Selçuk J. Appl. Math.
Vol. 12. No. 1. pp. 43-52, 2011

Selçuk Journal of Applied Mathematics

Analysis of Dominating Subset with Minimal Weight Problem

Burak Ordin, Fidan Nuriyeva

Department of Mathematics, Faculty of Science, Ege University, 35100 Bornova, Izmir, Turkiye
e-mail: burak.ordin@ege.edu.tr, nuriyevafidan@gmail.com

Received Date: December 11, 2009
Accepted Date: December 29, 2010

Abstract. The paper deals with the analysis of the combinatorial problem "dominating subset with minimal weight" problem that is equivalent to a subproblem in the cutting angle method which has been developed for solving a broad class of global optimization problems. An example is given for a more clear expression of the problem and its presentation is supported by simple graph notation. Then the complexity of the problem is discussed. It is proven that the problem is strongly NP-Complete by using the weighted set covering problem. Finally analysis of dominating subset with minimal weight problem is expressed for problems in small dimensions.

Key words: Combinatorial Optimization; NP-Complete in the strong sense; Dominating Subset with Minimal Weight Problem; Weighted Set Covering Problem; Set Covering Problem.
2000 Mathematics Subject Classification: 68Q25, 90C27, 65K05.

1. Introduction

The cutting angle method (CAM) for global optimization is introduced and studied in [1], [6], [8], [10]. It is an iterative method and in each iteration of CAM a subproblem has to be solved in turns, generally, a global optimization problem. Let (l_i^k) be an $m*n$ matrix, $m \geqslant n$, with m rows $k = 1, \ldots, m$, and n columns, $i = 1, \ldots, n$. All the elements of l_i^k are nonnegative. The first n rows of matrix (l_i^k) form a diagonal matrix, i.e., $l_i^k > 0$, only for $k = i$, $i = 1, ..., n$.

Let us introduce the function : $h(x) = \max\limits_k \min\limits_{i \in I(l^k)} l_i^k x_i$, where $I(l^k) = \{i : l_(i^k) > 0\}$.

The subproblem is formulated as follows:

$$h = \min_{x \in S} h\left(x\right)$$

subject to

$$x \in S = \{x | \sum_{i=1}^{n} x_i = 1, x_i \geqslant 0, i = 1, \ldots, n.\}$$

The solution of the subproblem is the crucial step in CAM and reformulation of the subproblem as dominating subset with minimal weight (DSMW) problem demonstrates that the approximation is very efficient for the solution ([6-8]).
In this work, DSMW problem is expressed in detail and an example is given for a more clear expression of the problem. Besides, it is presented as a mathematical model of DSMW problem and proved that this problem is NP-Complete in the strong sense. The presentation of the problem is supported by simple graph notation. Finally analysis of DSMW problem is expressed for problems in small dimensions.
The organization of the paper is as follows. In section 2, we give the definition of DSMW problem. It consists of an applied interpretation, an example and the mathematical model of the problem. In section 3 a simple graph notation has been developed for the presentation of the problem. Section 4 deals with the analysis of the problem.

2. A Brief Introduction to DSMW Problem

We will start this section with an interpretation of DSMW problem. Let (u_i^j) be a p * n matrix, with p rows, j = 1, 2, . . . , p and n columns, i = 1, 2, . . . , n and u_i^j are non-negative for all i, j .
The task is to choose some elements of the matrix such that:

(i) each row contains a chosen element, or contains some element which is less than some chosen element located in its column;

(ii) the sum of the chosen elements is minimal.

Let us give the following applied interpretation of this problem:
A task consisting of p (j = 1, 2, . . . , p) operations can be performed 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

$$u_i^{j_1} \leqslant u_i^{j_2} \leqslant ... \leqslant u_i^{jp} \tag{1}$$

for column i, then $u_i^{j_1}$ is the time (or cost) required for the accomplishment of operation j_1 by processor i; $u_i^{j_2}$is the time required 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 needed for the accomplishment of all operations ($j_1, j_2, \ . \ . \ . \ , j_p$) by processor i . The problem is to distribute operations among the processors minimizing the total time (or the total cost) required for the accomplishment of all tasks. Clearly, the problem is generalized as the assignment problem ([7]).

