Università di Pisa
Dipartimento di Statistica e Matematica
Applicata all'Economia

Report n.301

A branch and bound approach for a class of d.c. programs

Riccardo Cambini and Claudio Sodini

Pisa, gennaio 2008
- Stampato in proprio –

Via Cosimo Ridolfi, 10 - 56124 PISA - Tel. Segr. Amm. 050 2216231 Segr. Stud. 050 2216317 Fax 050 2216375
Cod. Fisc. 80003670504 - P. IVA 00286820501 - Web http://statmat.ec.unipi.it- E-mail: dipstat@ec.unipi.it

A branch and bound approach for a class of d.c. programs

Riccardo Cambini and Claudio Sodini

Department of Statistics and Applied Mathematics
Faculty of Economics, University of Pisa
Via Cosimo Ridolfi 10, 56124 Pisa, ITALY
e-mail: cambric@ec.unipi.it, csodini@ec.unipi.it

Abstract

The aim of this paper is to propose a branch and bound method for solving a class of d.c. programs. In this method the relaxations are obtained by linearizing the concave part of the objective function. The branch and bound solution method has been implemented and analyzed by means of a deep computational test.

Key words: d.c. programming, branch and bound, quadratic programming, multiplicative programming
AMS - 2000 Math. Subj. Class. 90C20, 90C26, 90C30.
JEL - 1999 Class. Syst. C61, C63.

1 Introduction

In this paper we propose a branch and bound method for solving a wide class of d.c. programs. This method allows to solve various kinds of problems, such as nonconvex quadratic problems [2, 3, 6, 7, 12, 14, 16, 17, 22], multiplicative problems [7, 13, 14, 15, 18, 19], some more general d.c. problems [1, 4, 7, 8, 10, 11, 12, 14, 20, 21, 22], and problems which can be rewritten in the considered d.c. form by means of an increasing transformation. In other words, the method proposed in this paper allows to solve in a unifying framework, that is to say with the same solution algorithm, various kinds of problems which have been studied and solved in the literature in different ways (see for example [1, 2, 3, 5, 9, 10, 14, 19]). It is also worth pointing out the relevance of the considered class of problems in real applications, see for example [7, 9, 14, 18, 21].

In Section 2 the considered class of d.c. problems is defined and some kinds of problems belonging to the class are provided, in order to witness the wideness of the class and its usefulness from an applicative point of view. In Section 3 the branch and bound method is described while in Section 4 the results of a deep computational test are provided.

2 Statement of the problem

In this paper we aim to study and to propose a solution method for the following class of d.c. problems having a polyhedral feasible region.

Definition 2.1 We define the following d.c. program:

$$P: \begin{cases} \min f(x) = c(x) - \sum_{i=1}^{k} g_i(d_i^T x) \\ x \in X \subset \Re^n \end{cases}$$

where X is a polyhedron which can be given by inequality constraints $Ax \leq b$ and/or box constraints $l \leq x \leq u$ and/or equality constraints $Mx = q$, where $A \in \Re^{m \times n}$, $b \in \Re^m$, $l, u \in \Re^n$, $M \in \Re^{h \times n}$, $q \in \Re^h$. Functions $c : \Re^n \to \Re$ and $g_i : \Re \to \Re$, $i = 1, \dots, k$, are assumed to be convex and continuous, while vectors $d_i \in \Re^n$, $i = 1, \dots, k$, verify the following property:

$$\exists \alpha, \beta \in \Re^k \text{ such that } \alpha_i \leq d_i^T x \leq \beta_i \ \forall x \in X, \ i = 1, \dots, k. \qquad (1)$$

For the sake of convenience, we also define the matrix $D = [d_1, \dots, d_k] \in \Re^{n \times k}$ whose i-th column is vector d_i, $i = 1, \dots, k$.

Notice that condition (1) does not imply the compactness of the feasible region X; notice also that no differentiability hypothesis is assumed for functions f, c and g_i, $i = 1, \dots, k$.

In order to point out the wideness and the usefulness of the class of problems P, it is worth pointing out the following particular cases.

Production models

Usually, in production models a convex cost function has to be minimized. In the case some profits could be gained by selling the overproduction, a concave function of the kind $-\sum_{i=1}^{k} \lambda_i \max\{0, d_i^T x + d_i^0\}$, where λ_i is the selling price for the i-th good, has to be added in the objective function. As a consequence the following function is obtained:

$$f(x) = c(x) - \sum_{i=1}^{k} \lambda_i \max\{0, d_i^T x + d_i^0\}$$

This function verifies the assumptions of P just considering the convex functions $g_i(y) = \lambda_i \max\{0, y + d_i^0\}$, $i = 1, \dots, k$.

