Yokohama Publishers
ISSN 1880-5221 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 2, No. 2, pp. 327–339, May 2006

COMPUTATIONAL STUDIES ON LARGE SCALE CONCAVE COST TRANSPORTATION PROBLEMS

Hiroshi Konno and Takaaki Egawa

Abstract: This paper is concerned with computational studies on classical and very difficult concave cost transportation problems. We apply successive piecewise linear approximation scheme by introducing zero-one integer variables. We will show that an approximately optimal solution is obtained within a practical amount of time for problems up to 100 concave source nodes using various types of data. This is a significant improvement over the past computational studies on the same problem, where the largest problem solved is up to several concave nodes.

Key words: *concave minimization, transportation problem, integer programming, piecewise linear approximation*

Mathematics Subject Classification: *65K05, 90B06, 90C06, 90C10, 90C26, 90C27, 90C59*

1 Introduction

This paper is concerned with computational studies on concave cost network flow problems. Though very important from the practical point of view, only small scale problems have been solved to optimality in the past since this problem belongs to the NP hard family [11]. The only exception is a class of multi-echelon production and inventory problem, for which polynomial time dynamic programming algorithms have been developed in the past [7, 16].

The problem under consideration is a class of minimal cost production transportation problem on a bipartite network in which concave production costs are associated with source (production) nodes. This problem has many practical applications including classical production transportation problems. Also, as shown in a recent paper by Tuy et al. [14], a general uncapacitated network flow problem with k concave cost arcs can be converted to a transportation problem with k concave production nodes.

Many authors have proposed exact algorithms for solving this problem. For example, Soland [13] applied a branch and bound method using a linear underestimator of a concave function. Also, Erickson et al. [4] applied a send-and-split method. More recently, Tuy et al. [13] proposed an algorithm based upon total enumeration of vertices by projecting the problem into a smaller dimensional space. Also one of the authors proposed an outer-approximation algorithm to an equivalent problem in a smaller dimensional space[15].

Unfortunately however, these exact algorithms can solve only small scale problems. In particular, the largest problem solved in [15] is a problem with six concave production nodes.

The purpose of this paper is to demonstrate that we can now solve an order of magnitude larger problem, say up to 100 concave production nodes by a successive piecewise linear approximation procedure using 0-1 integer variables [3]. The success depends upon the recent remarkable innovation in the field of integer programming methodologies and software [2].

One may argue that this approach may fail to generate a true optimal solution for two reasons. One is the reliability of the commercial software. Is there a guarantee that the calculated solution is in fact an optimal solution? The other is whether a successive piecewise linear approximation scheme will guarantee optimality within some tolerance.

These questions have been resolved, though not completely by a series of our numerical experiments on concave cost portfolio optimization problems [9], where we compared an (exact) branch and bound algorithm of Phong et al. [12] and successive piecewise linear approximation approach to these class of problems and confirmed that two methods generate the same solution within some tolerance. This in turn guarantees the acclaimed quality of the commercial software.

In this paper, we will present results of extensive computational studies on two classes of problems. The first class is a standard production-transportation problem on a bipartite network on two dimensional space where production cost is continuous and concave and transportation cost is linear.

The second is a class assignment problem. Associated with an assignment of student j to class i is a score c_{ij}. If the capacity of each class is fixed, then the problem of maximizing the total score becomes a simple transportation problem. However, in a practical class assignment, we need to modify the problem by varying the class capacity and imposing additional cost associated with capacity expansion. The problem can be formulated as a minimization of piecewise linear concave function under transportation type constraints. We applied a branch and bound algorithm but was unable to solve problems of practical size [15].

In the next two sections, we will explain the problem and its mixed integer programming formulations in detail. Section 4 will be devoted to the results of numerical experiments on large scale prblems. These results may be of interest to researchers in the field of concave cost network flow problems as well as general concave minimization problems. Also, it should be of interest to those who are not very familiar with the state-of-the-art of integer linear programming methodologies since there are still relatively few open literature reporting the computational results on large scale integer programming problems.

