward r_{ij}^k are the same as ones for this divided additive DP respectively, the optimal stationary policy for discounted DP with discount factor $\beta=0.95$ is

$$g=\begin{pmatrix}2\\2\\2\end{pmatrix}.$$

Following table shows this calculation:

TABLE 4.3.3.
Solution by the PIA for discounted DP with $\beta_{ij}^k \equiv 0.95$.

v_1		29.1641		32.1969		32.3873		
v_2		29.4747		32.9426		33.0489		
v_3		29.0985		32.3348		32.4463		
	1		1		2		2	
g	1		2		2		2	STOP
	1		2		2		2	

EXAMPLE 4. (Exponential Additive System)

We say that a dynamic programming $\{S, A, p, r\}$ has an *exponential additive system* if $\beta_{ij}^k=e^{r_{ij}^k}$ for any $(i, k, j)\in SAS$ and $t(r)=r$ in the recursive additive DP $\{S, A, p, r, \beta, t\}$. We call this dynamic programming "exponential additive DP". The reward system of the exponential additive DP is just describled by Fig. 3 except for replacing Ⓓ by Ⓔ. Here Ⓔ means

$$\begin{tikzcd} & w \arrow[d] \\ e^{u}\cdot v \arrow[r] & \textcircled{E} \arrow[d] \\ & e^{u+v}\cdot w \end{tikzcd}$$

In this case, we have

$$V(h)=r_1+e^{r_1}\cdot r_2+e^{r_1+r_2}\cdot r_3+\cdots+e^{r_1+r_2\cdots r_{n-1}}\cdot r_n+\cdots,$$

where $h=(s_1, a_1, s_2, a_2, \cdots)$, $r_k=r_{s_k s_{k+1}}^{a_k}$ for $k\geqq 1$. Moreover, if $t(r)=(1-r)e^r$, $\beta_{ij}^k=e^{r_{ij}^k}$ for any $(i, k, j)\in SAS$ in recursive additive DP, we have another objective function

$$V'(h)=(1-r_1)e^{r_1}+(1-r_2)e^{r_1+r_2}+(1-r_3)e^{r_1+r_2+r_3}$$
$$+\cdots+(1-r_n)e^{r_1+r_2+\cdots+r_n}+\cdots.$$

We call this dynamic programming $\{S, A, p, r\}'$ "modified exponential additive DP". In these exponential additive DP's, Assumption (I) means $r_{ij}^k<0$ for any $(i, k, j)\in SAS$.

We can find an example of modified exponential additive DP in Bellman's book ([1], Chap. III). The stage-wide rewards illustrated in the following table satisfy Assumption (I).

Table 4.4.1.

Data for exponential additive DP.

state	action	transition probability			stage-wise reward			generalized accumulator		
i	k	p^k_{i1}	p^k_{i2}	p^k_{i3}	r^k_{i1}	r^k_{i2}	r^k_{i3}	β^k_{i1}	β^k_{i2}	β^k_{i3}
1	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$-\frac{1}{2}$	$-\frac{1}{5}$	$-\frac{2}{5}$	$\beta^k_{ij}=e^{r^k_{ij}}$ for (i,k,j)		
	2	$\frac{1}{16}$	$\frac{3}{4}$	$\frac{3}{16}$	$-\frac{2}{5}$	$-\frac{1}{10}$	$-\frac{1}{5}$			
	3	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{5}{8}$	$-\frac{1}{5}$	$-\frac{3}{10}$	$-\frac{1}{5}$			
2	1	$\frac{1}{2}$	0	$\frac{1}{2}$	$-\frac{7}{10}$	$-\frac{1}{20}$	$-\frac{9}{10}$			
	2	$\frac{1}{16}$	$\frac{7}{8}$	$\frac{1}{16}$	$-\frac{2}{5}$	$-\frac{4}{5}$	$-\frac{2}{5}$			
	3	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$-\frac{1}{20}$	$-\frac{1}{20}$	$-\frac{1}{20}$			
3	1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{2}{4}$	$-\frac{1}{2}$	$-\frac{1}{10}$	$-\frac{2}{5}$			
	2	$\frac{1}{8}$	$\frac{3}{4}$	$\frac{1}{8}$	$-\frac{3}{10}$	$-\frac{1}{5}$	$-\frac{1}{10}$			
	3	$\frac{3}{4}$	$\frac{1}{16}$	$\frac{3}{16}$	$-\frac{1}{5}$	$-\frac{1}{20}$	$-\frac{2}{5}$			

