Mathematical & Computational Applications, Vol. 8, No. 1 pp. 27-34, 2003

ON COMPLEXITY OF A GLOBAL OPTIMIZATION PROBLEM

Urfat G. Nuriyev
Ege University, Department of Mathematics, 35100, Bornova-Izmir / TURKEY
urfat@sci.ege.edu.tr

Abstract: The Solution of the Subproblem of the Cutting Angle Method of Global Optimization for problems of minimizing Increasing Positively Homogeneous of degree one functions is proved to be NP-Complete. For the proof of this fact we formulate another problem which we call "Dominating Subset with Minimal Weight". The solution of this problem is also NP-Complete.
Keywords: Global Optimization Problem, Gutting Angle Method, Dominant Subset with Minimal Weight Problem, NP-Complete

1. INTRODUCTION

Cutting Angle Method described in the papers [1-5] was developed for solving a broad class of global optimization problems. This method is an iterative one requiring the solution of a subproblem (minimizing functions $f(x)$, defined on the set $S=\{x|\sum_{i=1}^{n} x_i =1,\ x_i \geq 0,\ i=1, \ldots, n\}$, where $x=(x_1,\ldots,x_n)$) , which in its turn is, generally, a global optimization problem. Different algorithms based on discrete programming and dynamic programming were offered for the solution of this subproblem in the papers[1, 3 - 6].

In this paper we study some properties of the optimal solutions of the subproblem and by means of these properties prove that this problem is equivalent to a problem of Boolean programming, which we call "Dominant Subset of Minimal Weight". The last problem can be used in other situations as well. By transformation of this problem to the Knapsack problem we prove that it is NP-Complete, therefore the subproblem considered above is also NP-Complete.

2. FORMULATION OF THE PROBLEM

Let (l_i^k) be an $(m*n)$ matrix, $m\geq n$, with m rows l^k, $k = 1, \ldots, m$, and n columns, $i = 1, \ldots, n$. All elements $l_i^k \geq 0$. The first n rows of (l_i^k) matrix form a diagonal matrix, i.e., $l_i^k>0$, only for $k = i$, $i=1, \ldots, n$.

Introduce a function

$h(x)=\max\limits_{k} \min\limits_{i\in I(l^k)} l_i^k x_i$, where $I(l^k)=\{i\colon l_i^k>0\}$.

The problem considered in this paper is formulated as

The Subproblem

$$\textit{Minimize } h(x) \tag{1}$$

subject to

$$x\in S=\{x|\sum_{i=1}^{n} x_i =1,\ x_i \geq 0,\ i=1,\ldots,n\} \tag{2}$$

3. SOME RESULTS CONCERNING OPTIMAL SOLUTIONS

The optimal solution for the case $m=n$ is as follows:

Theorem 1. [6] *If $m=n$, then The Subproblem (1)-(2) has a unique solution*

$$x_i = h(x)/l_i^i,\ i=1,\ \ldots,\ n, \text{ where} \tag{3}$$

$$h(x)=\min h(x)=1/\sum_{i=1}^{n}\frac{1}{l_i^i}. \tag{4}$$

Corollary 1. [6] *$\min h(x)$ for $m=n$ is the lower bound of $\min h(x)$ for any $m>n$.*

If $m>n$, then two cases are possible.

Case 1. For each $k>n$, there exists i, such that $l_i^k \leq l_i^i$.

Theorem 2. [6] If for every $k>n$, there exists i, such that $l_i^k \leq l_i^i$, then the Subproblem possesses a unique solution, which coincides with the solution for $m = n$.

