Global optimization of a generalized linear multiplicative program

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

In this paper a solution algorithm for a class of generalized linear multiplicative programs having a polyhedral feasible region is proposed. The algorithm is based on the so called optimal level solutions method. Some optimality conditions are used to improve the performance of the proposed algorithm. Results of a computational test are provided.

Key words: generalized linear programming, optimal level solutions, global optimization.
AMS - 2000 Math. Subj. Class. 90C05, 90C26, 90C31.
JEL - 1999 Class. Syst. C61, C63.

1 Introduction

The aim of this paper is to study, from both a theoretical, an algorithmic and a computational point of view, the following class of generalized linear problems:

$$P: \begin{cases} \inf \;\; f(x) = c^T x + (q^T x + q_0)\phi(d^T x + d_0) \\ \qquad x \in X = \{x \in \Re^n : \;\; Ax \leq b\} \end{cases}$$

where $A \in \Re^{m\times n}$, $b \in \Re^m$, $c, d, q \in \Re^n$, $q_0, d_0 \in \Re$ and $X \neq \emptyset$. The scalar function $\phi(\xi)$ is assumed to be continuous and strictly monotone, and is defined for all the values in $\Lambda = \left\{\xi \in \Re : \;\; \xi = d^T x + d_0, \;\; x \in X\right\}$.

Actually, in the rest of the paper we will consider only the case of a function $\phi(\xi)$ strictly increasing. This is not a loss of generality since in the case of a strictly decreasing function $\phi(\xi)$ we just have to rewrite the objective function as $f(x) = c^T x + (\hat{q}^T x + \hat{q}_0)\hat{\phi}(d^T x + d_0)$ where $\hat{q} = -q$, $\hat{q}_0 = -q_0$ and $\hat{\phi}(\xi) = -\phi(\xi)$, so that $\hat{\phi}(\xi)$ results to be strictly increasing. It is worth pointing out also that the strict monotonicity assumption regarding to function $\phi(\xi)$ is not restrictive, in the sense that it may be possible to partition the feasible region X in subsets where $\phi(\xi)$ is either strictly monotone or constant with respect to levels ξ.

The solution method proposed to solve this class of problems is based on the so called "optimal level solutions" method (see [1, 2, 3, 4, 5, 8]). It is known that this is a parametric method, which finds the optimum of the problem by determining the minima of particular subproblems. In particular, the optimal solutions of these subproblems are obtained by means of a sensitivity analysis which maintains the optimality conditions.

In Section 2 we describe how the optimal level solutions method can be applied to problem P; in Section 3 a solution algorithm is proposed and fully described; in Section 4 the results of a computational test are provided and discussed.

2 A parametric approach

In this section we show how problem P can be solved by means of the so called *optimal level solutions approach* (see for all [2, 3, 8]). With this aim, let $\xi \in \Re$ be a real parameter and let us define the corresponding parametrical subset of X:

$$X_\xi = \{x \in \Re^n : \; Ax \leq b, \; d^T x + d_0 = \xi\}$$

In the same way, the following further subset of X can be defined:

$$X_{[\xi_1,\xi_2]} = \{x \in \Re^n : \; Ax \leq b, \; \xi_1 \leq d^T x + d_0 \leq \xi_2\}$$

The parameter $\xi \in \Re$ is said to be a *feasible level* if the set X_ξ is nonempty; in this light, the previously defined set $\Lambda = \left\{\xi \in \Re : \; \xi = d^T x + d_0, \; x \in X\right\}$ is nothing but the set of all feasible levels. The convexity of the polyhedron X implies that the set Λ is a closed convex interval. In this light, the following further notations can be introduced:

$$\xi_{min} = d_0 + \inf_{x \in X} d^T x \quad \text{and} \quad \xi_{max} = d_0 + \sup_{x \in X} d^T x$$

Clearly, if X is a compact set then ξ_{min} and ξ_{max} are finite and the set Λ of feasible levels is compact too. The following parametric subproblem can then be obtained just by adding to problem P the constraint $d^T x + d_0 = \xi$:

$$P_\xi : \left\{ \begin{array}{c} \inf \; f_\xi(x) \\ x \in X_\xi \end{array} \right.$$

where:

$$f_\xi(x) = c^T x + (q^T x + q_0)\phi(\xi) = (c + \phi(\xi)q)^T x + \phi(\xi)q_0$$