2.1. An Example of DSMW Problem

In the following example, the DSMW problem is expressed in a simple way.

Example 1. Suppose that matrix (u_i^j) for DSMW problem is given as in Figure 1.

$$(u_i^j) = \begin{pmatrix} 9 & 4 & 2 \\ 3 & 12 & 8 \\ 10 & 1 & 6 \\ 5 & 7 & 11 \end{pmatrix}$$

Figure 1. DSMW problem matrix

Here, $u_1^1 = 9$ is the time (or cost) required for the accomplishment of operations 1, 2, 4 by processor 1. $u_1^2 = 3$ is the time (or cost) required for the accomplishment of only operation 2 by processor 1. $u_1^3 = 10$ is the time (or cost) required for the accomplishment of operations 1,2,3,4 by processor 1. $u_1^4 = 5$ is the time (or cost) required for the accomplishment of operations 2 and 4 by processor 1. Likewise, $u_2^1 = 4$ is the time (or cost) required for the accomplishment of operations 1,3 by processor 2. $u_2^2 = 12$ is the time (or cost) required for the accomplishment of all operations by processor 2. $u_2^3 = 1$ is the time (or cost) required for the accomplishment of just operation 3 by processor 2. $u_2^4 = 7$ is the time (or cost) required for the accomplishment of operations 1, 3, 4 by processor 2. Lastly, $u_3^1 = 2$ is the time (or cost) required for the accomplishment of only operation 1by processor 3. $u_3^2 = 8$ is the time (or cost) required for the accomplishment of operations 1, 2, 3 by processor 3. $u_3^3 = 6$ is the time (or cost) required for the accomplishment of operations 1, 3, 4 by processor 3. $u_3^4 = 11$ is the time (or cost) required for the accomplishment of all operations by processor 3.
The problem is to distribute operations among the processors while minimizing the total time (or the total cost) required for the accomplishment of all tasks. A feasible solution could be given by choosing element u_1^3. Then, the value of the objective function is 10. Optimal solution of this problem is found by choosing elements u_3^1, u_2^3 and u_1^4 and the value of the objective function is that $u_3^1 + u_2^3 + u_1^4 = 2 + 1 + 5 = 8$.

2.2 Mathematical Model of the DSMW Problem

The mathematical model of the DSMW problem is constructed as follows. Let us define the function $Sg(x)$ and variables x_i^j:

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

$$x_i^j = \begin{cases} 1, & \text{if } u_i^j \text{ is chosen} \\ 0, & \text{otherwise} \end{cases} .$$

Then, the mathematical model of the DSMW problem is given as follows:

$$\min_{x_i^j} \sum_{i=1}^{n} \sum_{j=1}^{p} u_i^j x_i^j \tag{2}$$

$$\sum_{i=1}^{n} x_i^j \leqslant 1, j = 1, 2, \ldots, p, \tag{3}$$

$$\sum_{j=1}^{p} x_i^j \leqslant 1, i = 1, 2, \ldots, n, \tag{4}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{p} x_i^j \geqslant 1, \tag{5}$$

$$\sum_{i=1}^{n} y_i^j \geqslant 1 \quad , j = 1, 2, \ldots, p, \tag{6}$$

$$x_i^j = 0 \vee 1, i = 1, 2, \ldots, n; j = 1, 2, ...p. \tag{7}$$

$$y_i^j = Sg(\max_{k=1,p} \{u_i^k x_i^k\} - u_i^j),\ i = 1, 2, \ldots, n;\ j = 1, 2, \ldots, p, \tag{8}$$

According to [6], the following theorem was proved:

Theorem 1. The Subproblem in CAM and DSMW problem are equivalent.

Minimization of the subproblem in CAM is essentially a combinatorial problem, and thus grows exponentially if all possibilities are tested. DSMW problem reduces the complexity of the crucial step in subproblem of cutting angle method. It is a new approach for solving the subproblem in the CAM by using theorem 1.