2 Concave Production Cost Transportation Problem

Let us consider a standard transportation problem with m supply nodes and n demand nodes. Associated with ith supply node is production capacity represented by a_i, $i = 1, 2, \dots, m$. Also jth demand node requires b_j units, $j = 1, 2, \dots, n$.

Let w_i be the amount of production at node i and let $g_i(w_i)$ be the accompanied cost. Also let x_{ij} be the unit of commodities to be transported from node i to j. Also, let c_{ij} be the unit transportation cost. Then the problem can be formulated as follows:

$$\left|\begin{array}{lrl} \text{minimize} & f(\boldsymbol{x}, \boldsymbol{w}) = & \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{m} g_i(w_i) \\ \text{subject to} & w_i \equiv & \displaystyle\sum_{j=1}^{n} x_{ij} \le a_i, \quad i = 1, 2, \dots, m \\ & & \displaystyle\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1, 2, \dots, n \\ & & x_{ij} \ge 0, \quad i = 1, 2, \dots, m;\ j = 1, 2, \dots, n. \end{array}\right. \tag{1}$$

As in [15] we will assume that the problem has a feasible solution, namely that $\sum a_i \geq \sum b_j$ and the production cost of the first s nodes are concave and the rest are linear. We will split the interval $[0,\ a_i]$, $i = 1, 2, \ldots, s$ into k subintervals of equal length and let $0 \equiv z_{i0} < z_{i1} < z_{i2} < \cdots < z_{ik} \equiv a_i$ be its end points.

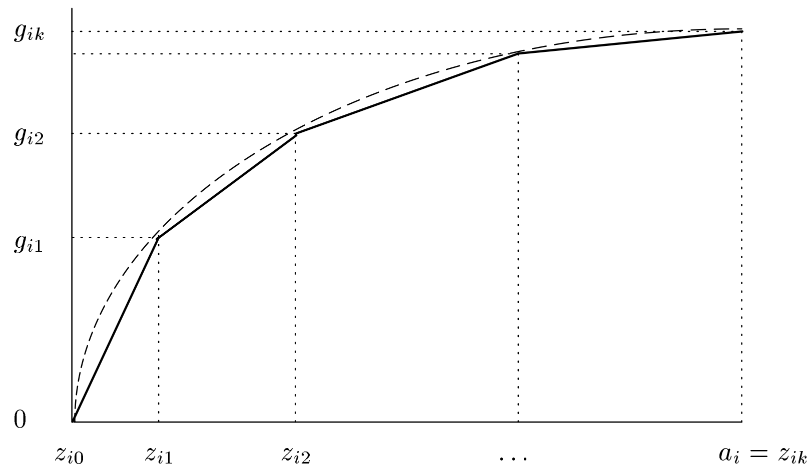

Figure 1: Piecewise Linear Approximation

Let $g_{il} = g_i(z_{il})$. Then we can represent the piecewise linear approximation of $g_i(w_i)$ as follows [3]:

$$g_i(w_i) = \sum_{l=1}^{k} \lambda_{il} g_{il}, \tag{2}$$

where

$$\begin{aligned}
&\sum_{l=0}^{k} \lambda_{il} = 1 \\
&\lambda_{il} \geq 0, \quad l = 0, 1, \ldots, k \\
&\lambda_{i0} \leq y_{i0} \\
&\lambda_{il} \leq y_{i\,l-1} + y_{il}, \quad l = 1, 2, \ldots, k \\
&\sum_{l=0}^{k} y_{il} = 1 \\
&y_{il} \in \{0,\ 1\}, \quad l = 0, 1, \ldots, k.
\end{aligned} \tag{3}$$