An optimal solution of problem P_ξ, if it exists, is called an *optimal level solution*. Given a feasible level $\xi \in \Re$, the set of optimal solutions of P_ξ is denoted with $S_\xi \subset X_\xi$, while the set of all the optimal level solutions is denoted with $S = \cup_{\xi\in\Lambda} S_\xi \subset X$. Obviously, an optimal solution of problem P is also an optimal level solution and, in particular, it is the optimal level solution with the smallest value. The idea of this approach is then to scan all the feasible levels, studying the corresponding optimal level solutions, until the minimizer of the problem is reached.

In order to propose such a kind of solution algorithm some properties of optimal level solutions are needed. With this aim, let $\xi' \in \Lambda$ and assume that the corresponding optimal level solution x' exists; notice that x' is the optimal solution for the linear programming problem $P_{\xi'}$:

$$P_{\xi'} : \begin{cases} \inf\ (c + \phi(\xi')q)^T x + \phi(\xi')q_0 \\ Ax \le b \\ d^T x = \xi' - d_0 \end{cases}$$

We can also assume that x' is a vertex of $X_{\xi'}$, so that it is either a vertex of X or a point belonging to an edge of X. Let:

$$\begin{array}{rcl} B & = & \{i\ :\ A_i x' = b_i,\ i = 1, \dots, m\} \\ N & = & \{1, \dots, m\} \setminus B \end{array}$$

where A_i is the i-th row of A, and let A_B, A_N, b_B and b_N be the corresponding submatrices of A and b, respectively. The following theorem shows the existence of segments of optimal level solutions.

Theorem 2.1 *Consider problem P and let x' and x'' be optimal level solutions corresponding to the feasible levels ξ' and ξ'', $\xi' < \xi''$, respectively. If x' and x'' share the same set of binding constraints B and the columns of $\left[A_B^T\ d\right]$ are linearly independent then the points of the segment $x'(\theta) = x' + \theta\frac{x''-x'}{\xi''-\xi'}$, $\theta \in [0, \xi'' - \xi']$, are optimal level solutions corresponding to the feasible levels $\xi' + \theta$, respectively.*

Proof The points in the segment $x'(\theta)$, $\theta \in [0, \xi'' - \xi']$, are trivially feasible due to the convexity of X. First, it is worth noticing that all the points in such a segment shares the same set of binding constraints B. This follows since:

$$\begin{array}{ll} A_B x' = b_B\ ,\ A_B x'' = b_B & \Rightarrow A_B x'(\theta) = b_B\ \forall\theta \in (0, \xi'' - \xi') \\ A_N x' < b_N\ ,\ A_N x'' < b_N & \Rightarrow A_N x'(\theta) < b_N\ \forall\theta \in (0, \xi'' - \xi') \end{array}$$

Being x' and x'' optimal level solutions and being the columns of $\left[A_B^T \; d\right]$ linearly independent then the following Karush-Kuhn-Tucker conditions are both necessary and sufficient:

$$\begin{array}{rcl} c + \phi(\xi')q = A_B^T \mu'_B + d\lambda' & \text{with} & \mu'_B \leqq 0,\ \mu'_N = 0 \\ c + \phi(\xi'')q = A_B^T \mu''_B + d\lambda'' & \text{with} & \mu''_B \leqq 0,\ \mu''_N = 0 \end{array}$$

It yields that:

$$\begin{cases} \mu'_B = \mu_B^{(c)} + \phi(\xi')\mu_B^{(q)} \\ \lambda' = \lambda^{(c)} + \phi(\xi')\lambda^{(q)} \end{cases} \qquad \begin{cases} \mu''_B = \mu_B^{(c)} + \phi(\xi'')\mu_B^{(q)} \\ \lambda'' = \lambda^{(c)} + \phi(\xi'')\lambda^{(q)} \end{cases}$$

where:

$$\begin{bmatrix} A_B \\ d^T \end{bmatrix}^T \begin{bmatrix} \mu_B^{(c)} \\ \lambda^{(c)} \end{bmatrix} = c \quad \text{and} \quad \begin{bmatrix} A_B \\ d^T \end{bmatrix}^T \begin{bmatrix} \mu_B^{(q)} \\ \lambda^{(q)} \end{bmatrix} = q \tag{1}$$

