JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS 79, 489–501 (1981)

Some Comments on Preference Order Dynamic Programming models

MOSHE SNIEDOVICH*

IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598

Submitted by E. S. Lee

A simple deterministic dynamic programming model is used as a general framework for the analysis of stochastic versions of three classical optimization problems: knapsack, traveling salesperson, and assembly line balancing problems. It is shown that this model can provide an alternative to the preference order models proposed for these problems. Counterexample to the optimality of the preference order models are presented.

1. INTRODUCTION

The preference order dynamic programming models developed by Mitten [1] and Sobel [2] provide an extremely flexible framework for formulation and analysis of sequential decision problems. Unlike standard dynamic programming models such as those developed by Bellman [3], Aris [4], Nemhauser [5], Mitten [6], Yakowitz [7], Hinderer [8], Kreps [9, 10] and others, the preference order models can be used in situaions where the optimality criterion is not based on real valued objective functions.

Attempts have been made to use the preference order models in the context of stochastic problems in which the optimality of the solution is based on probabilistic criteria related either to the reward or the constraints. For example, Kao [11] proposed such a model for a probabilistic constrained stochastic assembly line balancing problem and a similar model for a stochastic traveling salesperson problem with a probabilistic type objective function (Kao [12]). Also, Steinberg and Parks [13] proposed such a model for a knapsack problem with a probabilistic type objective function.

The objective of this paper is to show that the stochastic problems studied in [11–13] can be formulated as simple deterministic final state dynamic programming problems. In this sense the use of preference order models in

* Currently at the National Research Institute for Mathematical Sciences, CSIR, Pretoria 0001, South Africa.

0022-247X/81/020489–13$02.00/0

[11–13] is somewhat "artificial" for the problems under consideration can be analyzed in the context of Bellman's [3] model in which the optimality of the solutions is based on a real valued objective function.

We use Bellman's simple model as a basis for the construction of counterexamples to the optimality of the solutions derived by the procedures proposed in [11–13].

2. Final State Model

Under consideration is the simple deterministic sequential decision model introduced by Bellman [3, pp, 82, 83]. The truncated version of this model can be specified by the following elements:

(1) A sequence $\mathbf{N} = (n: n = 1, 2, \ldots, N)$ of *decision stages* and a *final stage*, $n = N + 1$, which is preserved for the description of the final state of the process.

(2) A nonempty state space S.

(3) A nonempty decision space D and a collection $(D_n : n \in \mathbf{N})$ of maps from S to the set of all the subsets of D such that $D_n(s)$ is interpreted as the set of *feasible decisions* associated with stage $n \in \mathbf{N}$, and state $s \in S$.

(4) A sequence $(T_n : n \in \mathbf{N})$ of *transformations* from $S \times D$ to S so that $s' = T_n(s, d)$ is interpreted as the state resulting at stage $n + 1$ if the decision $d \in D$ is applied to state $s \in S$ at stage n.

(5) A real valued *objective function*, g, defined on S.

The objective is to determine a sequence of decisions $(d_1, d_2, \ldots, d_N)$ so as to optimize the objective function which is applied to the final state of the process. That is, if the initial stae is s_1 and the decisions $(d_1, d_2, \ldots, d_N)$ are applied sequentially, starting at stage $n = 1$, the states $s_1, s_2 = T_1(s_1, d_1), \ldots, S_{N+1} = T_N(s_N, d_N)$ are generated and the total "reward" $g(s_{N+1})$ is realized.

Formally the problem under consideration can be stated as

$$G(s^0) := \underset{(d_1, \ldots, d_N)}{\mathrm{opt}} \; g(s_{N+1}), \qquad s^0 \in S, \tag{1}$$

subject to

$$s_1 = s^0, \quad d_n \in D_n(s_n), \quad s_{n+1} = T_n(s_n, d_n), \qquad \forall\, n \in \mathbf{N} \tag{2}$$

where the initial state, s^0, is viewed as a parameter and "opt" is either "min" or "max".

