CONCAVE COST SUPPLY MANAGEMENT PROBLEM WITH SINGLE MANUFACTURING UNIT

S. S. Chauhan[1], A. V. Eremeev[2], A. A. Kolokolov[3] and V. V. Servakh[4]

[1] *INRIA –SAGEP, UFR Scientifique, Universite de Metz, address: Ile du Saulcy-57045 Metz, France e-mail: Singh.Chauhan@loria.fr*;

[2-4] *Omsk Branch of Sobolev Institute of Mathematics SB RAS address: 13, Pevtsov str., 644099, Omsk, Russia e-mail: {eremeev; kolo; svv}@iitam.omsk.net.ru*

Abstract: The concave cost supply management problem consists in optimization of product delivery from a set of providers to the manufacturing units (single unit and single planning period in our case) with respect to delivery cost functions of concave type. Given the lower and upper bounds on the shipment size for each provider, the demand of the manufacturing unit has to be satisfied. In this chapter it is shown that it is *NP*-hard even to find a feasible solution to this problem. Considering the problem in integer programming formulation we propose a pseudo-polynomial algorithm, using the dynamic programming technique. Some possible approaches to solving the problem with multiple manufacturing units are discussed.

Key words: Concave cost, supply, dynamic programming, integer programming, complexity

1. INTRODUCTION

In this chapter, we consider the problem where a set of providers supply one type of product to a manufacturing unit, the quantity that can be delivered lies between the given minimum and maximum values, and the costs proposed by each provider are concave functions of quantity being delivered. The concavity assumption reflects a common situation taking place in industry since usually the unit cost of products and the transportation unit cost decrease as the size of an order increases. This problem is similar to, but different from the well known transportation problem with concave costs (see e.g. [1, 8]), but in our case a provider either delivers a quantity of product that lies between a lower bound and an upper bound or delivers nothing. The lower bound is the economical production quantity imposed by the provider and the upper bound is a technical constraint: it is the maximum quantity the provider is able to produce during the period under consideration. Formally the problem is stated as follows:

$$\sum_{i=1}^{n} k_i(x_i) \to \min, \tag{1}$$

$$\sum_{i=1}^{n} x_i = A, \tag{2}$$

$$x_i \in \{0\} \cup [m_i, M_i] \text{ for } i=1,2,\ldots,n. \tag{3}$$

Here n is the number of providers, x_i is the quantity of product delivered to the manufacturing unit from provider i; A is the total amount of product required for the manufacturing unit; m_i is the minimum quantity the provider i is prepared to deliver due to the economical reasons; M_i is the maximum quantity the provider i is able to deliver. All quantities of product here and in the rest of the chapter refer to some standard planning period (e.g. one week). The cost $k_i(x_i)$ is

$$k_i(x_i) = \begin{cases} 0 & \text{if} \quad x_i = 0 \\ a_i + g_i(x_i) & \text{if} \quad x_i > 0, \end{cases}$$

where $a_i \geq 0$ and $g_i(x_i) \geq 0$ are concave and non-decreasing functions when x_i is positive, $i=1,\ldots,n$. This problem formulation was suggested in [2] not only for the single manufacturing unit but also for the general case with multiple manufacturers (see the further discussion in Sect. 3). A number of useful properties of the problem have been shown and several heuristic algorithms were proposed and tested there. Our goal here is to investigate the exact solution methods and complexity issues of the problem with single manufacturing unit and some of its extensions.

Firstly, in Sect. 2 we demonstrate the *NP*-hardness of the problem and show that the standard dynamic programming approach allows to find the optimum in pseudo-polynomal time. A discussion on extension of exact solution methods for the case of several providers, and the conclusions are contained in Sect. 3 and 4.

2. PROBLEM COMPLEXITY AND PSEUDO-POLYNOMIAL TIME ALGORITHM

Theorem 1. *Finding a feasible solution to supply management problem (1)-(3) with rational input parameters is NP-hard.*

Proof: Let there be a polynomial time algorithm which finds a feasible solution satisfying (2) and (3) when such solutions exist. Assume that $A = \frac{1}{2}\sum_{i=1}^{n} m_i$, M_i , m_i are integer and $M_i = m_i$ for all $i = 1, 2, ..., n$. By substitution $x_i = z_i \cdot m_i$, $i = 1,2,...,n$ conditions (2) and (3) for this case may be written as follows:

$$\sum_{i=1}^{n} m_i z_i = \frac{1}{2}\sum_{i=1}^{n} m_i \ , \tag{4}$$

$$z_i \in \{0, 1\} \text{ for } i=1,2,...,n . \tag{5}$$

The polynomial time algorithm mentioned above is suitable to recognize the consistency of (4) and (5) which is equivalent to solving the *NP*-complete SUBSET SUM problem [3]. Q.E.D.