Case 2. $\exists\, K$, such that $l_i^k > l_i^i, \forall i=1,\ldots,n$, for $\forall k \in K$, i .e. the conditions of Theorem 2 are not satisfied. We will use the following notation:

$$h_k(x) = \min_{i\in I(l^k)} l_i^k x_i,\ \ k = 1, 2, \ldots, m, \tag{5}$$

$$h(x) = \max_{k=\overline{1,m}} h_k(x), \tag{6}$$

$$h^* = \min_{x\in S} h(x). \tag{7}$$

Clearly, if x^* is a solution of the subproblem (1)-(2), then for each k ($k = 1, 2, \ldots, m$) there is i_k such that $h_k(x^*) = l_{i_k}^k x_{i_k}^*$ and for $k \leq n$ we have $i_k = k$, i.e. $h_k(x^*) = l_k^k x_k^*$. Let $x\in S$ and for each i ($i = 1, 2, \ldots, n$) define

$$k_i(x) = \arg\max_{k=\overline{1,m}}\left\{h_k(x)\,\middle|\,h_k(x) = l_{i_k}^k x_i, i_k = i\right\}. \tag{8}$$

Clearly, if for the given i there is no $k > n$ with $h_k(x) = l_i^k x_i$, then $k_i(x) = i$, i.e.

$$h_i(x) = l_i^i x_i . \tag{9}$$

Remark 1. If for a given x, we call the smallest element of each row (i.e. the element $l_i^k x_i$ which equals to $h_k(x)$) a *chosen element*, then for each column i the row number of the largest chosen element of this column will be $k_i(x)$.

Let us define

$$\overline{l_i(x)} = l_i^{k_i(x)},\ \ i = 1, 2, \ldots, n. \tag{10}$$

Theorem 3. *If $x^*=(x_1^*, x_2^*, \ldots, x_n^*)$ is an optimal solution of the Subproblem (1)-(2), then*

$$\overline{l_i(x^*)}x_i^* = h^*, \quad i = 1, 2, \ldots, n. \tag{11}$$

Proof. Let $I = \{1, 2, \ldots, n\}$. It is clear from (6), (8) and (10)

$$\overline{l_i(x^*)}x_i^* \le h^*, \quad i = 1, 2, \ldots, n. \tag{12}$$

Now suppose that Theorem 3 is not true, i.e., there are $i_1, i_2, \ldots, i_t \in I$ such that

$$\overline{l_{i_k}(x^*)}x_{i_k}^* < h^*, \quad k = 1, 2, \ldots, t. \tag{13}$$

Let $I_t = \{i_1, i_2, \ldots, i_t\}$ and $\bar{I}_t = I/I_t = \{i_{t+1}, i_{t+2}, \ldots, i_n\}$. Then we obtain from (12) and (13)

$$\overline{l_i(x^*)}x_i^* = h^*, \quad i \in \bar{I}_t. \tag{14}$$

Let us consider new variables $\overline{x_i}$, $i \in I$:

$$\overline{x_i} = x_i^* + \alpha_i, \ i \in I_t, \tag{15}$$

$$\overline{x_i} = x_i^* - \alpha_i, \ i \in \bar{I}_t, \tag{16}$$

where $\alpha_i > 0$, $i \in I$ and the following conditions are satisfied :

$$\sum_{i \in I} \overline{x_i} = 1, \tag{17}$$

$$\overline{l_{i_1}(x^*)}\,\overline{x_{i_1}} = \overline{l_{i_2}(x^*)}\,\overline{x_{i_2}} = \ldots = \overline{l_{i_t}(x^*)}\,\overline{x_{i_t}} = \overline{l_{i_t+1}(x^*)}\,\overline{x_{i_t+1}} = \ldots = \overline{l_{i_n}(x^*)}\,\overline{x_{i_n}} \tag{18}$$

Substituting values of the variables $\overline{x_i}$ defined by (15) and (16) in (17) and (18) we obtain:

$$\sum_{i \in I_t}(x_i^* + \alpha_i) + \sum_{i \in \bar{I}_t}(x_i^* - \alpha_i) = 1,$$

$$\sum_{i \in I} x_i^* + \sum_{i \in I_t}\alpha_i - \sum_{i \in \bar{I}_t}\alpha_i = 1.$$