Therfore the minimization of the piecewise linear concave function is represented as follows:

$$
\left|\begin{array}{ll}
\text{minimize} & F(\boldsymbol{x},\,\boldsymbol{\lambda}) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{i=1}^{s}\sum_{l=1}^{k} g_{il}\lambda_{il} + \sum_{i=s+1}^{m} g_i \sum_{j=1}^{n} x_{ij} \\
\text{subject to} & \sum_{j=1}^{n} x_{ij} \le a_i, \quad i = 1,2,\dots,m \\
 & \sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1,2,\dots,n \\
 & x_{ij} \ge 0, \quad i = 1,2,\dots,m;\ j = 1,2,\dots,n \\
 & \sum_{j=1}^{n} x_{ij} = \sum_{l=1}^{k} z_{il}\lambda_{il}, \quad i = 1,2,\dots,s \\
 & \sum_{l=0}^{k} \lambda_{il} = 1, \quad i = 1,2,\dots,s \\
 & \sum_{l=0}^{k} y_{il} = 1, \quad i = 1,2,\dots,s \\
 & \lambda_{i0} \le y_{i0}, \quad i = 1,2,\dots,s \\
 & \lambda_{il} \le y_{il-1} + y_{il}, \quad i = 1,2,\dots,s;\ l = 1,2,\dots,k \\
 & \lambda_{il} \ge 0, \quad i = 1,2,\dots,s;\ l = 0,1,\dots,k \\
 & y_{il} \in \{0,\,1\}, \quad i = 1,2,\dots,s;\ l = 0,1,\dots,k.
\end{array}\right. \tag{4}
$$

This problem is a zero-one integer linear programming problem with $(k+1)\times s$ zero-one variables.

Remark. This formulation is valid for general $g(\cdot)$, not necessarily concave.

When $k = 10$ and $s = 50$, the problem contains 550 zero-one variables. The problem of the size can be solved by the state-of-the-art integer programming software such as CPLEX of ILOG and XPRESS of Dash Optimization. Let x^*_{ij}, $i = 1,2,\dots,m$, $j = 1,2,\dots,n$; λ^*_{il}, y^*_{il}, $i = 1,2,\dots,s$, $l = 0,1,\dots,k$ be an optimal solution. Let

$$
w^*_i = \sum_{j=1}^{n} x^*_{ij}, \quad i = 1,2,\dots,s.
$$

If

$$
\sum_{i=1}^{s}\{g_i(w^*_i) - \sum_{l=1}^{k} g_{il}\lambda^*_{il}\} / \sum_{i=1}^{s} g_i(w^*_i) \le \varepsilon, \tag{5}
$$

for small enough ε, then x^*_{ij}, $i = 1,2,\dots,m$; $j = 1,2,\dots,n$ is an approximate optimal solution.

If the condition (5) is not satisfied, we will introduce a finer subdivision around the current solution. In particular, the following subdivision strategy generates a truely optimal solution without fail as demonstrated in our earlier studies on portfolio optimization problems under concave transaction costs [9, 10].

Subdivision.

Let w_i^* be an optimal solution of (4) and let $w_i^* \in [z_{il}, z_{il+1}]$. Then we choose $[z_{il-1}, z_{il+2}]$ as a new inteval for w_i, $i = 1, 2, \ldots, s$ and subdivide this interval into k subintervals of equal length.

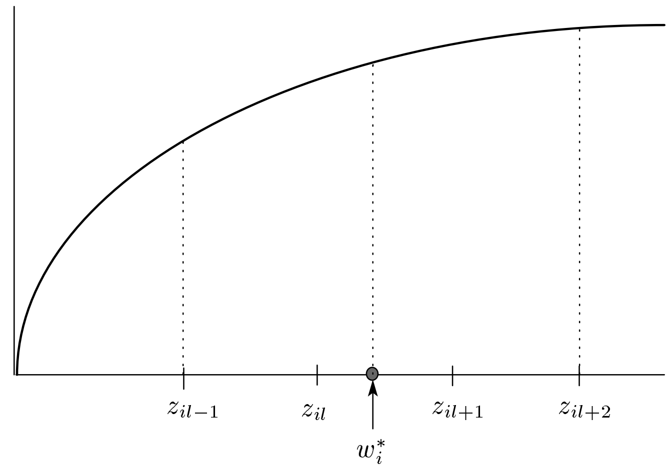

Figure 2: Finer Subdivision

3 Class Assignment Problem with Variable Class Capacity

Let there be m classes and n students. Each student j ($j = 1, 2, \ldots, n$) is supposed to belong to exactly one class i ($i = 1, 2, \ldots, m$). Associated with class i is a fixed capacity a_i. Let c_{ij} be the degree of satisfaction associated with assigning student j to class i. Then the problem can be formulated as a standard transportation problem.