In what follows we suppose that all A, m_i, M_i are integer, which is a certain limitation, nevertheless its influence may always be reduced by choosing the sufficiently fine-grained scale of the variables. In our analysis we will use a fact similar to Result 1 from [2], although here we do not require that functions $g_i(x)$ are continuously differentiable:

Theorem 2. *If problem (1)-(3) is solvable, then there exists an optimal solution* $X = \{x_1, x_2, ..., x_n\}$ *such that* $x_i = m_i$ *or* $x_i = M_i$ *or* $x_i = 0$ *for* $i = 1,2,...,n,$ *except for at most one* $j \in \{1,2,...,n\}$ *for which* $m_j < x_j < M_j$.

Proof: Let $X^1 = \{x_1^1, x_2^1, ...x_n^1\}$ be an optimal solution to problem (1)-(3). Assume there exists a pair $i, j \in \{1,2,....,n\}$ such that $m_i < x_i^1 < M_i$ and $m_j < x_j^1 < M_j$. Firstly, consider the case when we have

$$k_i(x_i^1+\delta)-k_i(x_i^1)\le k_j(x_j^1)-k_j(x_j^1-\delta), \tag{6}$$

where by definition $\delta=\min(M_i-x_i^1, x_j^1-m_j)$. Then we can set:

$$x_i^2=x_i^1+\delta \tag{7}$$
$$x_j^2=x_j^1-\delta. \tag{8}$$

Let us denote by X^2 the new solution obtained after replacing x_i^1 by x_i^2 and x_j^1 by x_j^2 in X^1. Then adding (7) and (8) we see that (2) and (3) still hold for X^2. Besides that, $k_i(x_i^2)+k_j(x_j^2)\le k_j(x_j^1)+k_i(x_i^1)$, so X^2 is optimal too.

Now, if (6) does not hold, analogously we can treat the case when

$$k_i(x_i^1)-k_i(x_i^1-\Delta)\ge k_j(x_j^1+\Delta)-k_j(x_j^1), \tag{9}$$

where by definition $\Delta=\min(M_j-x_j^1, x_i^1-m_i)$.

Finally, let us prove that other options are impossible, i.e. an assumption that both

$$k_i(x_i^1+\delta)-k_i(x_i^1) > k_j(x_j^1) - k_j(x_j^1-\delta), \tag{10}$$
$$k_i(x_i^1)-k_i(x_i^1-\Delta) < k_j(x_j^1+\Delta) - k_j(x_j^1) \tag{11}$$

hold, will lead to a contradiction. Indeed, since $k_j(x_j)$ is concave, so $\frac{\Delta+\delta}{\delta}k_j(x_j^1)\ge k_j(x_j^1+\Delta)+\frac{\Delta}{\delta}k_j(x_j^1-\delta)$, i.e. $\frac{\Delta}{\delta}(k_j(x_j^1)-k_j(x_j^1-\delta))\ge k_j(x_j^1+\Delta)-k_j(x_j^1)$, and by (10) we have:

$$k_i(x_i^1+\delta)-k_i(x_i^1) > \frac{\delta}{\Delta}(k_j(x_j^1+\Delta)-k_j(x_j^1)).$$

Combining this with (11) we conclude that:

$$\Delta\, k_i(x_i^1+\delta)+\delta\, k_i(x_i^1-\Delta) > (\Delta+\delta)\, k_i(x_i^1),$$

which implies a contradiction with concavity of $k_i(x_i)$.

Thus, either (6) or (9) must hold, and consequently we always have:

$$\sum_{i=1}^{n} k_i\,(x_i^2) \le \sum_{i=1}^{n} k_i\,(x_i^1)\,.$$

Continuing the same process will lead to a solution X indicated in the statement of the theorem. Q.E.D.

Since A, m_i, M_i are integer, so by Theorem 2 there exists an optimal solution where all x_i, $i=1,2,\ldots,n$ are integer also. Thus the original continuous problem can be considered as a discrete optimization problem. In our analysis of this problem we will use the standard dynamic programming technique. Let us consider all possible integer values of variable x_n.

