Journal of the Nigerian Association of Mathematical Physics
***Volume* 17 (November, 2010), pp 383 - 386**

Dynamic Programming for Production Planning; an application of Dijkstra's Model

[1]Ojarikre H.I and [2]Ighedo .O
[1,2]Department of Mathematics and Computer Science
Delta State University, Abraka.

Abstract

In this work, we present Dijkstra's model as it relates to Richard Bellman equation. This model finds the routes by cost precedence. It is numerically illustrated using the model to obtain the overall optimal policy that minimizes the total cost. We also describe an approach for exploiting structure in Markov decision processes with continuous state variables.

Key words: Dijkstra's Model, State variables, Bellman equation, Sequential Decision Problems SDP, Markov Decision Processes.

1.0 Introduction

Dijkstra's algorithm is one of the most popular algorithms in computer science and operations research. This model was strongly inspired by Bellman's principle of optimality and both conceptually and technically constitute a dynamic programming successive approximation procedure per excellence. Gass and Harris [5] in the Encyclopedia of Operations research and management science described this model as a "node labeling greedy algorithm" and a greedy algorithm described as "a heuristic algorithm, that at every step selects the best choice available at each step without regard to future consequences".

The objective of this paper is to present Dijkstra's model and its relationship to the methods and techniques of dynamic programming as presented by [3]. Riano [10] proposed the stochastic production planning model for multi-period, multi-product system, where the lead time to produce a product may be random. The model determines release times for products that guarantee the requirements in each time period are met with desired probabilities at a minimum cost. Riano et al (2003) further described how an advanced planning model (stochastic production planning) can be integrated with discrete event simulation model to make the simulations more realistic and informative and compared the performance of Stochastic production planning model with the material requirements planning model in a simulation study. Graves [7] identified a set of tactical decisions that are critical for handling uncertainty in production planning and described how these tactics can be incorporated into production planning systems as a proactive counter measures to address various forms of uncertainty.

Dynamic programming is a recursive method for solving sequential decision problems SDP. SDP can assume two types: discrete time and stochastic types. Different researchers have discovered backward induction as a way to solve sequential decision problems involving risk and uncertainty ([1] and [4]). They used backward induction to find optimal decision rules in games against nature and subgame perfect equilibrium of dynamic multi-agent games. Dynamic programming techniques can be used to develop a model for production planning and inventory control problem. Smith and Pass [13] investigated the problem of finding the best potential partner from a fixed number of potential partners using dynamic programming approach. Dijkstra's algorithm is a clever method for solving dynamic programming functional equation for the shortest path problem given that the arc lengths are non negative. Meyn [9] applied this technique in solving complex networks problem. We are going to apply it to production planning problem.

[1]Corresponding author: Tel; +2348034228587

2. **Dijkstra's Model**

Dijkstra's algorithm can be described as an iterative procedure that repeatedly attempts to improve an initial approximation {F(j)} of the exact values of {f(j)}. The initial approximation is F(i)=0 and F(j)= ∞ for j=2, 3, …, n

The method was designed for minimum cost path problem where negative distances are not allowed but cycles are
permitted. It finds the route by cost precedence, that is, by selecting option with minimum F(j) value.

If F(1)<∞, then

U=u\{k} (1)

F(j) = min{f(j), D(k , j) + F(k)} , j in u∩s(k) (2)

K = arg min {f(j): j in u} (3)

Where U is the set of cities that are not yet processed. S(j) denote the set of immediate successors of city j. The computation ends if either the destination city n is next to be processed, that is, k=n or F(k) = ∞. This implies that at this iteration F(j) = ∞ for all the cities j that are yet to be processed and the conclusion is that these cities cannot be reached from city 1. Therefore, when city k is processed, the {F(j)} values of its immediate successors that have not yet been processed are updated in accordance with 2. After that, the next city to be processed is one whose F(j) value is the Smallest over all the unprocessed cities using (3). After city k is processed, it is immediately deleted from being used. The above formulation does not explain how tours are constructed. It only updates the length of tours.

