Comput Manag Sci (2009) 6:447–457
DOI 10.1007/s10287-007-0052-1

ORIGINAL PAPER

Comparative studies on dynamic programming and integer programming approaches for concave cost production/inventory control problems

Hiroshi Konno · Takaaki Egawa · Rei Yamamoto

Published online: 29 March 2007

Abstract This paper is concerned with classical concave cost multi-echelon production/inventory control problems studied by W. Zangwill and others. It is well known that the problem with m production steps and n time periods can be solved by a dynamic programming algorithm in $O(n^4m)$ steps, which is considered as the fastest algorithm for solving this class of problems. In this paper, we will show that an alternative 0–1 integer programming approach can solve the same problem much faster particularly when n is large and the number of 0–1 integer variables is relatively few. This class of problems include, among others problem with set-up cost function and piecewise linear cost function with fewer linear pieces. The new approach can solve problems with mixed concave/convex cost functions, which cannot be solved by dynamic programming algorithms.

Keywords Concave cost inventory control problem · Dynamic programming algorithm · 0–1 integer programming

1 Introduction

Concave cost production and inventory control problem under deterministic demands is one of the classical and well studied problems in operations research.

H. Konno · T. Egawa
Department of Industrial and Systems Engineering, Chuo University, Tokyo, Japan

R. Yamamoto (✉)
Department of Industrial and Systems Engineering, Chuo University and Mitsubishi UFJ Trust Investment Technology Institute Co., Ltd, Tokyo, Japan
e-mail: yamamoto@mtec-institute.co.jp

The first such result is reported in the pioneering work by Wagner and Whitin (1958), where an efficient dynamic programming algorithm is proposed.

Among a number of extensions of this model, the best known is the multi-echelon problem studied by Zangwill (1968, 1969). He developed a dynamic programming algorithm by exploiting a nice structure of the extreme flow of the associated single source rectangular network flow problem. The problem with m production stages over n time periods can be solved in $O(n^4m)$ steps using dynamic programming recursion. Note that Zangwill's algorithm is virtually the only polynomial time algorithm for concave minimization problems and that it is considered as the most efficient algorithm for solving this problem.

This algorithm has been extended by a number of authors. For example, one of the authors studied the problem with backlog of demands and delay in production process (Konno 1973, 1988). It is shown that these problems can also be solved in $O(n^4m)$ steps. Further, Zangwill (1969) developed an $O(n^3m)$ algorithm for the problem with monotone cost structure . Also, Love studied another special class with nested demands (Love 1973). Readers are referred to Bitran et al. (1984) for further references in this area.

It is well known that concave cost network flow problems and even a general nonlinear cost network flow problems can be solved by piecewise linear approximation strategy by introducing a number of 0–1 integer variables (Dantzig 1959; Padberg 2000; Sherali 2001; Wolsey 1998). However, this method has been considered impractical since it was very difficult to solve the resulting mixed 0–1 integer programming problem until recently. However, due to the remarkable progress in integer programming methodologies, we can now solve a fairly large scale concave cost network flow problems using powerful state-of-the-art software. For example, we showed in Konno and Egawa (2006) that a large scale concave cost network flow problem over a bipartite network can be solved to optimality remerkably fast. In fact, it is an order of magnitude faster than the state-of-the-art branch and bound algorithm based upon hyper-rectangular subdivision strategy (Phong et al. 1995). This success motivated us to compare the dynamic programming (DP) algorithm and 0–1 integer programming (IP) approach for concave cost production/inventory control problem.

We will show that IP approach is faster for problem with set-up cost type function or piecewise concave functions with fewer linear pieces, where we need to introduce relatively few 0–1 integer variables. IP approach becomes more efficient for problems with larger n. Also, we will show that we can solve mixed cost problem which cannot be solved by DP algorithm. These problems include among others problems with convex backlog cost.

In the next two sections, we will briefly explain the multi-echelon production/inventory control problem and present dynamic programming algorithm for concave cost problems. Section 4 will be devoted to the description of integer programming algorithm. In Sect. 5, we present the results of computational experiments using DP algorithm and IP algorithm on a variety of test problems. It will be shown that IP algorithm outperforms DP algorithm for problems with piecewise linear concave cost functions with fewer linear pieces, while DP performs better for problems with general concave cost functions.