Since $x_i^* \in S$ we have $\sum_{i \in I} x_i^* = 1$ and we obtain:

$$\sum_{i \in I_t}\alpha_i = \sum_{i \in \bar{I}_t}\alpha_i. \tag{19}$$

(18) leads to the following system of equations:

$$\left.\begin{array}{rcl}
\overline{l_{i_1}(x^*)}(x_{i_1}^* + \alpha_{i_1}) & = & \overline{l_{i_n}(x^*)}(x_{i_n}^* - \alpha_{i_n}), \\
\overline{l_{i_2}(x^*)}(x_{i_2}^* + \alpha_{i_2}) & = & \overline{l_{i_n}(x^*)}(x_{i_n}^* - \alpha_{i_n}), \\
\multicolumn{3}{c}{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} \\
\overline{l_{i_t}(x^*)}(x_{i_t}^* + \alpha_{i_t}) & = & \overline{l_{i_n}(x^*)}(x_{i_n}^* - \alpha_{i_n}), \\
\overline{l_{i_t+1}(x^*)}(x_{i_t+1}^* - \alpha_{i_t+1}) & = & \overline{l_{i_n}(x^*)}(x_{i_n}^* - \alpha_{i_n}), \\
\multicolumn{3}{c}{\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots} \\
\overline{l_{i_n-1}(x^*)}(x_{i_n-1}^* - \alpha_{i_n-1}) & = & \overline{l_{i_n}(x^*)}(x_{i_n}^* - \alpha_{i_n}).
\end{array}\right\} \tag{20}$$

If we add equation (19) to this system (20) consisting of (n -1) equations we will have n equations for finding n undetermined $\alpha_i > 0$, $i \in I$. Finding α_i's from this system we can calculate new values of ($\overline{x_i}$, $i \in I$) by formulas (15) and (16).

Let $I_t \cap I(l^k) = I_t(l^k)$ and $\bar{I}_t \cap I(l^k) = \bar{I}_t(l^k)$.

Consider $h_k(\bar{x})$, where $\bar{x} = \left(\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}\right)$.

$$h_k(\bar{x}) = \min_{i \in I(l^k)} l_i^k \,\overline{x_i} = \min \{ \min_{i \in I_t(l^k)} l_i^k \,\overline{x_i} \,, \min_{i \in \bar{I}_t(l^k)} l_i^k \,\overline{x_i} \}. \tag{21}$$

Assume that

$$h_k(x^*) = l_{i_k^*}^k \, x_{i_k^*}^*, \; k=1,2,\ldots,m. \tag{22}$$

For each k we have two cases :

Case1. $i_k^* \in \bar{I}_t(l^k)$ then from (22), (10), (16) and (14) we obtain :

$$\min_{i \in \bar{I}_t(l^k)} l_i^k \,\overline{x_i} \le l_{i_k^*}^k \,\overline{x_{i_k^*}} \le \overline{l_{i_k^*}\left(x^*\right)} \,\overline{x_{i_k^*}} = \overline{l_{i_k^*}\left(x^*\right)} \, x_{i_k^*}^* - \overline{l_{i_k^*}\left(x^*\right)} \,\alpha_{i_k^*} = \mathrm{h}^* - \overline{l_{i_k^*}\left(x^*\right)} \,\alpha_{i_k^*} < h^* \tag{23}$$

Case2. $i_k^* \in I_t(l^k)$ then from (22), (10), (20), (16) and (14) we obtain :

$$\min_{i \in I_t(l^k)} l_i^k \,\overline{x_i} \le l_{i_k^*}^k \,\overline{x_{i_k^*}} \le \overline{l_{i_k^*}\left(x^*\right)} \,\overline{x_{i_k^*}} = \overline{l_{i_n}\left(x^*\right)} \overline{x_{i_n}} = \overline{l_{i_n}\left(x^*\right)} \, x_{i_n}^* - \overline{l_{i_n}\left(x^*\right)} \,\alpha_{i_n} = h^* - \overline{l_{i_n}\left(x^*\right)} \,\alpha_{i_n} < h^* \tag{24}$$

Now from (21), (23) and (24) we have $h_k(\bar{x}) < h^*$, $k = 1, 2, \ldots$, m. Therefore it follows from (6) and (7) that $h(\bar{x}) = \max\limits_k h_k(\bar{x}) < h^*$ and $\bar{h} < h^*$. But this is a contradiction with optimality of the solution h^* of the problem (1)-(2) .