2.1 Bellman equation

A Bellman equation is a necessary condition for optimality associated with the mathematical optimization method known as dynamic programming. The principle asserts that if the policy function is optimal for the infinite summation, then it must be the case that whatever the initial state and decision, the remaining decisions must constitute an optimal policy with regard to the state resulting from that first decision. In addition, a Bellman equation refers to a recursion for expected return (reward). Richard Bellman is widely accredited with recognizing the common structured dynamic problem and showing how backward induction can be applied to solve huge class of sequential decision problems (SDP) under risk and uncertainty. The term SDP was later change to dynamic problem.

Given an appropriate initial condition

$x_0 \in X$, the canonical infinite horizon dynamic programming problem is

$$\underset{\{x_{t+1}\}_{t=0}^{\infty}}{Max} \sum_{t=0}^{\infty} \beta^t F(x_t,\ x_{t+1}) = V(x_0)$$

Subject to the constraints

$$\{x_{t+1}\} \in \Gamma(x_t), \forall\ t = 0, 1, 2, \qquad (4)$$

x is a vector of state and control variables, indexed by discrete time t. $0 \le \beta \le 1$ is the discount factor [2]. Equation (4) is known as Bellman equation. The recursive restatement of (4) equation is

$$V(x) = \max_{y \in \Gamma(x)} [F(x, y) + \beta V(y)], \forall x \in X \qquad (5)$$

The function V that solves the Bellman equation is called the value function. The value function describes the optimized value of the problem as a function of state variable *x*. The function V(y) that describes the optimal choice as a function of the state is called the policy function. The expected reward for being in a particular state s following some fixed policy π is related as reward

$$V^{\Pi}(s) = R(s) + \beta \sum_{s'} P(s'/s, \Pi(s)) V^{\Pi}(s') \qquad (6)$$

R(s) is the return function. (6) describes the expected reward for taking the action prescribed by some policy π. Relating equations (3) and (4) , we have

$F_t(y_t) = \min f(y_{t-1}, x_t) + R(y_t)$

$0 \le x_t \le y_{t-1}$

Subject to

$$\{x_{t+1}\} \in \Gamma(x_t), \forall t = 0, 1, 2, \ldots.., n \qquad (7)$$

[1]Corresponding author: Tel; +2348034228587

F(y,x) is the function of the state variable while R is the return function. For a deterministic SDPs, the transition probabilities are usually degenerate. We can represent the state variables by deterministic function

$$x_{t+1} = f_k\left(s'_{t,} s_t\right) \quad (8)$$

2.2 Continuity property*:* For each j in Dijkstra's model, if $s^n_{j-1} \to s_{j-1}$ in S_{j-1}, and $y^n_j \to y_j$ in Y_j, as $n \to \infty$, where $y^n_j \to Y_j(s^n_{j-1})$, for all n, then $y_j \to Y_j(s_{j-1})$.

In this case, each F_j is the closed (hence, compact) graph of the set-valued mapping s_{j-1} into $Y_j(s_{j-1})$ in the compact space $S_{j-1} \times Y_j$. We require that $S_j = f_j(F_j)$ for all $j =$ 1, 2, . . . ,n so that, in particular, $S_1 = f_1(F_1)$, where $F_1 = \{s^0\} \times Y_1(s^0)$. Thus, each S_j consists of the set of feasible points, that is, attainable states in period j (city) j.

2.3 Efficiency *(finite optimality)*. Let $x \in X$. Then x is *efficient* (relative to α) if, for each y *in X*, and for each N such that $s_N(y) = S_N(x)$, we have $C_N(x|\alpha) \le CN(y|\alpha)$. Also known as *finite optimality*, this criterion was originally introduced in a special case by [8], who called it *finite horizon clamped endpoint optimality*.