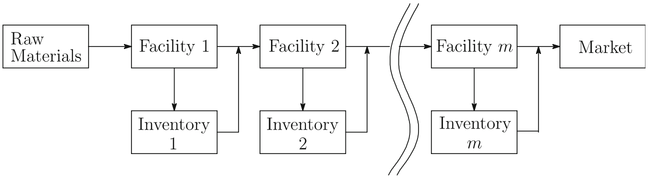


Fig. 1 Series production system

2 Network flow formulation of multi-echelon model

Let us consider a series production/inventory system depicted in Fig. 1.
Let us assume that there are m facilities where products processed at facility k is sent to facility $k+1$ for further processing or held as inventory for processing in the later period. Final products will be released to the market to fill the demand of amount r_i at period i. Let us consider the minimal cost production/inventory control problem over n periods.

Let x_i^k, y_i^k be, respectively, the amount of processing and inventory at facility k in period i and let $c_i^k(x_i^k)$ and $h_i^k(y_i^k)$ be the corresponding processing and holding cost. Also, let w_i be the amount of backlogged demand at period i and $p_i(w_i)$ be the cost associated with w_i. We will assume that the item processed at period i at facility k can be sent to facility $k+1$ only at the beginning of period i.

Therefore, the minimal cost production/inventory control problem can be formulated as follows.

$$
\left|\begin{array}{l}
\text{minimize } z = \sum_{i=1}^{n}\left[\sum_{k=1}^{m}\{c_i^k(x_i^k) + h_i^k(y_i^k)\} + p_i(w_i)\right] \\
\text{subject to } y_i^0 = y_{i-1}^0 - x_i^1,\ 1 \le i \le n-1 \\
\qquad y_i^k = x_i^k + y_{i-1}^k - x_i^{k+1},\ 1 \le i \le, n,\ 1 \le k \le m-1 \\
\qquad y_i^m = x_i^m + y_{i-1}^m + w_i - w_{i-1} - r_i,\ 1 \le i \le n \\
\qquad x_i^{k+1} \le y_{i-1}^k,\ 2 \le i \le n,\ 1 \le k \le m-1 \\
\qquad x_i^k \ge 0,\ y_i^k \ge 0,\ 1 \le i \le n,\ 1 \le k \le m \\
\qquad y_0^k = 0,\ 1 \le k \le m \\
\qquad y_0^0 = \sum_{i=1}^{n} r_i \\
\qquad w_i \ge 0,\ 1 \le i \le n-1 \\
\qquad w_n = 0.
\end{array}\right. \tag{1}
$$

This problem can be interpreted as a minimal cost network flow problem on a single source network depicted in Fig. 2 (see Konno 1988 for details).

3 Concave cost problems and dynamic programming algorithm

Let us assume that $c_i^k(\cdot), h_i^k(\cdot)$ and $p_i(\cdot)$ are nondecreasing concave function for all i and k. Then there exists an optimal solution among extreme flows, i.e., flows

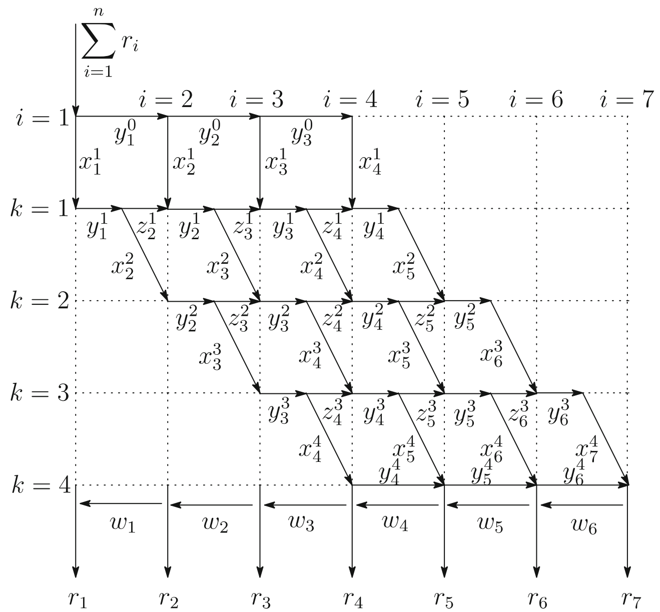


Fig. 2 Network representation of the problem $(n,m) = (7,4)$

without cycles. As a result, optimal flow f on certain arc can be represented as the sum of demands r_i for certain number consecutive periods (since no two flows can merge). Based on these observation the following DP algorithm can generate an optimal solution.