Following standard dynamic programming arguments [3] we consider the recursion

$$G_n(s) := \underset{d \in D_n(s)}{\text{opt}} \; G_{n+1}(T_n(s, d)), \qquad s \in S_n, \tag{3}$$

starting at $n = N + 1$ with

$$G_{N+1}(s) := g(s), \qquad s \in S_{N+1}, \tag{4}$$

where the state spaces $(S_n : n \in \mathbf{N} \cup \{N + 1\})$ are determined as follows: $S_1 := \{s^0\}$ and

$$S_{n+1} := \{s : s = T_n(s', d), s' \in S_n, d \in D_n(s')\}, \qquad 1 < n \leqslant N. \tag{5}$$

The sequential structure of the problem and the fact that $G(s^0)$ depends only on the last state guarantee that if $D_n(s)$ is finite for every $n \in \mathbf{N}$ and $s \in S_n$ then $G(s^0) = G_1(s^0)$ and the dynamic programming solutions are optimal.

In the next section we consider a special case of the above model which will enable the analysis of stochastic problems.

3. Stochastic Problems

In order to enable the use of the above model in the context of stochastic problems we consider the case $S = W \times F$, where W is a nonempty set and F is the set of all the distribution functions associated with real valued random variables. Thus, if $s_{N+1} = (w, f) \in S$ is the final state generated by applying $(d_1, ..., d_N)$ to the initial state s^0, then f is the distribution function generated by s^0 and $(d_1, ..., d_N)$ so that $g(w, f)$ can be a real number determined by f. For example, $g(w, f)$ can be the expected value of the random variable induced by f or the probability that this variable is equal to or greater than a given value.

Since the dynamic programming recursion is uniquely determined by the objects N, s^0, $\{T_n\}$, $\{D_n\}$ and g, in each application it would suffice to specify these objects. It should be noted that as far as computation is concerned, often it is possible to replace the distribution functions, $f \in F$, by their parameters.

4. Applications

The stochastic version of the final state model enables the formulation of a variety of stochastic sequential decision problems. Three such problems are considered here. In the sequel let $*$ denote the convolution operator, I be the set of non-negative integers, and $f(C)$ denote the probability that the random variable induced by f is equal to or smaller than C. We also denote by φ^0 the degenerate distribution function having zero mean and variance, and by $\varnothing$ the empty set.

4.1. *Stochastic Knapsack Problem*

Under consideration is the stochastic knapsack problem studied by Steinberg and Parks [13]. It can be described as follows: There are K types of items denoted by $k = 1, 2,..., K$, each of which is characterized by its weight, w_k, and its volume, v_k. In the stochastic version under consideration $(w_1,..., w_k)$ are non-negative integers and $(v_1,..., v_k)$ are independent random variables having known distribution function $(\varphi_1,..., \varphi_k)$. The objective is to select items so as to maximize the probability that their total volume would be equal to or greater than a given constant C, subject to the constraint that their total weight would be equal to or smaller than a given positive integer w^0.

For this problem we set $W := \{w: w = 0, 1,..., w^0\}$, and $S := W \times F$ and specify the elements of the final state model as follows: $N = K$, $s^0 = (w^0, \varphi^0)$, opt $\equiv$ "max." and

$$D_n(x, f) := \{d: d \cdot w_n \leqslant w, d \in I\}, \qquad (w, f) \in S; \tag{6}$$

$$T_n(w, f, d) := (w - d \cdot w_n, d * \varphi_n^{*d}), \qquad d \in I, \quad (w, f) \in S; \tag{7}$$

$$g(w, f) := 1 - f(C), \qquad (w, f) \in S; \tag{8}$$

where in (7) φ_n^{*d} is the distribution obtained by convoluting φ_n with itself d times. Notice that by construction at stage n the state $s_n = (w, f)$ can be interpreted as follows: w is the weight available after selecting items of type $1, 2,..., n-1$ and f is the distribution function of the total volume of these items.

4.2. *Stochastic Traveling Salesperson Problem*

Under consideration is the stochastic traveling sales person problem studied by Kao [12] in which a salesperson must visit a set of K cities denoted by k, $k \in L := \{1, 2,..., K\}$. Each city must be visited once and only once and each tour starts and terminates at an origin point denoted by 0. The travel time between cities i and j is a random variable, having a known distribution function φ_{ij}. For simplicity it is assumed that the traveling times are stochastically independent. The objective is to determine a tour so as to maximize the probability that the total travel time would be equal to or smaller than a given constant C.