Nonconvex quadratic problems

In [2, 3] it is shown how an indefinite or concave quadratic function $f(x) = \frac{1}{2}x^TQx + q^Tx$ can be decomposed in D.C. form.

- By means of a Lagrange decomposition or by means of eigenvectors (which determine the vectors $d_i \in \Re^n$, $i = 1, \dots, k$), it is possible to decompose $f(x) = \frac{1}{2}x^TQx + q^Tx$ in the form:

$$f(x) = (\frac{1}{2}x^T\overline{Q}x + q^Tx) - \sum_{i=1}^{k} \frac{1}{2}(d_i^Tx)^2$$

  where $\overline{Q}$ is positive semidefinite and k is equal to the number of negative eigenvalues of Q. The assumptions of P are fulfilled by considering the convex functions $c(x) = (\frac{1}{2}x^T\overline{Q}x + q^Tx)$ and $g_i(y) = \frac{1}{2}y^2$.

- By means of a diagonal decomposition (which determines the real values $\lambda_i \geq 0$, $i = 1, \dots, n$), it is possible to decompose $f(x) = \frac{1}{2}x^TQx + q^Tx$ in the form:

$$f(x) = (\frac{1}{2}x^T\overline{Q}x + q^Tx) - \sum_{i=1}^{n} \frac{1}{2}\lambda_i x_i^2$$

  where $\overline{Q}$ is positive semidefinite. The assumptions of P are fulfilled by considering the convex functions $c(x) = (\frac{1}{2}x^T\overline{Q}x + q^Tx)$ and $g_i(y) = \frac{1}{2}\lambda_i y^2$ and by assuming $d_1, \dots, d_n$ be the canonical vectors of $\Re^n$.

As a particular case, the following class of linear multiplicative functions can be considered:

$$f(x) = \sum_{i=1}^{r}(v_i^Tx + v_i^0)(d_i^Tx + d_i^0)$$

where $d_i, v_i \in \Re^n$ and $d_i^0, v_i^0 \in \Re$ for all $i = 1, \dots, r$. This function can be easily transformed in the form $f(x) = \frac{1}{2}x^TQx + q^Tx + q_0$, with Q simmetric $n \times n$ real matrix, by defining:

$$Q = \sum_{i=1}^{r}(v_i d_i^T + d_i v_i^T) \quad , \quad q = \sum_{i=1}^{r}(d_i^0 v_i + v_i^0 d_i) \quad , \quad q_0 = \sum_{i=1}^{r} d_i^0 v_i^0$$

so that it can be decomposed in D.C. form as it has been previously described. For the sake of completeness, notice that in [15] this class of linear multiplicative functions has been studied and solved with a different branch and bound approach.

Class of multiplicative problems
The following class of multiplicative functions can be converted in the form of P by means of a strictly monotone logarithmic transformation:

$$f(x) = e^{s(x)} \frac{\prod_{i=1}^{k} h_i\left(d_i^T x + d_i^0\right)}{\prod_{j=1}^{r} v_j(x)};$$

where the functions $h_i : \Re \to \Re_{++}$, $i = 1, \dots, k$, and the functions $v_j : \Re^n \to \Re_{++}$, $j = 1, \dots, r$ are assumed to be concave and positive, while the function $s : \Re^n \to \Re$ is required to be convex.

The strict monotonicity of the logarithmic function allows to study the following transformed function in place of $f(x)$:

$$\overline{f}(x) = log(f(x)) = s(x) - \sum_{j=1}^{r} log(v_j(x)) - \sum_{i=1}^{k} -log\left(h_i\left(d_i^T x + d_i^0\right)\right)$$

This function verifies the assumptions of P just considering the convex functions $c(x) = s(x) - \sum_{j=1}^{r} log(v_j(x))$ and $g_i(y) = -log\left(h_i\left(y + d_i^0\right)\right)$.

Low rank nonconvex structures
In the literature (see for all [14]), problems with objective function of the kind

$$f(x) = c(x) - \sum_{i=1}^{k} \lambda_i (d_i^T x + d_i^0)^2$$

have been solved by means of branch and cut or branch and select techniques. This function verifies the assumptions of P just considering the convex functions $g_i(y) = \lambda_i (y + d_i^0)^2$.