$$\left| \begin{array}{ll} \text{maximize} & \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \\ \text{subject to} & \displaystyle\sum_{j=1}^{n} x_{ij} \le a_i, \quad i = 1, 2, \ldots, m \\ & \displaystyle\sum_{i=1}^{m} x_{ij} = 1, \quad j = 1, 2, \ldots, n \\ & x_{ij} \in \{0, 1\}, \quad i = 1, 2, \ldots, m; \ j = 1, 2, \ldots, n. \end{array} \right. \tag{6}$$

This problem has an optimal solution $\boldsymbol{x}^* = (x_{11}^*, x_{12}^*, \ldots, x_{21}^*, x_{22}^*, \ldots, x_{mn}^*)$ if $\displaystyle\sum_{i=1}^{m} a_i \ge n$. However, in practice more than 70% of c_{ij}'s have large negative value, so that the total score may be unacceptably low. Then we need to modify the class capacity. Let v_i be the additional capacity of class i, where $v_i \le v_{i0}$ for some constant v_{i0}. Associated with an increase v_i is a piecewise linear concave cost $d_i(v_i)$. The problem then becomes:

$$\left|\begin{array}{ll}\text{maximize} & \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} - \sum_{i=1}^{m} d_i(v_i) \\ \text{subject to} & \displaystyle\sum_{j=1}^{n} x_{ij} \leq a_i + v_i, \quad i = 1, 2, \ldots, m \\ & \displaystyle\sum_{i=1}^{m} x_{ij} = 1, \quad j = 1, 2, \ldots, n \\ & 0 \leq v_i \leq v_{i0}, \quad i = 1, 2, \ldots, m \\ & v_i \in Z_+, \quad i = 1, 2, \ldots, m \\ & x_{ij} \in \{0, 1\}, \quad i = 1, 2, \ldots, m; j = 1, 2, \ldots, n, \end{array}\right. \tag{7}$$

where Z_+ is the set of non-negative integers.

Theorem3.1.
A matrix $A = (a_{ij})$ is totally unimodulor if

(a) $a_{ij} \in \{+1, -1, 0\}$ for all i, j.
(b) Each column contains at most two nonzero coefficients.
(c) There exists a partition (M_1, M_2) of the set M of rows such that each column j containing two nonzero coefficients satisfies $\sum_{i \in M_1} a_{ij} - \sum_{i \in M_2} a_{ij} = 0$.

Proof. See Theorem 8.9 of [13] or Proposition 3.2 of [9]. □

Corollary 3.2.
Every basic feasible solution of the linear system of (7) (without integer constraints) is integral.

Proof. It is straightforward to see that the constraint matrix satisfies the condition of Theorem 3.1. □

This means that a class assignment problem (7) can be solved by the standard linear programming algorithm. Also, the function $d_i(\cdot)$ is piecewise linear, so that piecewise linear approximation is free from approximation error.

Unfortunately, however we usually need to impose additional constraint like:

$$\sum_{i \in R_l} v_i \leq v_0, \quad l = 1, 2, \ldots, L,$$

where $R_l \subset \{1, 2, \ldots, m\}$. Then the problem is no longer totally unimodular.

This problem has been discussed in detail in [16], but was not solvable then.

4 Computational Experiments

We conducted extensive computational experiments by choosing several alternative data sets since the efficiency of the integer programming algorithm depends not only upon the size and the structure of the problem, but also on data.

All computation were conducted on Pentium(R) 4 CPU 2.80GHz, Memory 1.00GB and we used CPLEX9.0 for solving 0-1 integer programming subproblems.