To prove the level optimality of $x'(\theta)$, $\theta \in (0, \xi'' - \xi')$, we just have to verify the corresponding Karush-Kuhn-Tucker conditions. With this aim, for all $\theta \in [0, \xi'' - \xi']$ let:

$$\begin{array}{rcl} \mu'_B(\theta) & = & \mu_B^{(c)} + \phi(\xi' + \theta)\mu_B^{(q)} \\ \mu'_N(\theta) & = & 0 \\ \lambda'(\theta) & = & \lambda^{(c)} + \phi(\xi' + \theta)\lambda^{(q)} \end{array}$$

Then, it yields:

$$c + \phi(\xi' + \theta)q = A_B^T \mu'_B(\theta) + d\lambda'(\theta)$$

so that just the nonpositivity of $\mu'_B(\theta)$ is left to be verified. Since the scalar function $\phi(\xi)$ is continuous and strictly increasing then for all $\theta \in [0, \xi'' - \xi']$ it is:

$$\phi(\xi') \leq \phi(\xi' + \theta) \leq \phi(\xi'')$$

Hence, from $\mu'_B(0) = \mu_B^{(c)} + \phi(\xi')\mu_B^{(q)} \leq 0$ and $\mu'_B(\xi'' - \xi') = \mu_B^{(c)} + \phi(\xi'')\mu_B^{(q)} \leq 0$ it follows $\mu'_B(\theta) = \mu_B^{(c)} + \phi(\xi' + \theta)\mu_B^{(q)} \leq 0$ for all $\theta \in (0, \xi'' - \xi')$. □

The previous theorem showed that two optimal level solutions x' and x'' sharing the same set B of binding constraints are extrema of a segment of optimal level solutions. It is now worth determining the biggest segment (or halfline) of optimal level solutions containing x' and x''. With this aim, let $x'(\theta) = x' + \theta\Delta_x$ and let us use the notations introduced in the proof of Theorem 2.1. Since $A_B x'(\theta) = b_B$ for all $\theta \in \Re$, then points $x'(\theta)$ are feasible whenever $A_N x'(\theta) \leq b_N$, that is for all the values $\theta \in [F_L, F_R]$ where:

$$F_L = \begin{cases} -\infty & \text{if } A_N \Delta_x \geqq 0 \\ \max\limits_{i \in N,\ A_i \Delta_x < 0} \left\{ \frac{b_i - A_i x'}{A_i \Delta_x} \right\} & \text{otherwise} \end{cases} \tag{2}$$

$$F_R = \begin{cases} +\infty & \text{if } A_N \Delta_x \leqq 0 \\ \min\limits_{i \in N,\ A_i \Delta_x > 0} \left\{ \frac{b_i - A_i x'}{A_i \Delta_x} \right\} & \text{otherwise} \end{cases} \tag{3}$$

The points $x'(\theta)$, $\theta \in [F_L, F_R]$, results to be optimal level solutions if $\mu'_B(\theta) = \mu_B^{(c)} + \phi(\xi' + \theta)\mu_B^{(q)} \leq 0$, that is for values θ such that $\phi(\xi' + \theta) \in [\alpha, \beta]$, where:

$$\alpha = \begin{cases} -\infty & \text{if } \mu_B^{(q)} \geqq 0 \\ \max\limits_{i \in B,\ \mu_i^{(q)} < 0} \left\{ \frac{-\mu_i^{(c)}}{\mu_i^{(q)}} \right\} & \text{otherwise} \end{cases} \tag{4}$$

$$\beta = \begin{cases} +\infty & \text{if } \mu_B^{(q)} \leqq 0 \\ \min\limits_{i \in B,\ \mu_i^{(q)} > 0} \left\{ \frac{-\mu_i^{(c)}}{\mu_i^{(q)}} \right\} & \text{otherwise} \end{cases} \tag{5}$$