Some more examples
The wideness of the class of problems P is witnessed by the following further examples fulfilling the assumptions of function $f(x)$. With this aim, recall that the composition of a nondecreasing convex function by another convex function is convex too. In the following examples function $c(x)$ is assumed to be convex and the parameters $\lambda_i \in \Re$ are supposed to be positive.

- $f(x) = c(x) - \sum_{i=1}^{k} \frac{\lambda_i}{(d_i^T x + d_i^0)}$, where $g_i(y) = \lambda_i / (y + d_i^0)$ and the linear functions $d_i^T x + d_i^0$ are positive on the feasible region;
- $f(x) = c(x) - \sum_{i=1}^{k} \lambda_i e^{d_i^T x}$, where $g_i(y) = \lambda_i e^y$;
- $f(x) = c(x) + \sum_{i=1}^{k} \lambda_i log(d_i^T x + d_i^0)$, where $g_i(y) = -\lambda_i log(y + d_i^0)$ and the linear functions $d_i^T x + d_i^0$ are positive on the feasible region;

- $f(x) = c(x) + \sum_{i=1}^{k} \lambda_i \sqrt{d_i^T x + d_i^0}$, where $g_i(y) = -\lambda_i \sqrt{y + d_i^0}$ and the linear functions $d_i^T x + d_i^0$ are nonnegative on the feasible region;
- $f(x) = c(x) - \sum_{i=1}^{k} \lambda_i \left| d_i^T x + d_i^0 \right|$, where $g_i(y) = \lambda_i \left| y + d_i^0 \right|$;

Notice that any positive linear combination of these functions verifies the assumptions of P.

3 A branch and bound approach

The aim of this paper is to propose a branch and bound method for solving problem P and to provide detailed results of a computational experience. The approach is based on the relaxation of the objective function obtained by linearizing its concave part $-\sum_{i=1}^{k} g_i(d_i^T x)$ with respect to the functions $d_i^T x$, $i = 1, \ldots, k$. These relaxations provide, in the various iterations of the branch and bound process, convex subproblems with polyhedral feasible region which can be solved by means of any of the known algorithms looking for an optimal local solution. The used branching criterion splits the feasible region of the current subproblem with respect to one of the functions $d_i^T x$, $i \in \{1, \ldots, k\}$.

3.1 Main properties

Given a pair of vectors $\alpha, \beta \in \Re^k$, with $\alpha \leq \beta$, we can denote with $B(\alpha, \beta)$ the following set

$$B(\alpha, \beta) = \{x \in \Re^n : \ \alpha \underline{\leq} D^T x \underline{\leq} \beta\}$$

The concave part $-\sum_{i=1}^{k} g_i(d_i^T x)$ of function $f(x)$ can be linearized over $B(\alpha, \beta)$ as follows:

$$f_B(x) = c(x) - \sum_{i=1}^{k} [\mu_i (d_i^T x - \alpha_i) + g_i(\alpha_i)] = c(x) - \mu^T (D^T x - \alpha) - \sum_{i=1}^{k} g_i(\alpha_i)$$

where for all $i = 1, \ldots, k$ it is:

$$\mu_i = \begin{cases} \frac{g_i(\beta_i) - g_i(\alpha_i)}{\beta_i - \alpha_i} & if \ \alpha_i < \beta_i \\ 0 & if \ \alpha_i = \beta_i \end{cases}$$

Notice that $f_B(x)$ is an underestimation function for $f(x)$ over the set $B(\alpha, \beta)$, so that the following relaxed convex subproblem can be defined and used in the branch and bound scheme:

$$P_B(\alpha, \beta) : \begin{cases} \min f_B(x) \\ x \in X \cap B(\alpha, \beta) \end{cases}$$

The following result provides an estimation of the error done by solving the relaxed problem. With this aim the next functions will be used:

$$\begin{aligned} Err_B(x,i) &= \mu_i(d_i^T x - \alpha_i) - (g_i(d_i^T x) - g_i(\alpha_i)) \\ Err_B(x) &= f(x) - f_B(x) = \sum_{i=1}^{k} Err_B(x,i) \\ &= \mu^T(D^T x - \alpha) - \sum_{i=1}^{k} \left[g_i(d_i^T x) - g_i(\alpha_i) \right] \end{aligned}$$

Theorem 3.1 *Let us consider problems P and $P_B(\alpha,\beta)$ and let*

$$x^* = \arg \min_{x \in X \cap B(\alpha,\beta)} \{f(x)\} \quad \textit{and} \quad \overline{x} = \arg \min_{x \in X \cap B(\alpha,\beta)} \{f_B(x)\} \ .$$