1. If x_n=0, then $k_n(0)=0$, and the problem reduces to the following:

$$\sum_{i=1}^{i=n-1} k_i\,(x_i) \to \min\,,$$

$$\sum_{i=1}^{i=n-1} x_i = A\,,$$

$$x_i \in \{0\} \cup [m_i, M_i] \text{ for } i=1,2,\ldots,n-1.$$

Let $\varphi(p,a)$ denote the optimal objective function value for the problem:

$$\varphi(p,a) = \min_x \left\{ \sum_{i=1}^{p} k_i\,(x_i) \right\},$$

$$\sum_{i=1}^{p} x_i = a\,,$$

$$x_i \in \{0\} \cup [m_i, M_i] \text{ for } i = 1,2,\ldots,p\,.$$

According to this notation in case x_n=0 we have $\varphi(n,A) = \varphi(n-1,A)$.

2. If we consider some fixed $m_n \le x_n \le M_n$ then the problem (1)-(3) reduces to:

$$\sum_{i=1}^{i=n-1} k_i\,(x_i) + k_n\,(x_n) = \varphi(n-1, A - x_n) + k_n\,(x_n) \to \min\,,$$

$$\sum_{i=1}^{i=n-1} x_i = A - x_n\,,$$

$$x_i \in \{0\} \cup [m_i, M_i] \text{ for } i=1,2,\ldots,n-1.$$

Thus, if the positive shipment x_n is fixed then we need to solve the problem for n-1 providers and the smaller amount of product remaining. Combining the cases 1 and 2 what we have to find is the value of x_n that minimizes the goal function:

$$\varphi(n, A) = \min\left\{\varphi(n-1, A),\ \min_{m_n \le x_n \le M_n} (\varphi(n-1, A - x_n) + k_n(x_n))\right\}.$$

To find $\varphi(n, A)$ here we need the solutions to all problems of dimension n-1, which may be computed recursively through the problem with n-2 variables, etc. Finally we have the general formula:

$$\varphi(p, a) = \min\left\{\varphi(p-1, a),\ \min_{m_p \le x_p \le M_p;\, x_p \le a} (\varphi(p-1, a - x_p) + k_p(x_p))\right\},$$

$$p = 1,2,..., n;\ \ a = 0,1,2,..., A.$$

The computations with this formula are carried out through double loop: with $p=1,2,\dots,n$, and with a=0,1,2,…,A, assuming the initial conditions $\varphi(0,0)=0$, $\varphi(0, a) = \infty,\ a = 1,2,..., A$.

Calculation of $\varphi(p, a)$ requires not more than $M_p - m_p + 2$ comparison operations. So the total number of comparisons for solving (1)-(3) is bounded by $A \cdot \sum_{p=1}^{n} (M_p - m_p + 2)$.

Thus, there exists a pseudo-polynomial time algorithm for solving this problem (here we imply that functions $k_i(x_i)$, $i=1,2,\dots,n$ are computable in polynomial time). In fact if we divide the problem data input string into two parts: substring *s'* for encoding the functions $k_1, k_2 \dots, k_n$ and substring *s* for the rest of the data, then even in case of unbounded growth of the values of numeric parameters encoded in *s'* the running time will remain polynomial in the length of *s*. Therefore we have

Theorem 3. *Let s be the input substring, encoding A,* $m_1, m_2, \dots, m_n, M_1, M_2, \dots, M_n$. *If functions* $k_i(x_i)$, $i=1,2,\dots,n$ *are polynomial time computable in length of s for all* $0 \le x_i \le A$, *then there exists a pseudo-polynomial time algorithm (with respect to input substring s) solving problem (1)-(3).*

Note that the complete enumeration of solutions has the time complexity $O(n3^{n-1})$. If A is large and n is small, the complete enumeration method may be advantageous. However with bounded $A, m_1, m_2, \dots, m_n, M_1, M_2, \dots, M_n$ the running time of the dynamic programming will be smaller by an exponential factor.

3. SOME APPROACHES TO SOLVING MORE GENERAL PROBLEMS

It is interesting to consider the extension of the concave cost supply management problem to the case with multiple manufacturing units as it was formulated in [2]:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} k_{ij}(x_{ij}) \rightarrow \min, \tag{12}$$

$$\sum_{i=1}^{n} x_{ij} = A_j, \; j=1,2,...,m, \tag{13}$$

$$\sum_{j=1}^{m} x_{ij} \le M_i, \; i=1,2,...,n, \tag{14}$$

$$x_{ij} \in \{0\} \cup \lfloor m_{ij}, M_i \rfloor \text{ for } i=1,2,...,n; \; j=1,2,...,m. \tag{15}$$