Let us denote with ϕ^{-1} the inverse of the strictly increasing function ϕ, and let $\phi_{\xi_{max}} = \lim_{\xi \to \xi_{max}^-} \phi(\xi)$ and $\phi_{\xi_{min}} = \lim_{\xi \to \xi_{min}^+} \phi(\xi)$, so that the following values can be determined:

$$O_L = \begin{cases} \xi_{min} - \xi' & \text{if } \phi_{\xi_{min}} \geq \alpha \\ \phi^{-1}(\alpha) - \xi' & \text{if } \phi_{\xi_{min}} < \alpha \end{cases} \tag{6}$$

$$O_R = \begin{cases} \xi_{max} - \xi' & \text{if } \phi_{\xi_{max}} \leq \beta \\ \phi^{-1}(\beta) - \xi' & \text{if } \phi_{\xi_{max}} > \beta \end{cases} \tag{7}$$

We can then say that the optimality is guaranteed for all the values of $\theta \in [O_L, O_R]$ while the feasibility is guaranteed for all the values $\theta \in [F_L, F_R]$. As a consequence, the biggest segment (or halfline) of optimal level solutions containing x' and x'' is given by $x'(\theta)$ with:

$$\theta \in [\theta_L, \theta_R] \text{ where } \theta_L = \max\{O_L, F_L\} \text{ and } \theta_R = \min\{O_R, F_R\}$$

The previously described behaviour suggests how to algorithmically solve the problem. Starting from an optimal level solution a segment (or half-line) of optimal level solutions can be scanned; whenever the feasibility or the optimality is lost, we just have to change the considered subset of the binding constraints B and iteratively continue the visit of optimal level solutions. During this visit, the objective function $f(x)$ can be evaluated over the set of optimal level solution thus obtaining the global optimum.

Clearly, the described approach iteratively visits a finite number of vertices and edges of X in a simplex-like way, and this guarantees the finiteness of the method itself.

Finding the subset of the binding constraints B to be used in order to determine the segment of optimal level solutions, expecially in case of degeneration, could be not so easy. For this reason we apply a numerical approach for determining the segment of optimal level solutions. Let x' be the optimal level solution corresponding to the level ξ' and let x'_δ be the optimal level solution corresponding to the level $\xi' + \delta$, with $\delta > 0$ small enough to guarantee that x'_δ is not a vertex of X and that x' and x'_δ belong to the same segment of optimal level solutions. As a consequence, it is $B(x'_\delta) \subseteq B(x')$ where $B(x')$ and $B(x'_\delta)$ are the sets of binding constraints corresponding to x' and x'_δ, respectively. Notice that the strict inclusion holds only in the case x' is a vertex of X. Hence, we just have to determine a subset $\tilde{B}$ of $B(x'_\delta)$ such that the columns of $\left[A^T_{\tilde{B}}\ d\right]$ are linearly independent and $\operatorname{rank}\left(\left[A^T_{\tilde{B}}\ d\right]\right) = \operatorname{rank}\left(\left[A^T_{B(x'_\delta)}\ d\right]\right)$. It is worth noticing that in order to determine $\tilde{B}$ just the optimal level solution x'_δ has to be considered. Notice also that in case of nondegeneracy it is $\tilde{B} = B(x'_\delta)$ being x'_δ not a vertex of X.

The described approach is summarized in procedure "*Parameters*()" where x' is an optimal level solution which belongs to an edge of X but is not a vertex of X. This procedure determines a segment (or halfline) of optimal level solutions of the kind $x' + \theta\Delta_x$, $\theta \in (\theta_L, \theta_R)$. With this aim, notice that for all $\theta \in (\theta_L, \theta_R)$ it is:

$$\begin{cases} A_{\tilde{B}}(x' + \theta\Delta_x) & = & b_{\tilde{B}} \\ d^T(x' + \theta\Delta_x) & = & \xi' + \theta \end{cases}$$

so that it yields:

$$\begin{bmatrix} A_{\tilde{B}} \\ d^T \end{bmatrix} \Delta_x = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{8}$$

3 Solution algorithm

In order to find a global minimum (assuming that one exists) it would be necessary to solve problems P_ξ for all the feasible levels. In this section we will show that this can be done by means of a finite number of iterations, using the results of the previous section. The method scans all the feasible levels starting from a certain feasible level ξ'; the levels $\xi > \xi'$ are visited in increasing order, while the levels $\xi < \xi'$ are visited in decreasing order.