4.1 Computational Experiments on Problem(4)

We generate m factories and n warehouses randomly on a unit square and calculate the transportation cost c_{ij} proportional to physical distance. Also, we employed $g_i(w_i) = \alpha w_i^{0.6}$ for concave production function, where α is adjusted in such a way that the transportation cost and production cost is almost equal since other cases are much easier to solve. All computation were conducted using successive piecewise linear approximation under the condition $k = 10$ and $\varepsilon = 10^{-5}$.

Figure 3 shows the computation time for solving integer programming problem (4) where $s = 5$, $m = 100$ using CPLEX9.0. We also plotted the computation time reported in [16], which was conducted on NEWS 5000 workstation using outer-approximation algorithm.

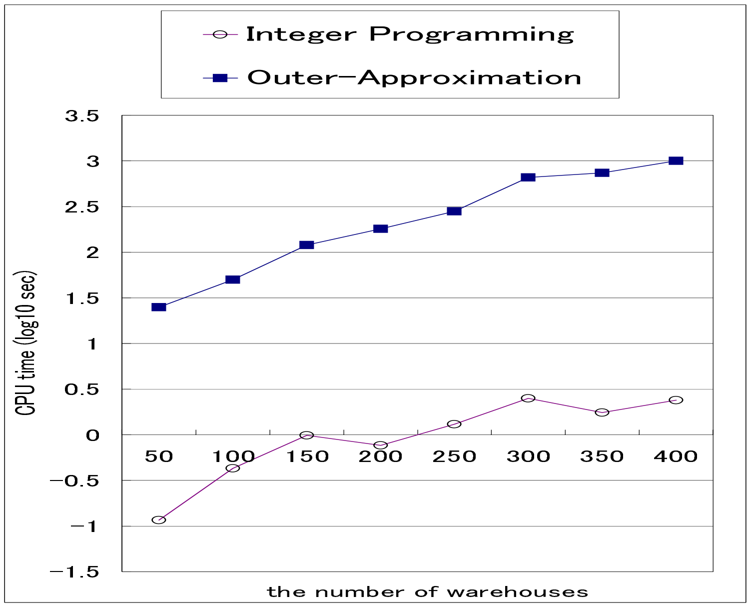

Figure 3: Comparison of Integer Programming and Outer-Approximation

Table 1: Integer Programming(CPU sec.)

data set \ n	100	200	300	400
1	0.64	1.12	1.7	2.58
2	0.71	0.32	1.91	1.88
3	0.09	0.62	5.79	3.73
4	0.45	1.17	2.08	2.04
5	0.25	0.59	1.03	1.76
Median	0.45	0.62	1.91	2.04
Average	0.43	0.76	2.50	2.40
St.Dev.	0.26	0.37	1.88	0.81

We see that the CPU time of (IP) remains more or less constant while that of (OA) sharply increases as s increases. Note that $s = 6$ was the maximal size of the problem solvable by (OA) algolithm [16]. On the other hand, as shown below we can solve problems with s over 100. Table 1 shows the detailed statistics of (IP) where the capacity $a_i^{'}$s and $b_j^{'}$s are chosen in such a way that $\sum a_i \simeq 1.3 \sum b_j$.

Table 2 shows computational results for the problem where the production capacity of each node is distributed uniformly over the interval [700, 1000] and $\sum a_i \simeq 1.2 \sum b_j$. Production cost functions $g_i(\cdot)$ are same as before and $s = m$, i.e., all production functions are nonlinear and concave.

Table 2: Computational Results for $n = 100$. (CPU sec.)

data set \ s	40	60	80	100
1	2.45	15.29	604.1	534.37
2	8.26	20.37	18.97	2049.6
3	2.09	24.94	33.85	487.55
4	3.21	13.25	22.18	2448.9
5	4.37	3.45	679.03	503.38
6	4.25	2.2	49.1	84.79
7	8.62	8.97	98.45	42.78
8	3.89	8.35	48.7	42.75
9	0.79	3.05	208.92	141.01
10	1.18	13.26	399.78	322.46
Median	3.55	11.11	73.78	405.01
Average	3.91	11.31	216.31	665.76
St.Dev.	2.68	7.59	253.44	861.33

Table 3: Computational Results for $n = 100$. (CPU sec.)