Here n is the number of providers and m is the number of manufacturing units, x_{ij} is the quantity of product delivered to the manufacturing unit j from provider i; A_j is the total amount of product required for the manufacturing unit j; m_{ij} is the minimum quantity the provider i is prepared to deliver to the manufacturing unit j; M_i is the maximum quantity the provider i is able to deliver to the manufacturing units. The cost $k_{ij}(x_{ij})$ is

$$k_{ij}(x_{ij}) = \begin{cases} 0 & \text{if} \quad x_{ij} = 0 \\ a_{ij} + g_{ij}(x_{ij}) & \text{if} \quad x_{ij} > 0, \end{cases}$$

where $a_{ij} \ge 0$ and $g_{ij}(x_{ij}) \ge 0$ are concave and non-decreasing functions when x_{ij} is positive, $i=1,\ldots,m$, $j=1,\ldots,n$.

The necessary conditions of the optimum formulated in [2] for this problem permit the development of a dynamic programming approach similar to that described above. However the time and memory resources required by such an algorithm might present serious obstacles. In this connection it seems to be appropriate to use the piecewise linear approximations of the functions $k_{ij}(x_{ij})$, since then the problem can be formulated as an integer linear programming problem. For example in the case of linear costs $k_{ij}(x_{ij}) = c_{ij}x_{ij}$, $i=1,\ldots,m$, $j=1,\ldots,n$ introducing supplementary Boolean variables z_{ij} we obtain the following mixed-integer problem:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij} x_{ij} \to \min$$

$$\sum_{i=1}^{m} x_{ij} = A_j, \ \ j = 1,\dots,n,$$

$$\sum_{j=1}^{n} x_{ij} \le M_i, \ \ i = 1,\dots,m,$$

$$\left.\begin{array}{l} x_{ij} \ge z_{ij} m_{ij} \\ x_{ij} \le z_{ij} M_i \\ x_{ij} \ge 0 \\ z_{ij} \in \{0,1\} \end{array}\right\} \ i = 1,\dots,m, \ \ j = 1,\dots,n.$$

A number of approaches may be used for solving such problem: the branch and bound methods of Land and Doig type (see e.g. [7]), the Benders decomposition method and cutting plane algorithms (see e.g. [1,4]), *L*-class enumeration algorithms [5,6], etc. Note that a further generalization of problem (12)-(15) may be done through the assumption that the size of shipment x_{ij} belongs to a range consisting of several intervals for all i and j. The approaches mentioned above could be extended to this case as well.

4. CONCLUSIONS

The concave cost supply management problem with single manufacturing unit was shown to be *NP*-hard and a dynamic programming pseudo-polynomial time algorithm was suggested for it. The possible approaches to solving the more general problem with multiple manufacturing units were discussed.

We expect that the further research will be aimed at the elaboration of the solution methods discussed in Sect. 3 and their theoretical and experimental comparison. Another direction for the further research is the analysis of a problem with lower bounds on consumption of product instead of the exact conditions (2) and (13) assumed in this chapter. In such a modification (at least in the single-unit case) the feasible solution is easier to find and fast approximation algorithms are appropriate.

ACKNOWLEDGEMENTS

The research was supported by INTAS Grant 00-217. The authors would also like to thank J.-M. Proth, G.G. Zabudsky and L.A. Zaozerskaya for the discussions and helpful comments on this paper.

REFERENCES

1. Bakhtin. A.E., Kolokolov A.A. and Korobkova. Z.B., 1978. *Discrete Problems of Production-Transportation Type*, Nauka, Novosibirsk. (in Russian).
2. Chauhan S.S. and Proth J.-M., 2003. The concave cost supply problem. *European Journal of Operational Research,* 148 (2), 374-383.
3. Garey. M.R. and Johnson. D.S., 1979. *Computers and Intractability. A Guide to the Theory of NP-Completeness,* W.H. Freeman and Company, San Francisco.
4. Hu T.C., 1970. *Integer Programming and Network Flows,* Addison-Wesley Pbl.
5. Kolokolov A.A., 1994. Decomposition algorithms for solving of some production-transportation problems. In: *Preprints of Triennial Symposium on Transportation Analysis II.*, Vol. 1, Capri, Italy, 179-183.
6. Kolokolov. A.A. and Levanova. T.V., 1996. Decomposition and *L*-class enumeration algorithms for solving some location problems. *Vestnik Omskogo Universiteta*, 1, Omsk, OmSU, 21-23. (in Russian).
7. Schrijver. A., 1986. *Theory of Linear and Integer Programming,* John Wiley Sons, Vol. 2.
8. Yan. S. and Luo. S.-C., 1999. Probabilistic local search algorithms for concave cost transportation network problems. *European Journal of Operational Research,* 117, 511-521.