Procedure **Parameters**(*inputs*: x'; *outputs*: Δ_x, F_L, F_R, O_L, O_R, θ_L, θ_R)
let $B(x') = \{i \ : \ A_i x' = b_i, \ i = 1, \ldots, m\}$ and $N = \{1, \ldots, m\} \setminus B(x')$;
let $\tilde{B} \subseteq B(x')$ such that the columns of $\left[A_{\tilde{B}}^T \ d\right]$ are linearly independent and $\operatorname{rank}\left(\left[A_{\tilde{B}}^T \ d\right]\right) = \operatorname{rank}\left(\left[A_{B(x')}^T \ d\right]\right)$;
let Δ_x be the solution of the linear system (8);
let $\mu_{\tilde{B}}^{(c)}$, $\lambda^{(c)}$, $\mu_{\tilde{B}}^{(q)}$ and $\lambda^{(q)}$ be the solutions of the linear systems (1);
let F_L, F_R, α, β, O_L, O_R as described in (2), (3), (4), (5), (6), (7);
set $\theta_L := \max\{O_L, F_L\}$ and $\theta_R := \min\{O_R, F_R\}$;
end proc.

For the sake of convenience, the described algorithm visits the feasible levels only in increasing order, hence the levels $\xi < \xi'$ can be analyzed by solving the following problem which is equivalent to P:

$$P \equiv \tilde{P} : \begin{cases} \inf \ f(x) = c^T x + (\tilde{q}^T x + \tilde{q}_0)\tilde{\phi}(\tilde{d}^T x + \tilde{d}_0) \\ \qquad\qquad x \in X \end{cases}$$

where $\tilde{\phi}(y) = -\phi(-y)$, $\tilde{q} = -q$, $\tilde{d} = -d$, $\tilde{q}_0 = -q_0$ and $\tilde{d}_0 = -d_0$. In this light, $\tilde{\phi}(y)$ is an increasing function just like $\phi(y)$, and the decreaseness of the feasible levels of P corresponds to the increaseness of the feasible levels of $\tilde{P}$.

The following procedures "*Main*()" and "*Visit*()" can then be proposed. Procedure "*Main*()" initialize the algorithm by determining the set of feasible levels and a starting incumbent solution, then it uses procedure "*Visit*()" to obtain the global optimal solution (if it exists). Notice that in procedure "*Main*()" there is also one more optional subprocedure, namely "*ImproveStartingValues*()", which is aimed to improve the starting incumbent optimal solution. This optional procedure will be discussed later.

Procedure "*Visit*()" scans iteratively the given set of feasible levels obtaining the best solution. Notice that "*Visit*()" uses a subprocedure "*MinRestriction*()" which determines the minimum of the continuous single valued function $z(\theta)$ in a closed interval. Observe that procedure "*MinRestriction*()" can be implemented numerically, and eventually improved for specific functions $f(x)$ (see [2, 3, 8]).

Notice that the optional subprocedure "*ImplicitVisit*()" evaluates the opportunity of avoiding the explicit visit of some feasible levels whenever $F_R < O_R$. Specifically speaking, in the interval $[0, O_R - F_R]$ the function $z(\theta)$ represents an underestimation for the objective function; as a consequence, if $\min_{\theta \in [0, O_R - F_R]} z(\theta) \geq UB$ there is no need to visit such an interval of feasible levels and the procedure can skip to the level $\xi' + (O_R - F_R)$.