Let $X^e(\alpha)$ denote the subset of X consisting of efficient strategies. Then the efficient strategies exists, that is, Ø C$X^e(\alpha) \subseteq X$, provided each of the spaces Y_j and S_{j-1} is *discrete*. (Although [12] assumed that the period costs were uniformly bounded, which has no effect on the definition of efficient strategy.) Before continuing with our comparisons of optimality criteria, given a sufficient condition for efficient solutions to exist in the case of Y_j and S_{j-1}. Fix N, and for each $s \in S_N$, let $X_N(s)$ denote the set of N-horizon feasible strategies which attain state s at the end of period N, that is,

$$X_N(s) = \{x \in X_N : s_N(x) = s\} = s^{-1}{}_N(s).$$

Since s_N is continuous, we thus obtain a partition $\{X_N(s) : s \in S_N\}$ of X_N consisting of compact sets, as well as a set-valued mapping $s \to X_N(s)$ of S_N into X_N with compact, nonempty values. Now, for each N and $s \in S_N$, consider the optimization problem.

$$\min_{x \in X_N(s)} C_N(x/\alpha)$$

If we let $X^*_N(s|\alpha)$ denote the set of optimal solutions to this problem, then this set is a closed, nonempty subset of X_N. We thus obtain another compact-valued set mapping of S_N into X_N given by $s \in X^*_N(s|\alpha)$. If we define

$$\chi^*_N(\alpha) = \bigcup_{s \in S_n} X^*_N(s/\alpha) \ ,$$

so that $\chi^*_N(\alpha)$ are nonempty and nested downward and $\chi^*_N(\alpha) = \bigcap^{\infty}_{N=1} \chi^*_N$

then it is not difficult to see that the efficient solutions are precisely the elements of $\chi^*_N(\alpha)$, that is, $X^e(\alpha) = \chi^*_N(\alpha)$

The following gives a sufficient condition for the existence of efficient solutions—in the continuous state case.

2.4 Theorem : *If, for each N, the set-valued mapping s_→$X_N(s)$ is continuous , then efficient solutions exist, that is, $X^e(\alpha) \neq \emptyset$, and $X^e(\alpha)$ is compact, for all* $0 < \alpha \le 1$.

Proof. It follows from our hypothesis and [6], that the set-valued mapping s_∈ $X^*_N(s|\alpha)$ is upper semi-continuous . Consequently, the space $X^*_N(\alpha)$ is compact for each N. Hence, $X^*(\alpha)$ is the intersection of a descending sequence of compact, nonempty sets, and is thus, compact and nonempty.

This generalizes the existence result for efficient solutions established in [12] for the discrete state case.

3.0 Let MJ be a Bakery Outfit with its production planning period for one year broken into 4 quarters. Let the expected sale on quarterly basis be estimated as presenteds in the table below:

Table 1: Annual Estimate

Quarter No	Estimated sales units	Cumulative sales units
1	600	600
2	700	1300
3	500	1800
4	1200	3000

[1]Corresponding author: Tel; +2348034228587

The cost of producing X_n units is $50X_n$ naira and storage costs are estimated at 2.00 naira per unit per quarter. The planning problem consist of determining quarterly productions X_n and inventory to meet the sales requirement at minimum total cost.

Solution

Xn = production in the n-quarter, y_n = inventory at the end of the n-quarter. Let each quarter represent a stage. For any period, inventory at the end of the period y_n is

$Y_n = y_{n-1}+X_n-S_n$ Thus, the return function of the n^{th} quarter with respect to table1 is

$R_n(y_{n-1}, x_n) = 50x_n + 2y_{n-1}$

For stage 4

$F_1 = 50x_4 + 2y_3$ Subject to , $y_4 = y_3 +x_4 – 1200$

$= 60000 – 48y_3$

For stage 3

$F_2 = 50x3 + 2y2 +60000 -48y3$, subject to , $y3 =y2 + x3 – 500 \geq 0$, $x3 \geq 0$