Let $f_i^k(\alpha,\beta)$ be the minimal cost for sending the flow of amount $R_\alpha^\beta \equiv \sum_{i=\alpha}^{\beta} r_i$ to node $(m,\alpha),(m,\alpha+1),\ldots,(m,\beta)$ from node (k,l).

Algorithm DP

(i) $k=m$:

$$f_i^m(\alpha,\beta) = \sum_{l=\alpha}^{i-1} p_l(R_\alpha^l) + \sum_{l=i+1}^{\beta} h_l^m(R_l^\beta),$$

$$1 \le \alpha \le i \le \beta \le n,\ m \le i \le n. \tag{2}$$

(ii) $1 \le k \le m-1$:

$$f_{i'}^k(\alpha,\beta) = \min_{\alpha-1 \le \gamma \le \beta} \{c_{i+1}^{k+1}(R_\alpha^\gamma) + f_{i+1}^{k+1}(\alpha,\gamma) + f_{i+1}^k(\gamma+1,\beta)\},$$

$$k \le i < n-m+k \tag{3}$$

$$f_{(n-m+k)'}^k(\alpha,n) = c_{n-m+k+1}^{k+1}(R_\alpha^n) + f_{n-m+k+1}^{k+1}(\alpha,n) \tag{4}$$

$$f_i^k(\alpha,\beta) = h_i^k(R_\alpha^\beta) + f_{i'}^k(\alpha,\beta),\ k \le i < n-m+k \tag{5}$$

$$f_{n-m+k}^k(\alpha,n) = h_{n-m+k}^k(R_\alpha^n) + f_{(n-m+k)'}^k(\alpha,n),\ 1 \le \alpha \le \beta \le n. \tag{6}$$

(iii) $k = 0$:

$$f_i^0(\alpha,\beta) = \min_{\alpha-1\le\gamma\le\beta}\{c_i^1(R_\alpha^1) + f_i^1(\alpha,\gamma) + f_i^0(\gamma+1,\beta)\},$$
$$1 \le i \le n-m+k-1,\ 1 \le \alpha \le \beta \le n \tag{7}$$

$$f_{n-m+k}^0(n,n) = f_{n-m+k}^1(n,n). \tag{8}$$

It is easy to see that $O(n^4m)$ arithmetic operations are required to obtain the minimum value of objective function of (1), i.e., $f_1^0(1,n)$.

4 Integer programming approach for nonlinear network flow problem

When the cost function is piecewise linear, the problem can be reformulated as a mixed 0–1 integer programming problems by introducing a number of 0–1 variables (Dantzig 1959; Padberg 2000; Sherali 2001). For example, if $c(\cdot)$ is piecewise linear with three linear pieces (Fig. 3), then it can be represented as a linear function as follows:

$$c(\xi) = c_1\lambda_1 + c_2\lambda_2 + c_3\lambda_3 \tag{9}$$
$$\xi = \xi_1\lambda_1 + \xi_2\lambda_2 + \xi_3\lambda_3 \tag{10}$$
$$\lambda_0 + \lambda_1 + \lambda_2 + \lambda_3 = 1 \tag{11}$$
$$\phi_1 + \phi_2 + \phi_3 = 1 \tag{12}$$
$$\lambda_0 \le \phi_1 \tag{13}$$
$$\lambda_1 \le \phi_1 + \phi_2 \tag{14}$$
$$\lambda_2 \le \phi_2 + \phi_3 \tag{15}$$
$$\lambda_3 \le \phi_3 \tag{16}$$
$$\lambda_k \ge 0, \quad k = 0,1,2,3 \tag{17}$$
$$\phi_k = 0 \text{ or } 1, \quad k = 1,2,3 \tag{18}$$

Also, a general nonlinear function can be approximated by a piecewise linear function within a required precision. It has been demonstrated in Konno et al. (2006) that following successive piecewise linear approximation scheme works well for concave cost bipartite network flow problems.