For this case we let $\mathbf{L}$ denote the set of all the subsets of L and set $W = L \times \mathbf{L}$, $S = W \times F$, so that the state $s = (i, l, f) \in S$ can be interpreted as follows: i the city we currently visit, l is the set of cities we have yet to visit and f is the distribution function of the total travel time from the origin point to city i through the cities $L - \{i\} - \{l\}$. The elements of the model are specifid by $N = K - 1$, $s^0 = (0, L, \varphi^0)$, opt $\equiv$ "max" and

$$D_n(i, l, f) := \{d: d \in l\}, \qquad (i, l, f) \in S, \tag{9}$$

$$T_n(i, l, f, d) := (d, l - \{d\}, f * \varphi_{i,d}), \qquad (i, l, f) \in S, \quad d \in l, \tag{10}$$

$$g(i, l, f) := [f * \varphi_{i,0}](C), \qquad (i, l, f) \in S. \tag{11}$$

Notice that at the final stage $n = N + 1$ the states $s_{N+1} = (i, l, f) \in S_{N+1}$ are such that l is the empty set; *i.e.*, we have already visited all the cities, so that in (11) we convolute f with φ_{i0}, which is the distribution function of the travel time from city i to the origin point.

4.3. *Stochastic Assembly Line Balancing Problem*

Under consideration is the stochastic assembly line balancing problem studied by Kao [11]. It can be described as follows: a set of K tasks denoted by $k \in L := \{1, 2, ..., K\}$ are to be processed by work stations placed serially along an assembly line. The processing time of each task is a random variable having a known distribution functions φ_k, and $(\varphi_1, ..., \varphi_K)$ are assumed to be independent. The objective is to assign tasks to work stations so as to minimize the total number of work stations required for the processing of the tasks, subject to two constraints: A precedence constraint is required, specified by a precedence relation B so that for each task k the set $B(k)$ is a subset of L interpreted as the immediate predecessors of k. That is, task k can be processed only after all the elements of $B(k)$ have been processed. It is also required that each station will complete the processing of the tasks assigned to it in C or less units of time with probability greater than or equal to α.

For this case we set $\mathbf{W} = L \times \mathbf{L}$, and $S = W \times F$, where $\mathbf{L}$ is the set of all the subsets of L so that the state (m, l, f) can be interpreted as follows: m is the number of stations to which we have already assigned tasks, l is the set of tasks we have already assigned and f is the distribution function of the processing time of the tasks assigned to the last station, i.e. the mth station. In order to guarantee the existence of feasible solutions it is assumed that for every k the condition $\varphi_k(C) \leqslant \alpha$ is satisfied and that there exists a permutation of tasks which satisfies the precedence constraint specified by B.

The elements of the final state model are specified by $N = K$, $s^0 = (0, \phi, \varphi^0)$, opt $\equiv$ "max", and

$$D_n(m, l, f) := \{d\colon d \in L, B(d) \subseteq l\}, \qquad (m, l, f) \in S, \tag{12}$$

$$T_n(m, k, f, d) := \begin{cases} (m, l \cup \{d\}, f * \varphi_d) & \text{if} \quad [f * \varphi_d](C) > \alpha \\ (m+1, l \cup \{d\}, \varphi_d) & \text{if} \quad [f * \varphi_d](C) \leqslant \alpha, \end{cases} \tag{13}$$

$$g(m, l, f) := m, \tag{14}$$

observing that for every stage n the states $s_n = (n, l, f) \in S_n$ are such that m is not greater than n, and l is a collection of $n-1$ tasks satisfying the precedence relations.

It should be emphasized that the final state recursion wold be simplified because (14) could be constructed by an additive recursion which does not require the incorporation of m in the state variables.

5. Discussion

Experience with dynamic programming has shown that in practice the dynamic programming recursion is often constructed on the basis of intuitive arguments in which Bellman's [3, p. 83] principle of optimality is invoked to justify the optimality of the solutions. As indicated by the counterexamples developed by Elmaghraby [14] and Erlenkotter [15] certain difficulties may be encountered if the principle of optimality is invoked "axiomatically."

Similarly, the question regarding what types of problems dynamic programming can handle is not trivial because the answer to it depends on how one defines the elements of the sequential decision problems. For example, Sniedoviich [16] shows that contrary to Askew's [17] and Rossman's [19] conclusions, dynamic programming can handle probabilistic constraints defined over the entire "life of the process."