Corollary 2. *The optimal solution of the Subproblem (1)-(2) is given as :*

$$x_i^* = h^* / \overline{l_i\left(x^*\right)}, \; i = 1, \ldots, n, \textit{ where} \tag{25}$$

$$h^* = 1 / \sum_{i=1}^{n} \frac{1}{\overline{l_i\left(x^*\right)}}. \tag{26}$$

Substituting (12) in (25) and (26) we obtain :

$$x_i^* = h^* / l_i^{k_i(x^*)}, \; i=1,\ldots,n, \textit{ where} \tag{27}$$

$$h^* = 1 / \sum_{i=1}^{n} \frac{1}{l_i^{k_i(x^*)}} \tag{28}$$

4. TRANSFORMATION OF THE SUBPROBLEM TO AN EQUIVALENT PROBLEM

Formulas (27), (28) are obtained from (3) and (4) by substituting $l_i^{k_i(x^*)}$ instead of l_i^i. This means that if the condition of Theorem 2 is not satisfied, the optimal solution will be obtained by substituting some of l_i^i's (those i's for which $h_i(x^*) \neq l_i^i x_i^*$, i.e. for which the condition (11) is not satisfied) by $l_i^{k_i}$.

Now let us study the change of the function $h = 1/\sum_{i=1}^{n} \frac{1}{l_i^i}$ in this substitution.

Without restriction of generality we can assume that h' is obtained from h by substitution of only two elements (say l_1^1 and l_2^2):

$$h = \frac{1}{\left(\frac{1}{l_1^1} + \frac{1}{l_2^2}\right) + \frac{1}{l_3^3} + \ldots + \frac{1}{l_n^n}}; \qquad h' = \frac{1}{\left(\frac{1}{l_1^{k_1}} + \frac{1}{l_2^{k_2}}\right) + \frac{1}{l_3^3} + \ldots + \frac{1}{l_n^n}}.$$

Since the condition of Theorem 2 is not satisfied then $l_1^{k_1} > l_1^1$ and $l_2^{k_2} > l_2^2$.

Denoting $L = \frac{1}{l_1^1} + \frac{1}{l_2^2} + \ldots + \frac{1}{l_n^n}$ and $\frac{1}{l_i^i} - \frac{1}{l_i^{k_i}} = u_i^{k_i}$, we will have $h = \frac{1}{L}$ and

$$h' = \frac{1}{\left((\frac{1}{l_1^{k_1}} - \frac{1}{l_1^1}) + (\frac{1}{l_2^{k_2}} - \frac{1}{l_2^2})\right) + (\frac{1}{l_1^1} + \frac{1}{l_2^2}) + (\frac{1}{l_3^3} + \ldots + \frac{1}{l_n^n})} = \frac{1}{-u_1^{k_1} - u_2^{k_2} + L},$$

$$h' - h = \frac{1}{L - u_1^{k_1} - u_2^{k_2}} - \frac{1}{L} = \frac{u_1^{k_1} + u_2^{k_2}}{L(L - (u_1^{k_1} + u_2^{k_2}))} = \frac{u_1^{k_1} + u_2^{k_2}}{L^2 - L(u_1^{k_1} + u_2^{k_2})}.$$

We see that if $u_1^{k_1} + u_2^{k_2}$ decreases then $(h' - h)$ also decreases. Therefore we must make changes $l_i^{k_i} \to l_i^i$, such that the sum above decreases, i.e.

$$\sum_i u_i^{k_i} \to \min. \tag{29}$$

We will use the following notation for simplicity:

$$p = m - n, \quad u_i^j = \frac{1}{l_i^i} - \frac{1}{l_i^{j+n}}, \quad i = 1,2,\ldots,n\,; j = 1,2,\ldots,p.$$

Clearly u_i^j is the increment of the denominator of the fraction that expresses the function h in the substitution $l_i^{j+n} \to l_i^i$.