This scheme is due to Dantzig (1959). Alternative schemes are proposed by Padberg (2000) and Sherali (2001), which are claimed to be superior to Dantzig's scheme above. However, a number of experiments on portfolio optimization problem under concave transaction costs (Konno et al. 2006) show that Dantzig's scheme performs best.

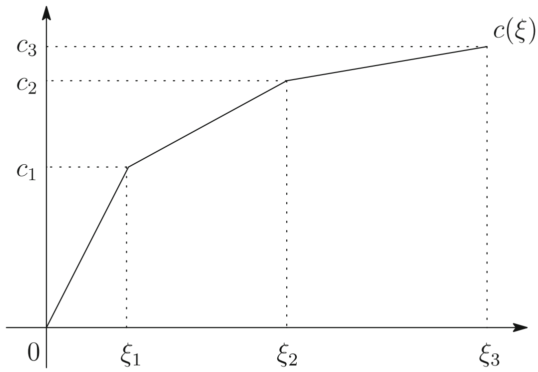


Fig. 3 Piecewise linear cost function

Successive piecewise linear approximation strategy

Let $\tilde{c}_i^k(x_i^k), \tilde{h}_i^k(y_i^k), \tilde{p}_i(w_i)$ be a piecewise linear approximation of $c_i^k(x_i^k), h_i^k(y_i^k)$, $p_i(w_i)$, respectively using subdivision of intervals under consideration. We solve the resulting linear programming problem containing 0–1 variables and let $\tilde{x}_i^k$, $\tilde{y}_i^k$ and $\tilde{w}_i$ be its optimal solutions. If the sum of discrepancies

$$\sum_{i=1}^{n}\left[\sum_{k=1}^{m}\left\{\left(c_i^k(\tilde{x}_i^k)-\tilde{c}_i^k(\tilde{x}_i^k)\right)+\left(h_i^k(\tilde{y}_i^k)-\tilde{h}_i^k(\tilde{y}_i^k)\right)\right\}+\left(p_i(\tilde{w}_i)-\tilde{p}_i(\tilde{w}_i)\right)\right] \Big/ \sum_{i=1}^{n}\left[\sum_{k=1}^{m}\left\{c_i^k(\tilde{x}_i^k)+h_i^k(\tilde{y}_i^k)\right\}+p_i(\tilde{w}_i)\right] \quad (19)$$

is less than $\varepsilon(>0)$, then the calculated solution is an ε-optimal solution. If not, we generate a finer mesh around the current optimal solution and solve the problem and continue this process until the required precision is attained.

There is no guarantee that this process always generates an ε-optimal solution, but our numerical experiments show that the solution obtained by this method never fails to converge to an optimal solution calculated by DP algorithm.

Among several possible subdivision strategies, the following strategy performs best.

Subdivision strategy

Let $f_1(x)$ be a linear underestimator of a concave function $f(x)$ over $[0, a]$ and let

$$\xi_1 = \underset{0\le x\le a}{\operatorname{argmin}}\{f(x) - f_1(x)\}. \quad (20)$$

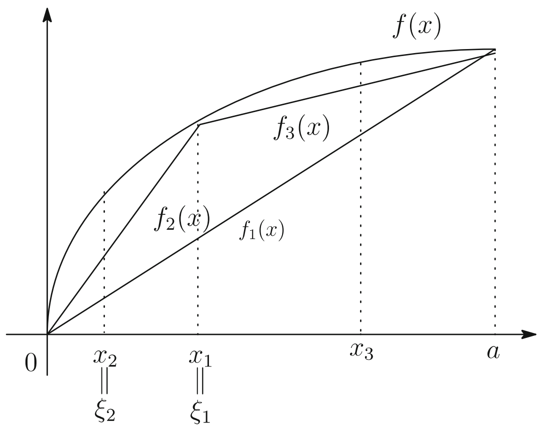


Fig. 4 Subdivision Strategy for a concave function

We subdivide $[0, a]$ into subintervals $[0, \xi_1]$, $[\xi_1, a]$. Let $f_2(x)$, $f_3(x)$ be linear underestimator of $f(x)$ over $[0, \xi_1]$, $[\xi_1, a]$, respectively.