3. The Notation of DSMW Problem by Graphs

DSMW problem can be thought as a minimal matching in a bipartite graph (Figure 3.1). Suppose that, the vertices W=(w_1, w_2, ..., w_n) in the first part of the graph G = (W, V, E) express processors and the vertices V=(v_1, v_2, ..., v_p) in the second part of G express the tasks. Each edge $e_i^j \in E$ (the edge e_i^j=(w_i, v_j)) corresponds to the element u_i^jin the matrix (u_i^j). When the value u_i^j is defined as the weight of edge e_i^j, it is chosen as an acceptable matching such that its total weight is minimum and it covers all vertices of V. Among the outgoing edges of each point in W, it is called the dominant element which is greater than the others. When any dominant edge is chosen, it means that other elements which are not dominant are chosen along with the dominant element. (A dominant edge is denoted by a line and a non dominant edge is denoted by a broken line in Figure 3). But, in this case, we take only the weight of dominant edge. If some elements in set V are end points of the chosen edge then it means that those vertices are covered.

Example 2. The problem in example 1 is denoted as follow.

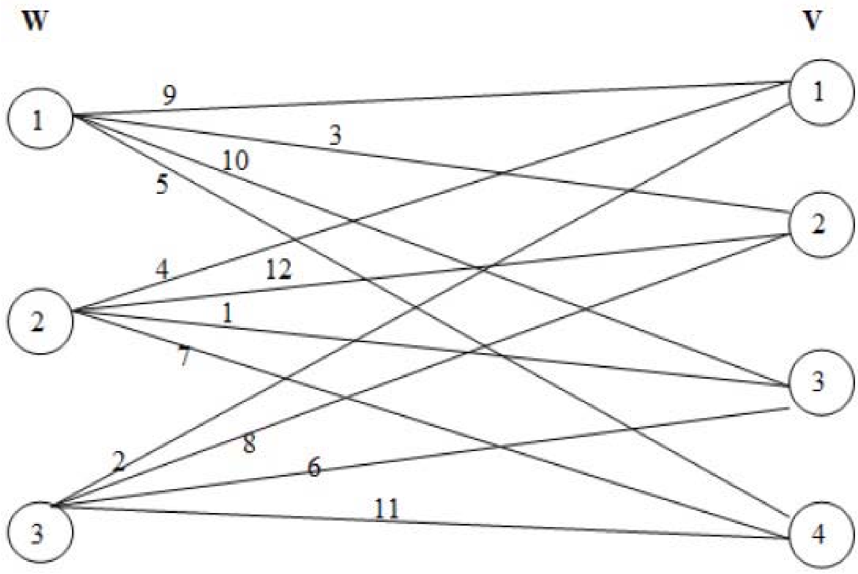

Figure 2. Graph notation of DSMW problem

There is a bipartite graph that consists of two parts for the matrix (u_i^j) as in Figure 2, where W={1, 2, 3} and V={1, 2, 3, 4}.
$u_1^1 = 9$ is the dominant of elements $u_1^2 = 3$ and $u_1^4 = 5$ in Figure 1, since $u_1^1 > u_1^2$ and $u_1^1 > u_1^4$. But $u_1^3 > u_1^1$, and so $u_1^3 = 10$ is the dominant of the element $u_1^1 = 9$.
$u_1 = u_1^3 = 10$ is the heaviest element of first column, so that, it covers all rows of that column. That is, it is a dominant subset. While $x_1^3 = 1$, the others are equal to 0, so there is an appropriate solution.
Same operations are valid for the heaviest elements of the other columns.
The set $\{u_3^1 = 2,\ u_2^3 = 1,\ u_1^4 = 5\}$ is a dominating subset with minimal weight for this example. In the second row, the element $u_1^2 = 3$ is not chosen and is covered by $u_1^4 = 5$, because u_1^4 is the dominant of u_1^2, so u_1^4 is chosen.
Consequently, $x^* = \{x_3^1 = x_2^3 = x_1^4 = 1, x_1^1 = x_1^2 = x_1^3 = x_2^1 = x_2^2 = x_2^4 = x_3^2 = x_3^3 = x_3^4 = 0\}$ is an optimal solution and the value of the objective function is 8.
Appropriate edges chosen in an optimal solution are denoted in Figure 2. These are the edges $e_3^1 = (3, 1)$, $e_2^3 = (2, 3)$ and $e_1^4 = (1, 4)$. The edge $e_1^2 = (1, 2)$ is denoted by a broken line, because the edge $e_1^4 = (1, 4)$, which is an outgoing edge from the same vertex with e_1^2, has greater weight $(u_1^4 > u_1^2)$. Only the weight of the chosen matching is taken ($u_1^2 = 3$ is not taken) and then all vertices of set V are covered.