The optimal policy for the exponential additive DP which is stationary is given as follows:

Table 4.4.2.

Solution by the PIA for exponential additive DP.

v_1		−1. 0831		
v_2		−1. 0807		
v_3		−1. 0868		
	↗		↘	
	2		2	
f	3		3	STOP
	2		2	

Moreover, we obtain following tables:

TABLE 4.4.3.
Solution by the PIA for discounted DP with $\beta^k_{ij} \equiv 0.95$.

v_1		-2.1503		
v_2		-2.0935		
v_3		-2.2093		
	↗		↘	
	2		2	
g	3		3	STOP
	2		2	

TABLE 4.4.4.
Solution by the PIA for modified exponential additive DP with $t(r)=(1-r)e^r$

v_1		9.8747		
v_2		10.3750		
v_3		9.3211		
	↗		↘	
	2		2	
h	3		3	STOP
	2		2	

Of course the transition probability p^k_{ij} and stage-wise reward r^k_{ij} are common in these three DP's.

Note that each optimal policy is common one;

$$f=g=h=\begin{pmatrix} 2 \\ 3 \\ 2 \end{pmatrix}$$

and that policy iteration algorithm converges at first cycle respectively.

EXAMPLE 5. (Logarithmic Additive System)

We say that a dynamic programming $\{S, A, p, r\}$ has a *logarithmic additive system* if $\beta^k_{ij}=\log r^k_{ij}$ for any $(i, k, j) \in SAS$ and $t(r)=r$ in the recursive additive DP $\{S, A, p, r, \beta, t\}$. We call this dynamic programming "logarithmic additive DP". The reward system of logarithmic additive DP is just described by Fig. 3 except for replacing $\textcircled{D}$ by $\textcircled{L}$. Here, $\textcircled{L}$ means

$$\begin{tikzcd} & w \arrow[d] \\ (\log u)v \arrow[r] & \textcircled{L} \arrow[d] \\ & (\log u)(\log v)w \end{tikzcd}$$

. Then, we have

$$V(h)=r_1+(\log r_1)r_2+(\log r_1)(\log r_2)r_3$$
$$+\cdots+(\log r_1)(\log r_2)\cdots(\log r_{n-1})r_n+\cdots,$$

where $h=(s_1, a_1, s_2, a_2, \cdots)$, $r_k=r^{a_k}_{s_k s_{k+1}}$ for $k \geqq 1$. Moreover, if $t(r)=\log r$, $\beta^k_{ij}=\log r^k_{ij}$ for any $(i, k, j) \in SAS$ in recursive additive DP, we have another logarithmic additive

DP where objective function is the expected return of the function

$$V'(h) = \log r_1 + (\log r_1)(\log r_2) + (\log r_1)(\log r_2)(\log r_3)$$
$$+ \cdots + (\log r_1)(\log r_2) \cdots (\log r_n) + \cdots .$$

We call this dynamic programming " modified logarithmic additive DP". If $1 \leqq r_{ij}^k < e$ for any $(i, k, j) \in SAS$, then Assumption (I) is satisfied for these logarithmic additive DP's. The following data satisfies Assumption (I).