We choose either one of the points

$$x_2 = \underset{0 \le x \le \xi_1}{\text{argmin}} \{f(x) - f_2(x)\} \tag{21}$$

$$x_3 = \underset{\xi_1 \le x \le a}{\text{argmin}} \{f(x) - f_3(x)\}, \tag{22}$$

whichever the associated discrepancy is larger as the next subdivision point and continue (see Fig. 4).

5 Computational results

We conducted numerical experiments using DP algorithm and IP algorithm using PentiumIV (512 Mbyte, 2.8 GHz) personal computer. We used CPLEX8.0 for solving IP problems.

(1) Production cost: $c_i^k(x)$

We considered three different types of functions (a)~(c).

(a) Set-up type cost (Fig. 5a)

$$c_i^k(x) = \begin{cases} a_i^k x + b_i^k & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

(b) Piecewise linear concave function with three linear pieces (Fig. 5b)

$$c_i^k(x) = \begin{cases} a_i^{1k} x + b_i^{1k} & \text{if } 0 \le x \le x_1 \\ a_i^{2k} x + b_i^{2k} & \text{if } x_1 \le x \le x_2 \\ a_i^{3k} x + b_i^{3k} & \text{otherwise} \end{cases} \tag{24}$$

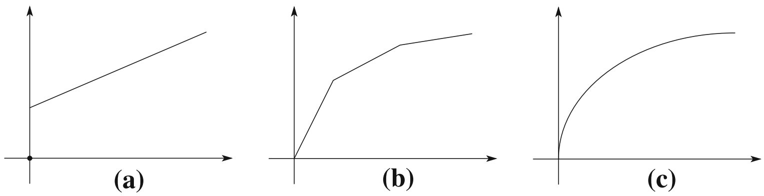


Fig. 5 Production cost function

Table 1 Data types

Type	Production cost	Holding cost	Backlog cost
I	Set-up	Linear	Linear
II	Piecewise linear	Linear	Linear
III	Concave	Linear	Linear
IV	Concave	Linear	Convex

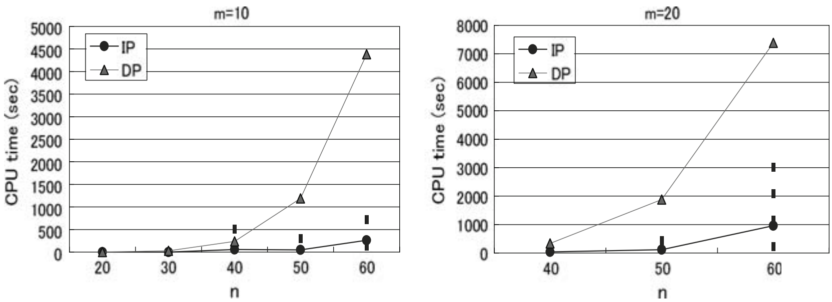


Fig. 6 Average CPU time of Type I problems (*dotted line* shows maximum and minimum of computation time)

(c) General concave function (Fig. 5c)

$$c_i^k(x) = x^{a_{ik}}, \quad (0 < a_{ik} < 1.0) \tag{25}$$

We conducted computational tests for three different combinations of cost functions listed in Table 1.

We generated ten test problems for each combination of (n, m) using randomly generated demand sequence r_i's. Number of 0–1 integer variables of integer programming formulation for Type I, II problems are $m(n-m-1)$ and $3m(n-m-1)$, respectively. For Type III problem, we apply successive piecewise linear approximation scheme of Sect. 4 to calculate an approximately optimal solution. At each iteration of this procedure we need to introduce $5m(n-m-1)$ zero-one integer variables.

We see from Fig. 6 that IP algorithm is more than ten times faster than DP algorithm on the average. On the other hand, computation time of IP algorithm is dependent on data, but it is almost always faster than DP algorithm.