data set \ s	40	60	80	100
1	0.75	1.66	2.33	14.24
2	1.07	0.89	1.44	10.29
3	0.77	0.64	3.29	17.03
4	0.68	1	3.3	5.51
5	1.66	1.37	10.88	7.29
6	2.07	0.43	4.27	3.62
7	2.19	1.49	1.5	2.36
8	2.56	2.84	1.85	2.7
9	1.06	1.91	2.12	5.34
10	1.07	0.54	1.83	5.41
Median	1.07	1.19	2.23	5.46
Average	1.39	1.28	3.28	7.38
St.Dev.	0.68	0.74	2.82	4.96

We see that the problem of the size up to $(s, n) = (80, 100)$ can be solved without difficulty.

Table 3 shows the result for alternative set of data where concave production cost functions differ from node to node. Typical production function have the form $g_i(w_i) = \alpha w_i^p$ where p varies in the interval [0.40, 0.99].

We see that these problems can be solved much faster. This is due to the fact that the initial linear relaxation of the 0-1 integer program generates a good upper bound of the optimal solution.

Figure 4 shows the computation time as a function of s for two different sets of data. We see that the computation time is significantly smaller for non-uniform production function, as expected.

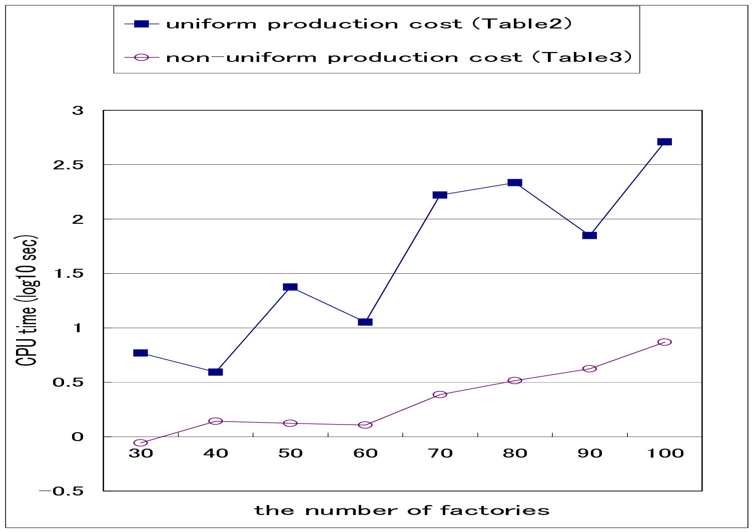

Figure 4: Computational Results for Alternative Sets of Data

Table 4 shows the statistics when we vary the number of warehouses n when the number of factories $m(=s)$ is fixed at 30. The production capacity of each node is generated randomly in the interval $[700, 1000]$ as before and $\sum a_i \simeq 1.4 \sum b_j$.

Table 4: Computational Results for $m = s = 30$. (CPU sec.)

data set \ n	100	200	300	400
1	11.97	91.97	10296	3844.1
2	1.98	1288	2953.9	2179.1
3	7.13	18.53	2247	*
4	6.95	940.88	608.25	8865.9
5	2.46	58.67	746.34	3444.3
Median	6.95	91.97	2247	3644.2
Average	6.10	479.61	3370.30	4583.40
St.Dev.	4.08	592.94	3997.40	2941.90

* We failed to obtain an optimal solution due to the shortage of memory capacity.

Table 5 shows similar results when the production cost has large variation among 30 concave production nodes, a.e., $g_i(w_i) = \alpha w_i^p$ where p varies in the interval [0.40, 0.99].

Table 5: Computational Results for $m = s = 30$. (CPU sec.)

data set \ n	100	200	300	400
1	0.95	1.45	5.73	2.4
2	0.75	1.76	1.95	3.2
3	0.66	0.74	3.27	3.42
4	0.89	0.98	1.89	2.27
5	0.61	1.47	5.13	2.79
Median	0.75	1.45	3.27	2.79
Average	0.77	1.28	3.59	2.82
St.Dev.	0.15	0.41	1.78	0.50

Figure 5 summarizes the results of Table 4 and 5. We see that the non-uniform case can be solved much faster.