Then, $f_B(\overline{x}) \leq f(x^*) \leq f(\overline{x})$, that is to say that

$$0 \leq f(x^*) - f_B(\overline{x}) \leq Err_B(\overline{x}).$$

Proof By means of the given definitions it is:

$$f(x^*) \leq f(\overline{x}) \quad \text{and} \quad f_B(\overline{x}) \leq f_B(x^*).$$

Noticing that $f_B(x^*) \leq f(x^*)$ it then follows:

$$f_B(\overline{x}) \leq f_B(x^*) \leq f(x^*) \leq f(\overline{x}).$$

The whole result is then proved being $Err_B(\overline{x}) = f(\overline{x}) - f_B(\overline{x})$. □

3.2 Branch and bound scheme

First notice that $2k$ linear programs are needed to determine, for $i = 1, \ldots, k$, the following values:

$$\tilde{\alpha}_i = \min_{x \in X}\{d_i^T x\} \quad \text{and} \quad \tilde{\beta}_i = \max_{x \in X}\{d_i^T x\}$$

As usual in branch and bound methods, a tolerance parameter $\epsilon > 0$ is needed in order to guarantee the numerical convergence of the algorithm. Furthermore, another step parameter $\delta > 0$ will be used within the branching criterion in order to estimate the local decreaseness of the objective function. The following branch and bound scheme can then be given.

Procedure Solve(P)
determine $\tilde{\alpha}$ and $\tilde{\beta}$;
fix the parameters $\epsilon > 0$ and $\delta > 0$;
initialize the global variables $x_{opt} := []$ and $UB := +\infty$;
Explore($\tilde{\alpha}, \tilde{\beta}$);
x_{opt} is an optimal solution and UB is its value;
end proc.

The core of the algorithm is the following recursive procedure "Explore()", which is based on a generalization of the so called "rectangular partitioning method" (see [5, 22]). Notice that this choice guarantees the convergence of the method, as it has been shown in [5].

Procedure **Explore**(α, β)
if $X \cap B(\alpha,\beta) \neq \emptyset$ *then*
Let $\overline{x}$ be the optimal solution of $P_B(\alpha,\beta)$;
if $f(\overline{x}) < UB$ *then* $UB := f(\overline{x})$; $x_{opt} := \overline{x}$ *end if*;
if $f_B(\overline{x}) < UB$ *and* $Err_B(\overline{x}) > \epsilon$ *then*
let $i = \arg\max_{j\in\{1,\dots,k\}}\{Err_B(\overline{x},j)\}$;
define $\alpha' := \alpha$, $\alpha'' := \alpha$, $\beta' := \beta$, $\beta'' := \beta$;
set $\beta'_i := d_i^T\overline{x}$, $\alpha''_i := d_i^T\overline{x}$;
if $f(\overline{x} + \delta d_i) > f(\overline{x})$ *then*
Explore(α', β');
Explore(α'', β'');
else
Explore(α'', β'');
Explore(α', β');
end if;
end if;
end if;
end proc.

Problem $P_B(\alpha,\beta)$ can be solved by any of the known algorithms for convex programs, that is any algorithm which finds an optimal local solution of a constrained problem. Clearly, if $c(x)$ is a quadratic [linear] function then $P_B(\alpha,\beta)$ is a quadratic [linear] problem. In other words, the easiest is the convex function $c(x)$ the more efficient can be the chosen solution algorithm for $P_B(\alpha,\beta)$.

Procedure "Explore()" opens no more branches in the case $f_B(\overline{x}) \geq UB$ (and hence, for Theorem 3.1, there is no possibility to improve the current solution) or $Err_B(\overline{x}) \leq \epsilon$ (and hence the error done in relaxing function $f(x)$ is sufficiently small). The visit criterion $f(\overline{x} + \delta d_i) > f(\overline{x})$ means that we aim to solve firstly the subproblem where the function $f(x)$ restricted along the direction d_i is locally decreasing.

In order to decrease as fast as possible the error $Err_B(\overline{x})$, the branch operation is done for the index i such that $i = \arg\max_{j\in\{1,\dots,k\}}\{Err_B(\overline{x},j)\}$. Notice also that condition $Err_B(\overline{x}) > \epsilon > 0$ implies that for any index i such that $i = \arg\max_{j\in\{1,\dots,k\}}\{Err_B(\overline{x},j)\}$ it results $Err_B(\overline{x},i) > 0$, so that $d_i^T\overline{x} \neq \alpha_i$ and $d_i^T\overline{x} \neq \beta_i$.