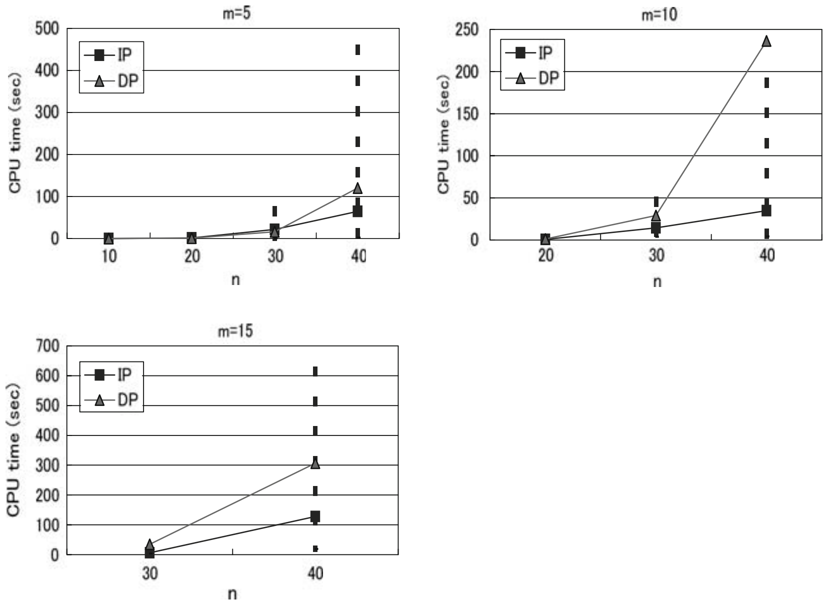


Fig. 7 Average CPU time of Type II problems (*dotted line* shows maximum and minimum of computation time)

For Type II problems, DP is somewhat faster than IP for smaller problems. However, IP is more than five times faster when $(n, m) = (40, 15)$ in spite of the fact that we have to handle problem with over one thousand 0–1 variables (Fig. 7).

To solve Type III problems by IP, we introduced five subdivision point for each concave arc and applied successive approximation scheme until using $\varepsilon = 10^{-4}$.

We see from Fig. 8 that DP algorithm outperforms IP for smaller (n, m)'s as expected. However, IP algorithm becomes more efficient when (n, m) is larger.

Finally, we solved Type IV problem using DP and IP algorithms. DP algorithm need not generate an optimal solution of the problem (1). For IP algorithm we used successive piecewise linear approximation of convex function and solved the resulting 0–1 integer programming problem. Note that we do not have to introduce 0–1 integer variables for approximating convex functions. Computation time for resulting 0-1 integer programming problems is more or less the same as Type II problems. Also, we see from Fig. 9 and 10 that DP algorithm fails to generate an optimal solution.

6 Conclusions

We showed in this paper that integer programming algorithm can solve a large scale concave cost production/inventory control problems much faster than dynamic programming algorithm when n, the number of planning periods is

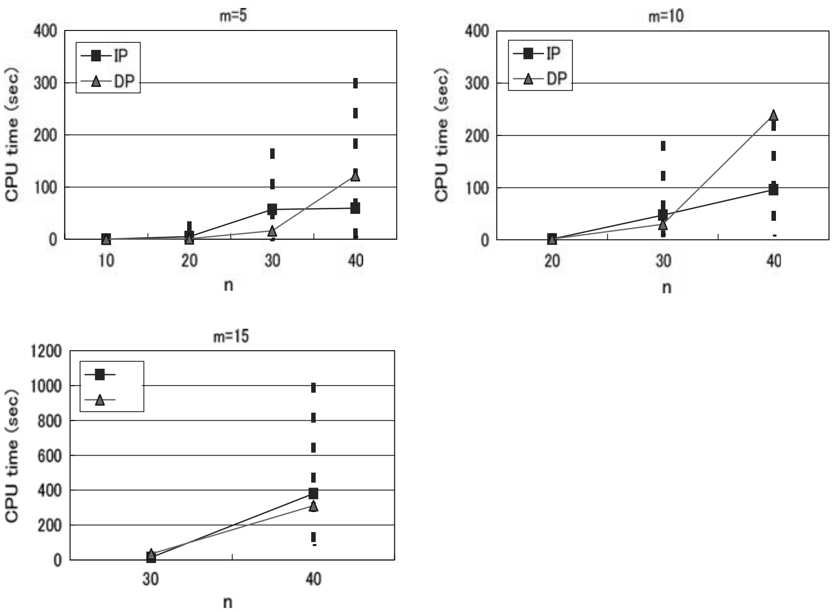
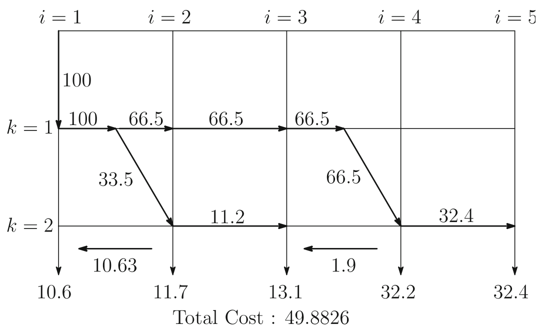