As indicated by Mitten [1, p. 43] and Sobel [2, pp. 967, 968] preference order dynamic programming models are suitable for the analysis of sequential decision processes in which the optimality of the solutions is not based on real-valued objective functions. In this sense, it is not clear why preference order models should be used for the analysis of the problems studied in [11–13] for, as indicated in the preceding section, these problems can be formulated as simple deterministic final state dynamic programming problems having real-valued objective functions. Moreover, as will be shown in the sequel, preference order models proposed in [11–13] may fail to provide the optimal solutions.

6. Preference Order Models

Since it is always possible to redefine the final state model as a preference order model according to Sobel's [2] formulation, the question is not whether or not the above problems constitute legitimate preference order dynamic programming problems but rather what the advantage is, if any, of such a formulation. For this reason we briefly describe the preference order models proposed in [11–13].

6.1. *Stochastic Knapsack Problem*

The preference order recursion proposed by Steinberg and Parks [13] for the stochastic knapsack problem discussed in (4.1) can be described as follows: For every positive integer $w \in I$ define $X_n(w) := \{x: x \cdot w_n \leqslant w, x \in I\}$ so that $X_n(w)$ is the set of all the feasible decisions available while selecting items of type n, given that we still have w units of weight.

Step 1. For $n = 1$ solve

$$f_1(w) := \mathop{\perp}_{x \in X_1(w)} \{\varphi_1^{*x}\}, \qquad w \in W. \tag{15}$$

Step 2. For $1 < n \leqslant N$ solve

$$f_n(w) := \mathop{\perp}_{x \in X_n(w)} \{\varphi_n^{*x} * f_{n-1}(w - x \cdot w_n)\}, \qquad w \in W, \tag{16}$$

where $\perp$ is a preference order operator on the set of all the subsets of F with values in F such that for any subset Q of F, the distribution $q^0 = \perp[Q]$ satisfies the condition

$$[q^0 * \varphi](C) \geqslant [q * \varphi], \qquad \forall q \in Q, \quad \varphi \in \mathbf{F}, \tag{17}$$

where $\mathbf{F}$ is the set of all the distribution function that can be generated by convoluting the elements $(\varphi_1, \varphi_2, ..., \varphi_N)$.

As indicated by Steinberg and Parks [13] the operator $\perp$ is transitive and reflexive but not necessarily complete. If, however, $\perp$ is complete, then Mitten's [1] monotonicity condition is satisfied and $f_N(w^0)$ is an optimal solution.

Obviously, because in (17) we convolute the arguments of $\perp$ with all the elements of $\mathbf{F}$ the completeness assumption is rather restrictive and the computational requirements can be significant. On the other hand, since the recursion is based on the elements $w \in W$, (16) is more efficient than the final state recursion as far as storage requirements are concerned. The difficulty here is that in general $\perp$ is not complete and hence the recursion is not always well defined.

6.2. *Stochastic Traveling Salesperson Problem*

Consider the problem described in (4.2) and let A_k denote the set of all the subsets of $\{0\} \cup L$ which include the origin point 0 and k distinct cities. That is, $a_k \in A_k$ is a set of $k+1$ elements including 0 and k elements of L, so that by construction $A_0 = \{0\}$ and $A_K = \{0 \cup L\}$. Define

$$W_{k+1} := \{(i, l): i \in L, l \in A_{k+1}, i \notin l\}, \qquad 0 < k < K, \tag{18}$$

with $W_0 := \{(i, \phi): i \in L\}$, so that $w = (i, l) \in W_k$ can be interpreted as follows: i is the city we currently visit and l is the cities we have yet to visit.