Procedure **Main**(*inputs*: P; *outputs*: *Opt*, *OptVal*)
  Compute the values $\xi_{min} := d_0 + \inf_{x\in X} d^T x$ and $\xi_{max} := d_0 + \sup_{x\in X} d^T x$;
  Let $\xi' \in (\xi_{min}, \xi_{max})$;
  if $\inf_{x\in X_{\xi'}} f_{\xi'}(x) = -\infty$ *then* $Opt := [\,]$ and $OptVal := -\infty$
  else
    if $\phi(\xi)$ is strictly decreasing *then*
      $\phi(\xi) := -\phi(\xi)$, $q := -q$, $q_0 := -q_0$
    end if;
    Let $x' := \arg\min_{x\in X_{\xi'}} f_{\xi'}(x)$, $\bar{x} := x'$ and let $UB := f(\bar{x})$;
    # Optional : $[x', \xi', \bar{x}, UB] := ImproveStartingValues()$;
    $[\overline{x}, UB] := Visit(P, \xi', \xi_{max}, x', \overline{x}, UB)$;
    $[\overline{x}, UB] := Visit(\tilde{P}, -\xi', -\xi_{min}, x', \overline{x}, UB)$;
    $Opt := \overline{x}$ and $OptVal := UB$;
  end if;
end proc.

Procedure **Visit**(*inputs*: P, ξ', ξ_{max}, x', $\bar{x}$, UB; *outputs*: *Opt*, *OptVal*)
  let $\delta > 0$ be the step parameter;
  while $\xi' < \xi_{max}$
   if $\inf_{x\in X_{\xi'+\delta}} f_{\xi'+\delta}(x) = -\infty$ *then* $\overline{x} := [\,]$; $UB := -\infty$; $\xi' := \xi_{max}$
   else
    let $\xi' = \xi' + \delta$;
    let $x' := \arg\min\{P_{\xi'}\}$ be an optimal level solution not vertex of X;
    $[\Delta_x, F_L, F_R, O_L, O_R, \theta_L, \theta_R] := Parameters(x')$;
    let $z(\theta) = (c + \phi(\xi' + \theta)q)^T (x' + \theta\Delta_x) + \phi(\xi' + \theta)q_0$;
    set $[\overline{\theta}, z_{inf}] := MinRestriction(z(\theta), [\theta_L, \theta_R])$;
    if $z_{inf} = -\infty$ *then* $\overline{x} := [\,]$; $UB := -\infty$; $\xi' := \xi_{max}$ *else*
     if $z_{inf} < UB$ *then* $UB := z_{inf}$; $\overline{x} := x' + \overline{\theta}\Delta_x$ *end if*;
     set $\xi' := \xi' + \theta_R$;
     # Optional : $\xi' := ImplicitVisit(\xi', F_R, O_R)$;
    end if;
   end if;
  end while;
  $Opt := \overline{x}$; $OptVal := UB$;
end proc.

Procedure **ImplicitVisit**(*inputs*: ξ', F_R, O_R; *outputs*: ξ')
  if $F_R < O_R$ and $\xi' < \xi_{max}$ *then*
    set $x' := x' + \theta_R \Delta_x$;
    let $z(\theta) = (c + \phi(\xi' + \theta)q)^T(x' + \theta\Delta_x) + \phi(\xi' + \theta)q_0$;
    set $\tilde{\theta} := \min\{O_R - F_R; \xi_{max} - \xi'\}$;
    set $[\overline{\theta}, z_{inf}] := MinRestriction(z(\theta), [0, \tilde{\theta}])$;
    if $z_{inf} \geq UB$ *then* $\xi' := \xi' + \tilde{\theta}$ *end if*;
  end if;
end proc.

In this light, as better is the value UB of the incumbent optimal solution as more effective is the implicit visit of the levels $\xi + \theta$ with $\theta \in [0, O_R - F_R]$. For this very reason, a "good" starting optimal level solution could improve the performance of the algorithm. The role of subprocedure "*ImproveStartingValues*()" is indeed to improve the starting optimal level solution by comparing it with the optimal level solutions corresponding to feasible levels close to ξ_{min} and ξ_{max}. Further improvements can be obtained by applying an algorithm looking for local optima starting from $\bar{x}$.

The correctness of the proposed algorithm follows since all the feasible levels are scanned (either explicitly or implicitly) and the optimal solution, if it exists, is also an optimal level solution.