TABLE 4.5.1.
Data for logarithmic additive DP.

state	action	transition probability			stage-wise reward			generalized accumulator		
i	k	p_{i1}^k	p_{i2}^k	p_{i3}^k	r_{i1}^k	r_{i2}^k	r_{i3}^k	β_{i1}^k	β_{i2}^k	β_{i3}^k
1	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	2.3	2.7	2.4	$\beta_{ij}^k = \log r_{ij}^k$ for (i, k, j)		
	2	$\frac{1}{16}$	$\frac{3}{4}$	$\frac{3}{16}$	2.7	2.3	2.6			
	3	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{5}{8}$	2.5	2.4	2.6			
2	1	$\frac{1}{2}$	0	$\frac{1}{2}$	2.7	2.3	2.4			
	2	$\frac{1}{16}$	$\frac{7}{8}$	$\frac{1}{16}$	2.6	2.4	2.7			
	3	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	2.4	2.1	2.5			
3	1	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$	2.6	2.5	2.7			
	2	$\frac{1}{8}$	$\frac{3}{4}$	$\frac{1}{8}$	2.7	2.6	2.4			
	3	$\frac{3}{4}$	$\frac{1}{16}$	$\frac{3}{16}$	2.6	2.7	2.5			

The PIA yields optimal policies associated with logarithmic additive DP, discounted DP with $\beta_{ij}^k \equiv \beta = 0.95$ and discounted DP with $t(r) = \log r$, $\beta_{ij}^k \equiv 0.95$ respectively.

TABLE 4.5.2.
Solution by the PIA for logarithmic additive DP.

v_1		19.2080		
v_2		19.1090		
v_3		19.7274		
	↗		↘	
	3		3	
f	1		1	STOP
	1		1	

TABLE 4.5.3.
Solution by the PIA for discounted DP with $\beta^k_{ij} \equiv 0.95$.

v_1		51. 7705			
v_2		51. 7635			
v_3		51. 8369			
	3	↗ ↘		3	
g	1			1	STOP
	1			1	

TABLE 4.5.4.
Solution by the PIA for modified discounted DP
with $\beta^k_{ij} = 0.95$, $t(r) = \log r$

v_1		19. 0064		
v_2		19. 0025		
v_3		19. 0318		
	3	↗ ↘		3
h	1			1
	1			1

In these three DPs we have a common optimal stationary policy

$$f = g = h = \begin{pmatrix} 3 \\ 1 \\ 1 \end{pmatrix}.$$

Throughout above five examples, we have a remark that the ranges of the value r^k_{ij} for $(i, k, j) \in SAS$ are $(-\infty, 0)$, $[0, 1)$, $(1, \infty)$, $[1, e)$ corresponding to exponential additive, multiplicative additive, divided additive and logarithmic additive DPs respectively. By virtue of these examples, it is natural that we call $\beta = (\beta^k_{ij})$ a generalized accumulator.

5. Additional comments.

In this section we will show that the recursive additive DP satisfying Assumption (I) can be reduced to a DP with an absorved state. We will give some comments on recursive additive DP.

Let $\{S, A, p, r, \beta, t\}$ be a recursive additive DP satisfying Assumption (I). Let $0 \notin S$ be a fictitious state and $K_0 = \{1\}$. That is, the set of available actions at state 0 consists of a single action 1. We define $\bar{S}$, $\bar{A}$, $\bar{p}$ and $\bar{r}$ as follows:

$$\bar{S} = S \cup \{0\},$$

$$\bar{A} = \{K_0, \underbrace{A, A, \cdots, A}_{N \text{ factors}}\},$$

$$\bar{p}^k_{ij} = \begin{cases} \bar{p}^1_{00} = 1, & \\ \bar{p}^k_{i0} = 1 - \sum_{j \in S} \beta^k_{ij} p^k_{ij} & \text{for} \quad (i, k) \in S \times A, \\ \bar{p}^k_{ij} = \beta^k_{ij} p^k_{ij} & \text{for} \quad (i, k, j) \in S \times A \times S, \end{cases}$$

$$r_i^k = \begin{cases} 0 & \text{for} \quad (i, k) = (0, 1), \\ \sum_{j \in S} p_{ij}^k t(r_{ij}^k) & \text{for} \quad (i, k) \in S \times A. \end{cases}$$

Then in a new-defined DP$\{\bar{S}, \bar{A}, \bar{p}, \bar{r}\}$ the stage-wise reward $\bar{r} = (\bar{r}_i^k)$ does not depend on next state $j \in S$ but the set of available actions depends on current state $i \in S$.