4 Computational results

The previously described branch and bound method has been fully implemented with the software MatLab 7.5 R2007b and tested on a computer having 2 Gb RAM and two Xeon dual core processors at 2.66 GHz.

In each category various problems have been randomly created and solved; in particular, the needed matrices and vectors have been generated with components in the interval [-10,10] by using the "rand()" MatLab function (numbers generated with uniform distribution). The subproblems $P_B(\alpha,\beta)$, depending on the structure of their objective function, are solved by means of the "linprog()", "quadprog()", or "fmincon()" MatLab functions.

The average number of iterations and the CPU times spent by the algorithm to solve the problems are given as the result of the test.

For the sake of simplicity, the test have been done with respect to the following type of problems:

$$\begin{cases} \min f(x) = \left(\frac{1}{2}x^TQx + q^Tx\right) - \sum_{i=1}^{k} g_i(d_i^Tx) \\ x \in X = \{x \in \Re^n : Ax \leq b, l \leq x \leq u\} \end{cases}$$

where $Q \in \Re^{n\times n}$ is a positive definite matrix, $q \in \Re^n$, $A \in \Re^{m\times n}$, $b \in \Re^m$, $l,u \in \Re^n$, and where we used $m = n$. The following particular objective functions, where $\lambda, d_0 \in \Re^k$, have been studied with respect to the same feasible regions:

- $g_i(y_i) = \lambda_i(y_i + d_i^0)^2$, $i = 1,\dots,k$;
- $g_i(y_i) = \lambda_i(y_i + d_i^0)^4$, $i = 1,\dots,k$;
- $g_i(y_i) = \lambda_i|y_i + d_i^0|$, $i = 1,\dots,k$;

The obtained results are summarized in Table 1, Table 2 and Table 3, respectively, where the cases $k = 2$, $k = 5$ and $k = 8$ have been considered and where the column *num* provides the number of randomly generated solved problems.

n	num	Iterations			CPU Time (secs)		
		k=2	k=5	k=8	k=2	k=5	k=8
5	4000	6.4615	62.569	439.05	0.086885	0.42167	2.4645
10	2000	11.945	177.22	1650.5	0.16365	1.858	17.354
15	1000	16.212	314.53	3475.7	0.33219	5.4048	60.426
20	500	18.844	442.86	5446.4	2.4682	49.486	612.52

Table 1: $g_i(y_i) = \lambda_i(y_i + d_i^0)^2$

n	num	Iterations			CPU Time (secs)		
		k=2	k=5	k=8	k=2	k=5	k=8
5	4000	5.0095	47.141	333.34	0.042375	0.28855	1.8921
10	2000	8.492	113.44	1028.8	0.1121	1.2045	10.824
15	1000	11.132	187.48	1890.9	0.23052	3.2243	32.517
20	500	12.624	266.4	2947.6	1.679	29.48	325.79

Table 2: $g_i(y_i) = \lambda_i(y_i + d_i^0)^4$

The obtained results confirm that the performance of the branch and bound method decreases as the number of the variables n and the number k of functions g_i increase.

Notice that the wideness of the class of programs covered by problem P does not allow a direct comparison of the proposed approach with others appeared in the literature.

5 Conclusions

In this paper we pointed out that a wide range of problems (multiplicative problems, d.c. problems, nonconvex quadratic problems, and so on) can be solved in a unifying framework by means of the same branch and bound method. The method has been concretely implemented and tested from a computational point of view, showing good performances in the case of problems having in the objective function a small number of functions g_i, $i = 1, \ldots, k$.

References

[1] Cambini R. and C. Sodini (2002), "A finite algorithm for particular d.c. quadratic programming problem", *Annals of Operations Research*,

n	num	Iterations			CPU Time (secs)		
		k=2	k=5	k=8	k=2	k=5	k=8
5	4000	6.493	35.068	172.76	0.044085	0.20336	0.94829
10	2000	9.146	63.614	326.21	0.091105	0.52905	2.6543
15	1000	10.666	90.66	560.04	0.16268	1.1374	6.7816
20	500	11.512	106.19	741.78	1.1027	8.3436	55.793

Table 3: $g_i(y_i) = \lambda_i |y_i + d_i^0|$

vol.117, pp.33-49.

[2] Cambini R. and C. Sodini (2005), "Decomposition methods for solving nonconvex quadratic programs via branch and bound", *Journal of Global Optimization*, vol.33, n.3, pp.313-336.