The preference order dynamic programming recursion proposed by Kao [12] can be described as follows:

Step 1. For $k = 0$ solve

$$G_0(i, \phi) := \phi_{i,0}, \qquad (i, \phi) \in W_0. \tag{19}$$

Step 2. For $0 < k \leqslant K$ solve

$$G_k(i, l) := \underset{d \in l}{\perp} \{\varphi_{i,d} * G_{k-1}(d, l - \{d\})\}. \tag{20}$$

where $\perp$ is a preference order operator on the set of all the subsets of F with values in F satisfying the following condition: For any subset Q of F the distribution function $q^0 = \perp[Q]$ is such that

$$[q^0 * \varphi](C) \leqslant [q * \varphi](C), \qquad \forall q \in Q, \quad \varphi \in F. \tag{21}$$

As indicated by Kao [12] the operator $\perp$ is transitive and reflexive but not necessarily complete because in (21) its arguments are convoluted with all the elements of F. However, if $\perp$ is complete then it satisfies Mitten's [1] monotonicity condition and therefore the optimality of the preference order solution is guaranteed. That is, $G_K(0, L)$ is the optimal distribution function of the total travel time.

Obviously, since in (21) the arguments of $\perp$ are convoluted with all the elements of F, the completeness assumption is extremely restrictive and the computational requirements can be extremely demanding. In view of these difficulties, Kao [12, p. 1036] proposes that $\perp$ can be modified as follows: While applied in (20) its arguments could be convoluted only with the elements of the set $\{\varphi_{jl}: j \in L, \ j \notin l\}$. In this case the completeness assumption is less restrictive and the recursion is more efficient. However, as will be shown in the sequel under this condition, Mitten's [1] monotonicity condition is not implied by the completeness condition and hence the proposed procedure may generate non-optimal solutions.

6.3. *Assembly Line Balancing Problem*

Consider the problem described in (4.3) and the objects $(a_1, \ldots, a_J)$, Y, Δ and $\perp$ defined as follows:

(1) a_1 is an element of L such that $B(a_1) = \varnothing$, a_J is the set L and for every $1 < j < J$, the set a_j is a subset of L such that for every $i \in a_j$ the set $B(i)$ is equal to a_{j^0} for some $j^0 < j$. Details regarding how such a sequence can be generated can be found in Kao [12, p. 1088].

(2) Y is a map from the set of all the subsets of L to itself such that for every a_j, $1 < j \leqslant J$, the set $Y(a_j)$ consists of all the elements, k, of a_j such that $a_j - \{k\} = a_{j^0}$, for some $1 \leqslant j^0 < j$.

(3) Δ is the map from $I \times F \times L$ to $I \times F$ defined by

$$\Delta(m, f, i) = \begin{cases} (m, f * \varphi_i), & \text{if } \quad [f * \varphi_i](C) \geqslant \alpha, \\ ((m+1), \varphi_i), & \text{if } \quad [f * \varphi_i](C) < \alpha. \end{cases} \tag{22}$$

(4) $\perp$ is a preference order operator such that when applied to sets of the form $Q = \{(m_i, f_i)\colon\ i = 1, 2, \ldots, q\}$, the element $(m_{i^0}, f_{i^0}) = \perp[Q]$ is selected according to the criteria

$$m_{i^0} \leqslant m_i, \qquad \forall\, i = 1, 2, \ldots, q, \tag{23}$$

and if i^0 is not unique then the condition

$$[f_{i^0} * \varphi]^{-1}(\alpha) \leqslant [f_{i'} * \varphi]^{-1}(\alpha), \qquad \forall\, \varphi \in F, \tag{24}$$

is used to resolve the tie between all elements i' for which $m_{i^0} = m_{i'}$, where $\varphi^{-1}(\alpha) := \inf\{t\colon \varphi(t) \leqslant \alpha\}$.

The proposed preference order dynamic programming recursion proposed by Kao [11] can be described as follows:

Step 1. Set

$$G(a_1) := (1, \varphi_{a_1}). \tag{25}$$

Step 2. For $1 < j \leqslant J$ solve

$$G(a_j) := \mathop{\perp}_{i \in Y(a_j)} \{\Delta(G(a_j - \{i\}), i)\}. \tag{26}$$

As indicated by Kao [11], if $\perp$ is complete then $(m, f) = G(a_J)$ is optimal in that m is the minimum number of work stations required for the processing of the tasks, subject to the precedence and probabilistic constraints. However, since in (24) the convolution is over all the elements of F the completeness assumption is restrictive and the computational requirements can be extremely demanding.