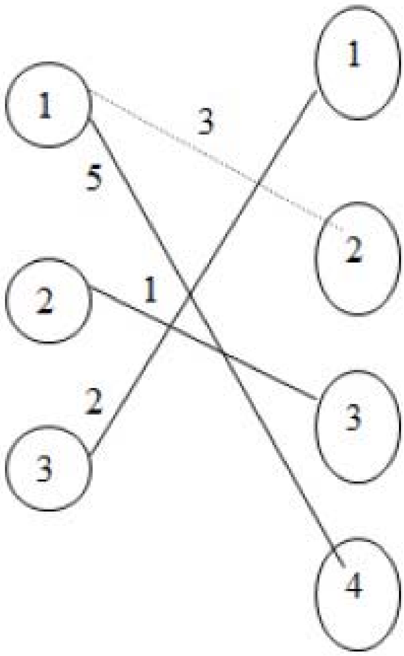

Figure 3. Graph notation of the optimal solution of the problem

4. Analysis of DSMW Problem

In this section, we prove that DSMW problem is NP-Complete in the strong sense by using weighted set covering problem. Weighted set covering problem could be defined as follow.

4.1. Weighted Set Covering Problem (WSCP)

At first, let us introduce the Set-Covering problem. An instance (X,F) of the Set-Covering problem consists of a finite set X and a family F of subsets of X such that every element of X belongs to at least one subset in F:

$$X = \bigcup_{S \in F} S$$

We say that a subset S∈F covers its elements. The problem is to find a maximum size subset C⊆F whose members cover all of X:

$$X = \bigcup_{S \in C} S$$

We say that any C satisfying above equation covers X.
It is known that this problem is NP-complete in the strong sense [8].

Theorem 2. The vertex cover problem is polynomially transformable to the set-covering problem. Therefore, the set-covering problem is NP-Complete.

In [2], the WSCP is interpreted as follows: The data of the problem consists of finite sets $P_1, P_2, ..., P_n$ and positive numbers $c_1, c_2, ..., c_n$. We denote $\bigcup(P_j : 1 \leqslant j \leqslant n)$ by I and write $I=\{1,2,...,m\}$, $J=\{1,2,...,n\}$. A subset $J^* \subseteq J$ is called a cover if $\bigcup(P_j : j \in J^*) = I$; the cost of this cover is $\sum(c_j : j \in J^*)$. The problem is to find a cover with minimum cost. This problem could be thought such that:
Let us define an mxn matrix $A = (a_i^j)$ where $a_i^j = \begin{cases} 1, & \text{if } i \in P_j \\ 0, & \text{otherwise} \end{cases}$. So, n columns of A are the incidence vectors of $P_1, P_2, ..., P_n$. Clearly, the incidence vector $x = (x_j)$ of an arbitrary cover satisfies :

$$\sum_{j=1}^{n} a_{ij} x_j \geqslant 1 \textit{for all i,}$$

$$x_i \geqslant 0 \textit{for all j.}$$

Let c^T be a positive row vector of length n and let e be the column vector whose all m components are ones. The set covering problem is to minimize $c^T x$ subject to $Ax \geqslant e$ and $x = 0 \vee 1$.
Let us take a WSCP instance with $pxn=q$ variables and p conditions that is equal to DSMW problem. The coefficients of the objective function in this problem are determined as follows:

$c_1 = u_1^1$, $c_2 = u_1^2$,..., $c_p = u_1^p$, $c_{p+1} = u_2^1$, $c_{p+2} = u_2^2$,..., $c_{2p} = u_2^p$, $c_{2p+1} = u_3^1$,..., $c_q = u_n^p$.