Let us define the following function:

$$Sg(x) = \begin{cases} 1, & \text{if } x \geq 0, \\ 0, & \text{if } x < 0 \end{cases}$$

and consider variables x_i^j, $i = 1,2,\ldots,n$; $j = 1,2,\ldots,p$:

$$x_i^j = \begin{cases} 1, & \textit{if the substitution } l_i^{j+n} \to l_i^j \text{ is accomplihed} \\ 0, & \text{otherwise} \end{cases}$$

So the Subproblem (1)-(2) is transformed into the following Boolean (0 – 1) programming problem :

$$\sum_{i=1}^{n}\sum_{j=1}^{p} u_i^j x_i^j \to \min_{x_i^j} \tag{30}$$

$$\sum_{i=1}^{n} x_i^j \le 1, \; j=1,2,\ldots,p, \tag{31}$$

$$\sum_{j=1}^{p} x_i^j \le 1, \; i=1,2,\ldots,n, \tag{32}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{p} x_i^j \ge 1, \tag{33}$$

$$\sum_{i=1}^{n} y_i^j \ge 1, \; j=1,2,\ldots,p, \tag{34}$$

$$x_i^j = 0 \vee 1, \; i=1,2,\ldots,n; \; j=1,2,\ldots,p, \tag{35}$$

$$y_i^j = Sg\left(\max_{k=1,p}\left\{u_i^k x_i^k\right\} - u_i^j\right), \; i=1,2,\ldots,n; \; j=1,2,\ldots,p, \tag{36}$$

where the condition (30) is obtained from the condition (29), the condition (31) from (5) and (32) from (6). Since the condition of Theorem 2 is not satisfied the condition (9) will not be satisfied for all i's (i=1,2,…,n), i.e. at least one substitution $l_i^{j+n} \to l_i^j$ will be accomplished in the optimal solution and this means the condition (33). The condition (34) is obtained from (7), (8), (10) and the definition of the variables y_i^j (i.e. from (36)).
So we can obtain the optimal solution of the Subproblem (1)-(2) by the substitution $l_{i^*}^{j^*}$ by $l_{i^*}^{j^*+n}$ in formulas (3)-(4) for all $x_{i^*}^{j^*} = 1$ in the optimal solution of the problem (30)-(36) and vice versa. In other words the following Theorem holds.
Theorem 4. *The Subproblem (1)-(2) and Problem (30)-(36) are equivalent.*

5. DOMINATING SUBSET OF MINIMAL WEIGHT PROBLEM

Let us call the problem (30)-(36) “Dominating Subset with Minimal Weight”. We can interpret this problem as follows:
Let (u_i^j) be a $(p * n)$ matrix, with p rows, j = 1, 2, …, p and n columns, i = 1, 2, …, n and nonnegative u_i^j for all i,j.

The task is to choose some elements of the matrix such that:

- Each row contains a chosen element, or contains some element which is less than some chosen element located in its column;
- The sum of the chosen elements is minimal.

We can give the following economic interpretation of this problem:
A task consisting of p (j = 1, 2, …, p) operations can be accomplished by n (i = 1, 2, …,n)

processors. Suppose that the matrix (u_i^j) gives the time necessary for accomplishment of the task as follows: if

$$u_i^{j_1} \le u_i^{j_2} \le \ldots \le u_i^{j_p} \tag{37}$$

for column i, then $u_i^{j_1}$ is the time (or cost) for accomplishment of operation j_1 by processor i; $u_i^{j_2}$ is the time for the accomplishment of operations j_1 and j_2 by processor i, and so on, at last $u_i^{j_p}$ is the time for the accomplishment of all operations ($j_1, j_2, \ldots, j_p$) by processor i. The problem is to distribute operations between the processors minimizing the total time (or the total cost) required for accomplishment all tasks.