Consequently, Kao [11, p. 1099] proposes that while applying (26), in order to resolve a possible tie in (23), there is no need to convolute φ_i with all the elements of F and that the convolution can be restricted only to the elements $\{\varphi_i : i \notin a_j,\ i \in L\}$. However, as will be shown in the sequel in this case, the completeness of $\perp$ does not necessarily guarantee the validity of Mitten's [1] monotonicity condition and hence the recursion may provide non-optimal solutions.

6.4. *Summary*

The completeness assumption and the computational requirements associated with the proposed preference order recursions are extremely restrictive. In the sequel we show that the modified procedures proposed in [11–13] for overcoming these difficulties may generate non-optimal solutions.

7. Counterexamples

The fact that in the final state model it was necessary to incorporate distribution functions in the state variables could be used as an argument to question the optimality of the proposed preference order procedures. It should be noted that in order for the preference order procedures to be computationally feasible it is necessary that $\perp$ be complete. However, since in each case it is proposed to modify the original structure of $\perp$ so as to reduce the amount of computation, it is no longer obvious that if the simplified version of $\perp$ is complete the procedure will generate optimal solutions. The counterexamples to be presented below were constructed on the basis of the final state model which is guaranteed to provide optimal solutions.

7.1. *Knapsack Problem*

Consider the numerical example presented by Steinberg and Parks [13, pp. 143, 144]. It is defined by $N = 10$, $C = 30$, the variables $(w_1, w_2, \ldots, w_N)$ and the normal distributions $(\varphi_1, \varphi_2, \ldots, \varphi_N)$ whose means $\{\mu_n\}$ and variances $\{\sigma_n^2\}$ are specified in Table I.

TABLE I

Parameters for Numerical Example

n	1	2	3	4	5	6	7	8	9	10
w_n	5	7	11	9	8	4	12	10	3	6
μ_n	7	12	14	13	12	5	16	11	4	7
σ_n^2	15	20	15	10	8	20	8	15	20	25

For $w^0 = 25$ the optimal solution obtained by the preference order procedure proposed by Steinberg and Parks [13] is $x_2 = 1$, $x_4 = 2$, $x_n = 0$, $\forall n \neq 2, 4$. This solution generates a normal distribution with mean $\mu = 38$ and variance $\sigma^2 = 40$. For this distribution the probability that the total volume would be equal to or greater than $C = 30$ is 0.896.

On the other hand, the final state model generates the solution $x_4 = 1$, $x_5 = 2$, $x_n = 0$, $\forall n \neq 4, 5$. This solution generates the normal distribution with mean $\mu = 37$ and variance $\sigma^2 = 26$, for which the probability that the total volume would be C or more is 0.915. Obviously, the preference order solution is not optimal although the recursion is well defined.

7.2. *Traveling Salesperson Problem*

Consider the traveling salesperson problem specified by $N = 4$, $C = 70$ and the normal distributions $\{\varphi_{ij}\}$ whose means and variances are specified by

$$\{\mu_{ij}\} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} - & 70 & 70 & 70 & 13 \\ 70 & - & 10 & 15 & 70 \\ 8 & 70 & - & 20 & 70 \\ 14 & 70 & 24 & - & 70 \\ 70 & 8 & 70 & 70 & - \end{bmatrix} \end{array}, \qquad \{\sigma_{ij}^2\} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} 0 & 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} - & 0 & 0 & 0 & 0 \\ 0 & - & 6 & 2 & 0 \\ 2 & 0 & - & 12 & 0 \\ 9 & 0 & 2 & - & 0 \\ 0 & 1 & 0 & 0 & - \end{bmatrix} \end{array}.$$

For this problem Kao's [12] procedure generates the tour (0, 4, 1, 3, 2, 0) which generates the normal distribution with mean $\mu = 68$ and variance $\sigma^2 = 7$, yielding a probability of 0.775 for the travel time to be less than or equal to $C = 70$.

On the other hand, the final state model generates the tour (0, 4, 1, 2, 3, 0) which generates the normal distribution with mean $\mu = 65$ and variance $\sigma^2 = 28$, yielding a probability of 0.829 for the tour to be completed in $C = 70$ units of time or less. Obviously, the preference order solution is not optimal.

7.3. *Assembly Line Problem*

Consider the stochastic assembly line problem specified by $N = 5$, $C = 14.3$, $\alpha = 0.95$ the normal distributions $(\varphi_1, ..., \varphi_N)$ and the preference relation B specified in Table II.