[3] Cambini R. and C. Sodini (2006), "A computational comparison of some branch and bound methods for indefinite quadratic programs", Report n.288, Dept. of Statistics and Applied Mathematics, University of Pisa.

[4] Churilov L. and M. Sniedovich (1999), "A concave composite programming perspective on D.C. programming", in *Progress in Optimization*, edited by A. Eberhard, R. Hill, D. Ralph and B.M. Glover, *Applied Optimization*, vol.30, Kluwer Academic Publishers, Dordrecht.

[5] Falk J.E. and R.M. Soland (1969), "An algorithm for separable nonconvex programming problems", *Management Science*, vol.15, no.9, pp.550-569.

[6] Floudas C.A. and V. Visweswaran (1995), "Quadratic optimization", in *Handbook of Global Optimization*, edited by Horst R. and P.M. Pardalos, Nonconvex Optimization and Its Applications, vol.2, Kluwer Academic Publishers, Dordrecht, pp.217-269.

[7] Floudas C.A., Pardalos P.M. et al. (1999), *Handbook of Test Problems in Local and Global Optimization*, Nonconvex Optimization and Its Applications, vol.33, Springer, Berlin.

[8] Hiriart-Urruty J.B. (1985), "Generalized differentiability, duality and optimization for problems dealing with differences of convex functions", in *Convexity and Duality in Optimization*, Lecture Notes in Economics and Mathematical Systems, vol.256, Springer-Verlag.

[9] Horst R. and P.M. Pardalos (eds.) (1995), *Handbook of Global Optimization*, Nonconvex Optimization and Its Applications, vol.2, Kluwer Academic Publishers, Dordrecht.

[10] Horst R. and H. Tuy (1996), *Global optimization: Deterministic approaches*, 3rd rev., Springer Verlag, Berlin.

[11] Horst R. and N.V. Thoai (1999), "DC programming: Overview", *Journal of Optimization Theory and Applications*, vol.103, pp.1-43.

[12] Horst R., Pardalos P.M. and N.V. Thoai (2001), *Introduction to Global Optimization*, 2nd ed., Nonconvex Optimization and Its Applications, vol.48, Springer, Berlin.

[13] Konno H. and T. Kuno (1995), "Multiplicative programming problems", in *Handbook of Global Optimization*, edited by Horst R. and P.M. Pardalos, Nonconvex Optimization and Its Applications, vol.2, Kluwer Academic Publishers, Dordrecht, pp.369-405.

[14] Konno H., Thach P.T. and H. Tuy (1997), *Optimization on low rank nonconvex structures*, Nonconvex Optimization and Its Applications, vol.15, Kluwer Academic Publishers, Dordrecht.

[15] Konno H. and K. Fukaishi (2000), "A branch and bound algorithm for solving low-rank linear multiplicative and fractional programming problems", *Journal of Global Optimization*, vol.18, pp.283-299.

[16] Le Thi Hoai An and Pham Dinh Tao (1997), "Solving a class of linearly constrained indefinite quadratic problems by D.C. algorithms", *Journal of Global Optimization*, vol.11, pp.253-285.

[17] Phong Thai Quynh, An Le Thi Hoai and Tao Pham Dinh (1995), "Decomposition branch and bound method for globally solving linearly constrained indefinite quadratic minimization problems", *Operations Research Letters*, vol.17, no.5, pp.215-220.

[18] Ryoo H.-S. and N.V. Sahinidis (2003), "Global optimization of multiplicative programs", *Journal of Global Optimization*, vol.26, pp.387-418.

[19] Schaible S. and C. Sodini (1995), "Finite algorithm for generalized linear multiplicative programming", *Journal of Optimization Theory and Applications*, vol.87, n.2, pp.441-455.

[20] Tuy H. (1995), "D.C. optimization: theory, methods and algorithms", in *Handbook of Global Optimization*, edited by Horst R. and P.M. Pardalos, Nonconvex Optimization and Its Applications, vol.2, Kluwer Academic Publishers, Dordrecht, pp.149-216.

[21] Tuy H. (1996), "A general D.C. approach to location problems", in *State of the art in global optimization*, edited by Floudas C.A. and P.M. Pardalos, Nonconvex Optimization and Its Applications, vol.7, Kluwer Academic Publishers, Dordrecht, pp.413-432.

[22] Tuy H. (1998), *Convex analysis and global optimization*, Nonconvex Optimization and Its Applications, vol.22, Kluwer Academic Publishers, Dordrecht.