LEMMA 4.1. *Let $\{S, A, p, r, \beta, t\}$ be a recursive additive DP satisfying Assumption (I). Then, $\{\bar{S}, \bar{A}, \bar{p}, \bar{r}\}$ is a DP with an absorved state 0. Furthermore any policy π in recursive additive DP$\{S, A, p, r, \beta, t\}$ can be extended to the policy $\bar{\pi}$ in reduced DP$\{\bar{S}, \bar{A}, \bar{p}, \bar{r}\}$ such that*

$$V(\pi)(i) = \bar{V}(\bar{\pi})(i) \quad on \quad S, \qquad \bar{V}(\bar{\pi})(0) = 0.$$

Conversely, any policy $\bar{\pi}$ in reduced DP$\{\bar{S}, \bar{A}, \bar{p}, \bar{r}\}$ can be regarded as a policy π in recursive additive DP$\{S, A, p, r, \beta, t\}$ such that

$$\bar{V}(\bar{\pi})(i) = V(\pi)(i) \quad on \quad S.$$

In this case $\bar{V}(\bar{\pi})(0) = 0$ is trivial consequence.

PROOF. Let P be $N \times N$ stochastic matrix and $\bar{P} = \begin{bmatrix} 1 & 0 \\ x & P \end{bmatrix}$ be $(N+1) \times (N+1)$ matrix. Then we have $\bar{P}^n = \begin{bmatrix} 1 & 0 \\ x_n & P^n \end{bmatrix}$ for each $n \geqq 1$. The proof is an immediate consequence of this result. The details are ommitted.

It should be noted that the generalized accumulator $\beta = (\beta_{ij}^k)$ in recursive additive DP is embedded into the transition law $\bar{p} = (\bar{p}_{ij}^k)$ of reduced DP. Hence it is rather difficult to get five examples in Section 4 from the reduced one.

There is a kind of sensitivity analysis as follows: Is there any relation between general additive DP$\{S, A, p, r, \beta\}$ and recursive additive DP$\{S, A, p, r, \beta, t\}$? In other words, what is a sufficient condition on $t = t(\cdot)$ for optimal policy in the former to remain optimal in the latter? The analysis of this sensitivity will require a detail analysis of Howard's policy iteration algorithm.

Acknowledgement. The author wishes to express his hearty thanks to Professor T. Kitagawa and Doctor N. Furukawa for some useful comments.

References

[1] R. BELLMAN, Dynamic Programming. Princeton Univ. Press. (1957).

[2] D. BLACKWELL, *Discrete Dynamic Programming.* Ann. Math. Stat., **33**, (1962), 719-726.

[3] D. BLACKWELL, *Discounted Dynamic Programming.* Ann. Math. Stat., **36**, (1965), 226-235.

[4] D. BLACKWELL, *Positive Dynamic Programming.* Proc. Fifth Berkeley Symp. Math. Statist. Probability, Vol. I, Univ. of California, Press. Berkeley, California, (1967), 415-418.

[5] N. FURUKAWA and S. IWAMOTO, *Markovian Decision Processes with Recursive Reward Functions.* Bull. Math. Statist., **15**, No. 3-4 (1973), 79-91.

[6] R. A. HOWARD, Dynamic Programming and Markov Processes. M. I. T. Press, Cambridge, Massachusetts, (1960).

[7] R. E. STRAUCH, *Negative Dynamic Programming.* Ann. Math. Stat., **37** (1966), 871-890.