In other words, $c_t = u_i^j$ and t = p.(i-1) .j, 1$\leqslant$ $k \leqslant q$,substitutions are performed.
Suppose that $c_k = u_i^j$ for some i, j; namely, c_k is an element of i^{th} column and j^{th} row of matrix (u_i^j) and it satisfies condition (1) for column i. Besides, let $c_k$be the s^{th} element of the sequence in respect of (1). Then,
$a_k^{j_1} = a_k^{j_2} = ... = a_k^{j_s} = 1$ and $a_k^{j_s+1} = a_k^{j_s+2} = ... = a_k^{j_p} = 0$.
So, there are ones as the dominance degree of the element $u_i^{j_s} = c_k$ that is in column k of matrix (a_k^j) (according to (1), it corresponds to first s elements) and zeros for $n-s$ elements (according to (1), it corresponds to last $n-s$ elements). We determine the elements of the matrix (a_k^j) as follows,

$$a_k^j = \begin{cases} 1, & \text{if} \quad u_k^j \leq u_k^{j_i} \\ 0, & \text{if} \quad u_k^j > u_k^{j_i} \end{cases}$$

and the problem is formulated as,

$$\min \sum_{i=1}^{q} c_i z_i \tag{9}$$

$$\sum_{i=1}^{q} a_i^j z_i \geqslant 1 \quad , j = 1, 2, ..., p \tag{10}$$

$$z_i = 0 \vee 1, i = 1, ..., q. \tag{11}$$

.
The problem (9)-(11) is a WSCP where each column of the matrix (a_i^j) corresponds to some P_j in [2]. Here *t=p*(i-1)+j*.

It is clear that, all operations in this transformation are bounded by $O(p^3 n)$. Since we determine the number $q = p * n$ for elements $c_k = u_i^j$, and the number $p * q = p^2 * n$ for elements a_i^j, so, $O(p)$ operation is performed.

4.2. Some Properties of DSMW Problem

Let us take a decision version of DSMW problem, D(DSMW). Each I∈D(DSMW) instance is determined by a given matrix (u_i^j) and a real number k. The aim is to find if there exist x_i^j, y_i^j such that the value of the objective function does not exceed k ? The following functions could be used as Length and Max functions for D(DSMW) problem.

$$Length[I] = \left|(u_i^j)\right| = p * n$$

$$Max[I] = \max\{u_i^j \,|i = 1, ..., n; j = 1, ..., p\}$$

Now, let us give some Lemmas about *D(DSMW)*.

Lemma 1. *D(DSMW)* problem is a number problem.

Proof: In DSMW problem, there aren't any bounds for numbers u_i^j. Therefore, there is no polynomial P , such that $Max[I] \leqslant P(Length[I])$, ∀I∈D(DSMW). Then, *D(DSMW)* is a number problem. *o*

Lemma 2. *D(DSMW)*∈ *NP*

Proof: It is clear that, the answer in any solution is either yes or no and for checking this answer, $O(p)$ operation is performed. A chosen cover set consists of at most p elements, furthermore the number of the operations done for checking whether it covers the matrix or not, is bounded by $O(p)$. To compute its weight and compare with k, $O(p)$ operation is required. Consequently, the number of the total operation is bounded by $O(p)$. Namely, *D(DSMW)* ∈ *NP*.