$= -46y_2 + 94000$

For stage 2

$\mathbf{F}_3 = 50x_2 + 2y_1 – 94y_2 +94000$, subject to , $y_2 = y_1 + x_2 – 700$

$= -48y_1 +35000$

For stage 1

$F_4 = 50x_1 + 2y_0 – 48y_1 + 35000$ subject to $y_1 = y_0 + x_1 – 600$

$=2x_1 - 46y_0 + 63800.$

Now using backward substitution, we have that $x_1 = 600$ implies that that $y_1 = y_0 = 0$. $F_4 = 63800 – 1200 = \#62,600$

$Y_2 =0$, $y_3 = 300$, $y_4 = 0$. $x_1= 600$, $x_2 = 700$ $x_3 = 800$ and $x_4 = 9000$

Conclusion

Uncertainty in production sector is a major problem. The optimal pattern of production, storage and sales in a company guarantees growth and sustainable development in any environment. Dijkstra's model finds the routes by cost precedence obtaining the F(j) value with minimum cost. There are two important variables in dynamic programming problem, the state variables and the decision variables which act as the control variable. The reward or payoff function depends on the realized state and decision from that period. The idea of this model is based on the fact that for every minimum route, all costs are considered as positive numbers and that is why negative distances are not considered.

One major limitation of this model is that it can not be used in cases where the network under consideration is cyclic negative. For such cases, at least one pair of distinct city (i, j) using equation (2), one cannot compute the F(j) value before one computes the F(i) value neither can the F(i) value be computed before the F(j) value. However, this model is instructive in that it suggests that the solution of equation (2) can be carried out not just by the order dictated by the constraints but rather in an order dictated by the values of {f(j)}, that is, equation (3).

References

[1] Arrow, K.J, Harris, T and Marschak, J (1951), Optimal Inventory Policy, Econometrica 19, 250-172.

[2] Bellman, R.E (1952) On the theory of dynamic programming, Proceedings of National Academy of Sciences 38, pp716 - 719.

[3] Bellman, R.E (1957) Dynamic programming, Princeton University Press, Princeton, NJ

[4] Blackwell, D, Arrow K.. and Girshick, M.A (1949), Bayes and Minimax Solutions of Sequential Decision Problems Econometrica 17, 213-244.

[5] Gass, S.I. and Harris, C.M (1996), Encyclopedia of Operations Research and management science, Kluwer, Boston.

[6] Goldberg, R.R (1964), Methods of Real of Analysis, Blaisdell, Walthaam.

[7] Graves, S.C (2008) Uncertainty and Production Planning, MIT Press, Cambridge, MA.

[8] Halkim, H (1974), Necessary conditions for Optimal control problems with infinite Horizons, Econometrica 42, pp267 - 272

[9] Meyn, S.P (2007) Control Techniques for complex networks. Cambridge University Press, Cambridge.

[10] Riano,G Serforzo,R.F, Hackman, S.T, Ng, S.H and Chain, L.P (2002) A stochastic production Planning Model, Technical Report, National University of Singapore.

[1]Corresponding author: Tel; +2348034228587

[11] Riano,G, SerforzoR, F, Hackman, S.T, Ng, S.H and Chain, L.P (2003) Benchmarking of a stochastic production planning model in a simulation testbed, Proceedings of 2003 Winter simulation, pp1183-1191.
[12] Schochetman, I.E and Smith, R.L (1998). Existence and Discovery of average optimal solutions in deterministic infinite horizon optimal, Mathematics of Operation Research, vol 23, No 2, pages 416 – 432.
[13] Smith, D.K and Pass, M. (1999) Dynamic programming: an introduction, Millennium Mathematics Project, Chichester, UK
[14] Sniedovich, M. (1992), Dijkstra Algorithm; Dynamic Programming, Marcel Dekke, New York.

[1]Corresponding author: Tel; +2348034228587