6. COMPLEXITY OF THE SUBPROBLEM

Now we transform problem (30)-(36) into the equivalent Multiple-Choice Knapsack problem with $p*n = q$ binary variables and p constraints.
Coefficients of the objective function of this problem are defined as
$c_1 = u_1^1,\ c_2 = u_1^2, \ldots, c_p = u_1^p,\ c_{p+1} = u_2^1,\ c_{p+2} = u_2^2, \ldots, c_{2p} = u_2^p,\ c_{2p+1} = u_3^1, \ldots, c_q = u_n^p$
Consider a $1 \le k \le q$. Suppose $c_k = u_i^j$ for some i,j, i.e., c_k equals to some element in i-th column and j-th row of matrix (u_i^j) and for i-th column of this matrix the condition (37) above is satisfied. Assume that c_k is in s-th place in row (37) i.e. $c_k = u_i^{j_s}$. Then
$a_k^{j_1} = a_k^{j_2} = \ldots = a_k^{j_s} = 1$ and $a_k^{j_s+1} = a_k^{j_s+2} = \ldots = a_k^{j_p} = 0$.
We obtain the following problem:

$$\sum_{i=1}^{q} c_i z_i \to \min \tag{38}$$

$$\sum_{i=1}^{q} a_i^j z_i \ge 1, \quad j = 1,2,\ldots,p \tag{39}$$

$$z_i = 0 \vee 1, \quad i = 1,\ldots, q. \tag{40}$$

To explain this transformation let us consider the following example:
Let matrix (u_i^j) be as follows :

$$(u_i^j) = \begin{pmatrix} 2 & 4 & 9 \\ 8 & 12 & 3 \\ 10 & 6 & 5 \end{pmatrix}.$$

Then $c = (2,\ 8,\ 10,\ 4,\ 12,\ 6,\ 9,\ 3,\ 5)$ and matrix (a_i^j) is : $\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$

Problem (38) – (40) is a Multiple-Choice Knapsack problem and since it is

NP-Complete [7], [8], problem (30) –(36) is also NP-Complete. So the following Theorem and Corollary hold :

Theorem 5. *The problem (30)-(36) is NP-Complete.*

Corollary 3. *The Subproblem (1)-(2) is NP-Complete.*

7. CONCLUSION

The Cutting Angle Method (CAM) solves a broad class of the Global Optimization problems. Its computational efficiency is significantly affected by the efficiency of solving the subproblem, which is solved at each iteration. We have proved that the Subprolem is NP-Complete, so is recommended to solve the Subproblem by means of effective heuristic algorithms, which leads to exact solution algorithms, thus increasing the overall efficiency of the CAM.

Acknowledgements: The author would like to express his gratitude to Dj. A. Babayev for bringing his attention to this problem and for very usefull discussions.

REFERENCES

[1] M.Yu. Andramónov, A. M.Rubinov and B. M.Glover, Cutting Angle methods for minimizing increasing convex along rays functions, *Research Report* 97/7, SITMS, University of Ballarat, 1997.

[2] M.Yu. Andramonov, A. M.Rubinov and B. M.Glover, Cutting Angle methods in Global Optimization, *Applied Mathematics Letters*, **12**, 95-100, 1999.

[3] A. M. Rubinov and M. Andramonov, Minimizing increasing star-shaped functions based on abstract convexity, *Journal of Global Optimization,* **15**, 19-39, 1999.

[4] A. M. Rubinov and M. Andramonov, Lipshitz programming via increasing convex along rays functions, *Optimization Methods and Software,* **10,** 763-781, 1999.

[5] A. M. Bagirov and A. M. Rubinov, Global minimization of increasing positively homogeneous functions over the unit simplex , *Annals of Operations Research*, **98,** 171 - 187, 2000.

[6] Dj. A. Babayev, An Exact Method for Solving the Subproblem of the Cutting Angle Method of Global Optimization, *In book "Optimization and Related Topics", in Kluwer Academic Publishers, series "Applied Optimization",* Vol. **47**, 412 –425, Dodrecht/ Boston/ London, December, 2000.

[7] R.M.Garey and D.S.Johnson, *Computers and Intractability. A Guide to the Theory of NP-Completeness.* Freeman, San Francisco, 1979.

[8] S.Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations.* John Wiley and Sons, Chichester, 1990.