Theorem 3. *D(DSMW)* problem is *NP-Complete in the strong sense.*

Proof: As seen in section 4.1, WSCP is polynomially transformable to DSMW problem. Besides, Set Covering Problem (SCP), which is a special type of WSCP where $c_1, c_2, ..., c_n = 1$, is *NP-Complete* and it is not a number problem. So, SCP is NP-Complete in the strong sense ([4], [9]). For any Π ∈ *D(DSMW)* problem and any polynomial P, let Π_p denote the subproblem of Π obtained by $\Pi_P = \{I \in \Pi \,|Max[I] \leqslant P(Length[I])\}$, then Π_p is not a number problem. So, according to Lemma 1 and Lemma 2, *D(DSMW)* problem is *NP-Complete in the strong sense.*
Analysis of DSMW problem in small dimensions could be given as follows.
DSMW problem (where u^* is the optimal solution, $p = 1, 2$ and $n = 1, 2$) is solved in polynomial time of its dimensions.
For $p = 1$, $u^* = \min_{i=1,n}\{u_i^1\}$,clearly, $O(n)$ operation is performed.
For $p = 2$, $u^* = \min\{\min_{i=1,n}\{u_i^1, u_i^2\}, (\min_{i=1,n}\{u_i^1\} + \min_{i=1,n}\{u_i^2\})\}$

It is obvious that the number of the operations should be *O(2n)+O(n)+O(n)=O(n).*
For $n = 1$, $u^* = \max_{j=\overline{1,p}}\{u_1^i\}$, the number of the operations are $O(p)$. For $n = 2$,
$u^* = \min\{\min\{\max_{j=1,p}\{u_i^1\},$

$\max_{j=1,p}\{u_2^i\}, \min\{\min_{j=1,p}\{u_1^j + \max\{u_2^j|j \in P - V(u_1^j)\}\}, \min_{j=1,p}\{u_2^j + \max\{u_1^j|j \in P - V(u_2^j)\}\}\}\}$.
Here, $2.O(p)$ operation is performed to determine $\min\left\{\max_{j=\overline{1,p}} u_1^i\right\}, \max_{j=\overline{1,p}}\{u_2^i\}\}$,

$O(p^2)$ operation is performed for determining $\min\limits_{j=\overline{1,p}}\{u_1^j+\max\{u_2^j|j \in P-V(u_1^j)\}\}$.
and $\min\limits_{j=\overline{1,p}}\{u_2^j + \max\{u_1^j \; |j \in P - V(u_2^j)\}\}$.
Consequently, the number of the total operations are 2.O(p)+2O(p^2) $= O(p^2)$ for $n = 2$.

5. Conclusion

In this paper, the analysis of a combinatorial problem that is equal to a subproblem in the cutting angle method which has been developed for solving a broad class of global optimization problems is expressed. The presentation of the problem is supported by simple graph notation. It is proved that the problem is NP-Complete in the strong sense by using weighted set covering problem. For small values (n=1, 2 and p=1, 2), the problem could be solved in polynomial time analytically.

References

1. Dj. Babayev, A.(2000): 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, December, Dordrecht/Boston/ London, 472pp.
2. Chvatal, V.(1979): A Greedy Heuristic for the Set Covering Problem, Mathematics of Operations Research, 4(3):233-235.
3. Garey, R.M. and Johnson, D.S.(1979): Computers and Intractability. A Guide to the Theory of NP-Completeness., San Francisco, Freeman.
4. Johnson, D.S.(1974): Approximation Algorithms for Combinatorial Problems, Journal of Computer and System Sciences, 9:256-278.
5. Lovasz, L.(1975): On the ratio of optimal integral and fractional covers, Discrete Mathematics, Vol. 13, pp. 383-390.
6. Nuriyev, U.G.(2005): An Approach to the Subproblem of the Cutting Angle Method of Global Optimization, Journal of Global Optimization, 31, 353-370.
7. Nuriyev, U.G., Ordin B.(2004): Computing Near-Optimal Solutions For The Dominating Subset with Minimal Weight Problem, International Journal Of Computer Mathematics, Vol 81, No 11, 1309-1318.
8. Ordin, B.(2009): “The Modified Cutting Angle Method for global minimization of increasing positively homogenius functions over the unit simplex”, Journal of Industrial and Management Optimization (JIMO), Vol 5 (4), 825-834.
9. Papadimitriou, C.H., Steiglitz, K.(1982): Combinatorial Optimization:Algorithms and Complexity, Prentice-Hall.
10. Rubinov, A.M.(2000): Abstract Convexity and Global Optimization, Kluwer Academic Pub, Dordrecht.