For this case Kao's [11] procedure generates the assignment ((1, 3), (2, 4), (5)) which requires three stations while the final state model generates the assignment ((1, 2), (3, 4, 5)) which requires only two stations. Obviously, the preference order solution is not optimal.

TABLE II

Parameters for Numerical Example

Task n	$B(n)$	μ_n	σ_n^2
1	ϕ	8	0
2	$\{1\}$	4	0
3	$\{1\}$	3	2
4	$\{2, 3\}$	4	1
5	$\{4\}$	4	1

8. Conclusions

The preference order procedures proposed in [11–13] should be used with caution because they may generate non-optimal solutions. While it is true that if the original form of the preference order operator is used the solutions are optimal, this form of the operator is extremely restrictive because of the completeness assumption and the computational requirements.

On the other hand, these procedures can be used as efficient heuristic methods, with the understanding that the optimality of the solutions is not guaranteed. From the methodological point of view it should be emphasized that the final state model could be used as a general framework for the analysis of these problems and therefore the use of the preference order models is somewhat artificial.

More details regarding the non-optimality of the preference order models and the counterexamples presented in this paper can be found in [19–21].

References

1. L. G. Mitten, Preference order dynamic programming, *Management Sci.* **21** (1974), 43–46.
2. M. J. Sobel, Ordinal dynamic programming, *Management Sci.* **21** (1975), 967–975.
3. R. Bellman, "Dynamic Programming," Princeton Univ. Press, Princeton, N. J., 1957.
4. R. Aris, "Discrete Dynamic Programming," Blaisdell, New York, 1964.
5. G. L. Nemhauser, "Introduction to Dynamic Programming," Wiley, New York, 1966.
6. L. G. Mitten, Composition principles for synthesis of optimal multistage processes, *Operations Res.* **12** (1964), 610–619.
7. S. J. Yakowitz, "Mathematics of Adaptive Control Processes," Amer. Elsevier, New York, 1969.
8. K. Hinderer, "Foundation of Non-stationary Dynamic Programming with Discrete Time Parameters," Springer–Verlag, New York, 1970.
9. D. M. Kreps, Decision problems with expected utility criteria. I. Upper and lower convergent utility, *Math. Operations Res.* **2** (1977), 45–53.

10. D. M. KREPS, Decision problem with expected utility criteria. II. Stationary, *Math. Operations Res.* **2** (1977), 266–274.
11. E. P. KAO, A preference order dynamic program for stochastic assembly line balancing, *Management Sci.* **22** (1976), 1097–1104.
12. E. P. KAO, A preference order dynamic program for a stochastic traveling salesman problem, *Operations Res.* **26** (1978), 1033–1045.
13. E. STEINBERG AND M. S. PARKS, A preference order dynamic program for a knapsack problem with stachastic rewards, *J. Opl. Res. Soc.* **30** (1979), 141–147.
14. S. E. ELMAGHRABY, The concept of "state" in discrete dynamic programming, *Math. Anal. Appl.* **29** (1975), 523–557.
15. D. ERLENKOTTER, Sequencing expansion projects, *Operations Res.* **21**, No. 1 (1973), 542–553.
16. M. SNIEDOVICH, On the reliability of reliability constraints, RC 7080, IBM Thomas J. Watson Research Center, Yorktown Heights, N. Y., 1978.
17. A. J. ASKEW, Chance constrained dynamic programming and the optimization of water resources systems, *Water Resour. Res.* **10** (1974), 1099–1106.
18. L. A. ROSSMAN, Reliability-constrained dynamic programming and randomized release rules in reservoir management, *Water Resour. Res.* **13** (1977), 247–255.
19. M. SNIEDOVICH, Preference order stochastic assembly line balancing problem: A counterexample, RC7904, IBM Thomas J. Watson Research Center, Yorktown Heights, N. Y., 1979.
20. M. SNIEDOVICH, Preference order stochastic traveling salesperson problem: A counterexample, RC7905, IBM Thomas J. Warson Research Center, Yorktown Heights, N. Y., 1979.
21. M. SNIEDOVICH, Preference order stochastic knapsack problem: A counterexample, RC 7766, IBM Thomas J. Watson Research Center, Yorktown Heights, N. Y., 1979.