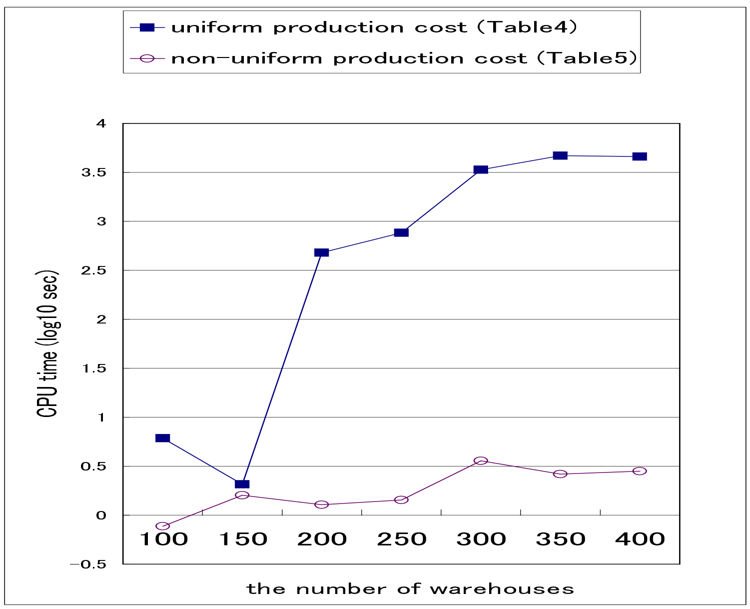

Figure 5: Computation Time for Two Alternative Sets of Data

Table 6 shows the results for larger problems with non-uniform production costs.

Let us add that linear production cost problems can be solved much faster, in less than 1% of the computation time required to solve nonlinear problems.

4.2 Computational Experiments on Problem(7)

Finally, Table 7 shows the statistics for the class assignment problem (7). Data sets 1 and 2 correspond to the real assignment data at the Department of Industrial and Systems

Table 6: Computational Results for $m = s = 100$. (CPU sec.)

data set \ n	400	600	800	1000
1	282.18	3846.59	970.43	4049.07
2	1608.93	887.77	246.15	3531.56
3	40.71	428.07	181.96	4075.03
4	24.19	420.73	241.78	249.94
5	47.48	126.67	80.57	197.47
Median	47.48	428.07	241.78	3531.56
Average	400.70	1141.97	344.18	2420.61
St.Dev.	683.74	1536.25	356.40	2017.26

Engineering at Chuo University, where $(m, n) = (13, 154)$, $(a_i, v_{i0}) = (12, 3)$ for all i and c'_{ij}s have values between 10 to 100, while some c'_{ij}s have large negative value i.e., -10^5.

In addition, we imposed several constraints on v'_is including $\sum_{i=1}^{m} v_i \leq 0.1 \sum_{i=1}^{m} a_i$. Data sets $3 \sim 5$ are artificial data genarated by mixing data sets 1 and 2. Data for large scale problems of columns $2 \sim 4$ are generated by duplicating the data of column 1. The concave cost function $d_i(v_i)$ employed in this computational is $25 v_i^{0.7}$ for all i.

We see that the real problems $(m, n) = (13, 154)$ were solved in less than a second. Note that we could not solve the similar problem by the commercial integer programming software ten years ago. Also, larger problems $(m, n) = (247, 2926)$ were solved in less than 100 seconds.

Table 7: Computational Results for (m, n). (CPU sec.)

data set \(m, n)	(13, 154)	(91, 1078)	(169, 2002)	(247, 2926)
data1	0.03	6.95	27.66	43.25
data2	0.03	5.83	26.84	58.42
data3	0.05	27.47	24.66	67.23
data4	0.07	5.82	25.04	68.67
data5	0.05	4.43	22.18	49.01
Median	0.04	6.39	25.94	62.825
Average	0.05	11.52	26.05	59.39
St.Dev	0.02	10.65	1.43	11.68

5 Conclusions

We showed in this paper that large scale concave cost transportation problems with up to 100 concave production nodes can now be solved within a practical amount of time by applying classical piecewise linear approximation of concave functions using 0-1 integer variables. The resulting 0-1 integer programming problem can be solved by the state-of-the-art integer programming software.