It remains to verify the convergence (finiteness), that is to say that the procedure stops after a finite number of steps. With this aim it is worth pointing out that:

- if $O_R < F_R$ at least one of the multipliers corresponding to the binding constraints vanishes in a relative interior point of the edge. As a consequence, the feasible level $\xi' + O_R$ admits alternative optimal level solutions; in the next iteration a new edge (obviously not connected with the previous one) is determined and visited;

- if $F_R \leq O_R$ the whole edge is visited, the optimal solution corresponding to the feasible level $\xi' + F_R$ is a vertex of X, and in the next iteration a new edge is determined and visited.

As a consequence, at every iterative step of the proposed algorithm, a new edge of the feasible region is visited (partially, if the case); note also that the level is increased from ξ' to at least $\xi' + \theta_R > \xi'$, so that it is not possible to consider an already visited edge; the convergence then follows since in a polyhedron there is a finite number of possible edges.

4 Computational results

The previously described procedures have been fully implemented with the software MatLab 7.8 R2009a on a Mac OSX computer having 6 Gb RAM and two Xeon dual core processors at 2.66 GHz.

For the sake of convenience, in the computational test we considered functions $f(x) = c^T x + (q^T x + q_0)(d^T x + d_0)^\alpha$ with $\alpha = 3, 1, -1, -3$ so that both multiplicative and fractional problems are considered (see for all [6, 7]).

We considered problems with a number of variables from $n = 10$ to $n = 100$ and a number of inequality constraints equal to $m = \lceil \frac{7}{2} n \rceil$.

The problems have been randomly created; in particular, matrices, vectors and scalars $A \in \Re^{m \times n}$, $c, d, q \in \Re^n$, $b \in \Re^m$ and $q_0, d_0 \in \Re$ have been generated with components in the interval [-10,10] by using the "randi()" MatLab function (integer numbers generated with uniform distribution). In the fractional case, that is for $\alpha < 0$, the value $d_0 \in \Re$ has been chosen in order to have function $d^T x + d_0$ positive over the feasible region. Within the procedures, the linear problems have been solved with the "linprog()" MatLab function. Various instances have been randomly generated and solved. In particular, for the values of n equal to 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, we solved a number of random problems equal to 1000, 1000, 600, 300, 160, 80, 60, 40, 20, 20, respectively.

The average number of iterations (that is, number of solved relaxed subproblems) and the average CPU times spent by the algorithm to solve the instances are given as the results of the test (see Table 1).

Regarding to Table 1 notice that:

- "α" represents the power of the considered function $\phi(\xi) = \xi^\alpha$;
- "**n**" represents the number of variables in the considered problems;
- the columns corresponding to T_1, T_2 and T_3, provide the results obtained by using no improvements (T_1), by using just "*ImplicitVisit*()" subprocedure (T_2), by using both "*ImplicitVisit*()" and "*ImproveStartingValues*()" subprocedures (T_3).

The obtained results point out the effectiveness of the improvements proposed in Section 3 and a different behaviour of the multiplicative case with respect to the fractional one. In particular, the performance increases as the value of α decreases. The results related to fractional functions $\phi(\xi)$ (that is $\alpha < 0$) are much better that the ones related to multiplicative functions $\phi(\xi)$ (that is $\alpha > 0$).

5 Conclusions

The proposed algorithm allows to solve a class of nonconvex problems. The computational test shows that it is possible to efficiently handle problems with up to 100 variables. In particular, the improvement criteria suggested in Section 3 resulted to be extremely effective in making the algorithm efficient. The correctness of the method guarantees that the global minimum is found even in the case of unbounded feasible regions.

References

[1] Cambini R. (1994) A class of non-linear programs: theoretical and algorithmic results. In: Generalized Convexity, S. Komlósi, T. Rapcsák and S. Schaible (eds), Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin, 405:294-310.

[2] Cambini R. and C. Sodini (2003) A finite algorithm for a class of nonlinear multiplicative programs. Journal of Global Optimization, 26:279-296.

[3] Cambini R. and C. Sodini (2007) An unifying approach to solve a class of parametrically-convexifiable problems. In: Generalized Convexity and Related Topics, I.V. Konnov, D.T. Luc and A.M. Rubinov (eds), Lecture Notes in Economics and Mathematical Systems, Springer, Berlin, 583:149-166.