Fig. 8 Average CPU time of Type III problems (*dotted line* shows maximum and minimum of computation time)

Fig. 9 Solution obtained by IP

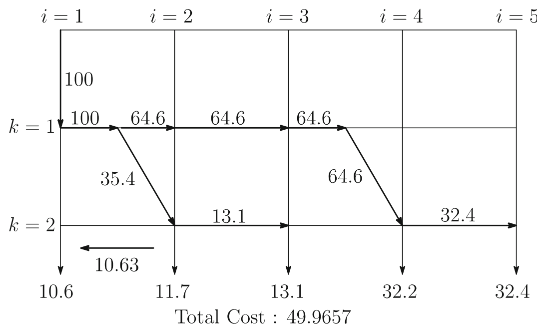


large and cost function is either set up type function or piecewise linear function with fewer linear pieces.

If, on the other hand the majority of cost functions are general concave functions, then DP algorithm would outperform IP algorithm since the computation time of DP algorithm depends on the size of the problem, while IP algorithm depends on the number of 0–1 variables.

We showed in additions that IP approach can be successfully applied to general piecewise linear cost functions, not necessarily concave, while DP algorithm fails to generate an optimal solution.

Fig. 10 Solution obtained by DP



Acknowledgments The research of the first author was supported in part by the Grant-in-Aid for Scientific Research B18310109 of the Ministry of Education, Science, Sports and Culture of the Government of Japan.

References

Bitran GR, Magnanti TL, Yanasse HH (1984) Approximation methods for the uncapacitated dynamic lot size problem. Manage Sci 30:1121–1140

Dantzig GB (1959) Note on solving linear programs in integers. Naval Res Logist Q 6:75–76

Konno H (1973) Minimum concave series production system with deterministic demands—a backlogging case. J Oper Res Soc Jpn 16:246–253

Konno H (1988) Minimum concave cost production systems : A further generalization of multi-echelon model. Math Program 4:185–193

Konno H, Egawa T (2006) Computational studies on large scale concave cost transportation problems. Pac J Optim 2:327–340

Konno H, Izumi K, Yamamoto R (2006) Comparison of Branch and Bound Algorithms and 0–1 Integer Programming Algorithm for Concave Minimization Under Linear Constraints. ISE06-01, Department of Industrial and System Engineering, Chuo University

Love SF (1973) A facilities in series inventory model with nested schedules. Manage Sci 18:327–338

Padberg MW (2000) Approximating separable nonlinear functions via mixed zero-one programs. Oper Res Lett 27:1–5

Phong TQ, An LTH, Tao PD (1995) Decomposition branch and bound method for globally solving linearly constrained indefinite quadratic minimization problems. Oper Res Lett 17:215–220

Sherali HD (2001) On mixed-integer 0–1 representations for separable lower-semicontinuous piece-wise linear functions. Oper Res Lett 28:155–160

Wagner MH, Whitin TM (1958) Dynamic version of economic lot-sizing models. Manage Sci 5:89–96

Wolsey LA (1998) Integer programming. Wiley, New York

Zangwill WI (1968) Minimum concave cost flows in certain-networks. Manage Sci 14:429–450

Zangwill WI (1969) A backlogging model and a multi-echelon model of a dynamic economic lot size production system—a network approach. Manage Sci 15:506–527