In addition, we showed that large scale class assignment problems with concave cost and additional constraints destroying the total unimodularity property of the constraint set can be solved fast. Note that smaller problems solvable within a second now were not solvable a decade ago.

We believe that a combination of global optimization methodology and integer programming methodology can solve other difficult class of large scale concave minimization problems.

Acknowledgements

The research of the first author was supported in part by the Grant-in-Aid for Scientific Research B(2) 15310122 and 15656025 of the Ministry of Education, Science, Sports and Culture of Government of Japan.

References

[1] A. Balakrishnan and S.C. Graves, A composite algorithm for concave-cost network flow problems, *Networks* 19 (1989) 175–202.

[2] R.E. Bixby, Solving real-world linear programs: a decade and more of progress, *Oper. Res.* 50 (2002) 3–15.

[3] G.B. Dantzig, *Linear Programming and Extensions*, Princeton University Press, 1963.

[4] R.E. Erickson, C.L. Monma and A.F. Jr. Veinott, Send-and-split method for minimum-concave-cost network flows, *Math. Oper. Res.* 12 (1987), 634–664.

[5] G. Gallo, C. Sandi and C. Sodini, An algorithm for the min concave cost flow problem, *European J. Oper. Res.* 4 (1980) 248–225.

[6] T.C. Hu, *Integer Programming and Network Flows*, Addison-Wesley Publishing Co., Ma., 1969.

[7] H. Konno, Minimum concave cost production system: a further generalization of multi-echelon model, *Math. Program.* 41 (1988) 185–193.

[8] H. Konno and Z. Zhe, A class assignment problem: a case study at Tokyo Institute of Technology, *Communications of the OR Society of Japan* 36 (1991) 85–89.

[9] H. Konno and A. Wijayanayake, A mean-absolute deviation portforio optimization model under transaction cost, *J. Oper. Res. Soc. Japan* 42 (1999) 422–435.

[10] H. Konno and R. Yamamoto, Global optimization vs integer programming in portfolio optimization, ISE03-07 Department of Industrial and Systems Engineering, Chuo University, 2003 (to appear in *J. Global. Optim*).

[11] T. Larsson, A. Migdalas and M. Ronnqvist, A lagrangean heuristic for the capacitated concave minimum cost network flow problem, *European J. Oper. Res.* 78 (1994) 116–129.

[12] T.Q. Phong et al, Decomposition branch and bound method for globally solving linearly constrained indefinite quadratic minimization problems, *Oper. Res. Lett.* 17 (1995) 215–220.

[13] R.M. Soland, Optimal facilities location with concave costs, *Oper. Res.* 22 (1974) 373–382.

[14] H. Tuy, S. Ghannadan, A. Migdalas and P. Varbrand, The minimum concave cost network flow problems with fixed number of sources and nonlinear arc costs, *J. Global. Optim.* 6 (1995) 135–151.

[15] L. Wolsey, *Integer Programming*, John Wiley & Sons, New York, 1998.

[16] Y. Yajima and H. Konno, An algorithm for concave production cost network flow problem, *Japan J. Indust. Appl. Math.* 16 (1999) 243–256.

[17] W.I. Zangwill, Minimum concave cost flows in certain networks, *Management Science* 14 (1968) 429–450.

Manuscript received 3 September 2005
revised 28 December 2005
accepted for publication 3 January 2006

Hiroshi Konno
Department of Industrial and Systems Engineering, Chuo University, 1-13-27 Kasuga Bunkyo-ku, Tokyo 112-8552, Japan
E-mail address: konno@indsys.chuo-u.ac.jp

Takaaki Egawa
Department of Industrial and Systems Engineering, Chuo University, 1-13-27 Kasuga Bunkyo-ku, Tokyo 112-8552, Japan
E-mail address: etaka-ki@fine.ocn.ne.jp