[4] Cambini R. and C. Sodini (2009) Global optimization of a rank-two nonconvex program. Mathematical Methods of Operations Research, published online.

[5] Ellero A. (1996) The optimal level solutions method. Journal of Information & Optimization Sciences, 17:355-372.

[6] Frenk J.B.G. and S. Schaible (2005) Fractional programming. In: Handbook of Generalized Convexity and Generalized Monotonicity, Nonconvex Optimization and Its Applications, 76:335-386, Springer, New York.

[7] Konno H. and T. Kuno (1995) Multiplicative programming problems. In: Handbook of Global Optimization, R. Horst and P.M. Pardalos (eds), Nonconvex Optimization and Its Applications, Kluwer Academic Publishers, Dordrecht, 2:369-405.

[8] Schaible S. and C. Sodini (1995) A finite algorithm for generalized linear multiplicative programming. Journal of Optimization Theory and Applications, 87:441-455.

α	**n**	Number of Iterations			CPU Times (seconds)		
		T_1	T_2	T_3	T_1	T_2	T_3
3	10	34.475	25.499	24.385	1.3944	1.4102	1.413
3	20	96.475	73.344	69.989	5.3343	5.0209	4.9237
3	30	177.41	136.62	129.69	14.253	12.679	12.266
3	40	270.27	204.6	196.55	30.836	25.576	25.052
3	50	374.31	287.99	273.23	63.012	51.484	49.539
3	60	500.45	391.75	369.62	117.79	96.147	91.827
3	70	621.72	485.05	463.25	196.62	157.28	151.66
3	80	757.9	598.25	566.48	314.51	251.63	240.29
3	90	875.8	666.3	633.6	466.29	358.41	348.5
3	100	955.75	724.2	705.7	625.99	475	467.46
1	10	34.265	23.345	21.726	1.3578	1.2372	1.2344
1	20	96.222	66.109	60.377	5.1877	4.2784	4.0111
1	30	176.96	121.65	109.11	13.948	10.724	9.7367
1	40	270.6	182.65	162.25	30.46	21.924	19.631
1	50	375.56	257.6	221.66	62.638	44.671	38.505
1	60	501.57	343.05	295.85	117.28	82.02	71.113
1	70	622.83	430.42	366.65	195.87	136.9	117.23
1	80	759.88	528.65	450.32	314.41	219.65	188.19
1	90	881	593.35	508.35	468.03	316.05	273.06
1	100	956.15	642.35	558.7	627.4	416.53	366.92
-1	10	22.945	8.938	7.382	0.97263	0.61754	0.57573
-1	20	57.068	13.754	11.737	3.3237	1.2265	1.2211
-1	30	99.505	18.478	16.04	8.3892	2.2019	2.2208
-1	40	147.19	22.773	19.133	17.547	3.5825	3.5077
-1	50	206.36	27.35	24.169	36.124	6.0076	6.0331
-1	60	270.44	31.125	27.387	66.105	9.2639	9.2497
-1	70	333.43	36.367	32.35	109.19	13.948	13.941
-1	80	404.15	42.5	37.425	172.87	20.75	20.269
-1	90	473.25	46.25	43.75	260.89	28.745	29.63
-1	100	540.1	53.7	43.5	366.53	40.487	36.086
-3	10	23.379	8.503	6.784	0.98345	0.57711	0.51309
-3	20	56.456	12.768	10.202	3.2898	1.1156	1.0096
-3	30	98.252	17.078	13.362	8.2859	2.007	1.8056
-3	40	146.39	21.6	15.56	17.454	3.3535	2.8069
-3	50	204.33	25.094	19.831	35.731	5.493	5.0298
-3	60	269.29	28.613	21.55	65.635	8.4711	7.3846
-3	70	328.13	34.3	27.2	107.34	13.139	11.949
-3	80	398.43	39.6	29.975	169.81	19.341	16.658
-3	90	471.85	40.6	28.6	258.38	25.259	20.581
-3	100	537.45	46.8	32.05	363.31	35.353	27.996

Table 1: Computational Results