J Optim Theory Appl (2008) 137: 105–120
DOI 10.1007/s10957-007-9323-9

Global Maximization of a Generalized Concave Multiplicative Function

H.P. Benson

Published online: 11 December 2007

Abstract This article presents a branch-and-bound algorithm for globally solving the problem (P) of maximizing a generalized concave multiplicative function over a compact convex set. Since problem (P) does not seem to have been studied previously, the algorithm is apparently the first algorithm to be proposed for solving this problem. It works by globally solving a problem (P1) equivalent to problem (P). The branch-and-bound search undertaken by the algorithm uses rectangular partitioning and takes place in a space which typically has a much smaller dimension than the space to which the decision variables of problem (P) belong. Convergence of the algorithm is shown; computational considerations and benefits for users of the algorithm are given. A sample problem is also solved.

Keywords Global optimization · Branch-and-bound algorithms · Multiplicative programming · Quadratic programming · Bilinear programming

1 Introduction

1.1 Problem Statement

The problem of central interest in this article is given by

$$\text{(P)} \quad v = \max_{x \in X} f(x) \triangleq \max_{x \in X} g(x) + \sum_{i=1}^{p} f_i(x) g_i(x),$$

where $p \geq 2, g, f_i$ and $g_i, i = 1, 2, \ldots, p$, are concave functions defined on $\mathbb{R}^n$, $X \subseteq \mathbb{R}^n$ is a nonempty, compact convex set, and, for each $i = 1, 2, \ldots, p$, $f_i(x) > 0$

H.P. Benson (✉)
Department of Decision and Information Sciences, University of Florida, Gainesville, FL, USA
e-mail: harold.benson@cba.ufl.edu

and $g_i(x) > 0$ for all $x \in X$. Problem (P) does not appear to have been studied previously. Apparently, the problem in the literature most closely related to problem (P) is the generalized convex multiplicative programming problem studied by Konno, Kuno, and Yajima [1]. This is the problem obtained from (P) by minimizing (rather than maximizing) $f(x)$ over X under the same assumptions as for problem (P), except that g, f_i and g_i, $i = 1, 2, \ldots, p$, are assumed to be convex, rather than concave, functions on $\mathbb{R}^n$. Adapting the terminology of Konno, Kuno and Yajima [1], we will refer to problem (P) as the generalized concave multiplicative programming problem. Although the generalized concave multiplicative programming problem appears to be new, it has many important applications.

1.2 Applications

Many applications of problem (P) follow from the observation that the problem

$$
\begin{aligned}
\text{(Q)} \quad & w = \max h(x) + \sum_{i=1}^{p} [\langle a^i, x\rangle + \alpha_i][\langle b^i, x\rangle + \beta_i], \\
& \text{s.t.} \quad x \in X,
\end{aligned}
$$

where p and X are as in problem (P), h is a concave function on $\mathbb{R}^n$, and no sign restrictions are imposed upon $\langle a^i, x\rangle + \alpha_i$ or $\langle b^i, x\rangle + \beta_i$, $i = 1, 2, \ldots, p$, is a special case of problem (P). To see this, as suggested by Konno, Kuno and Yajima [1], for each $i = 1, 2, \ldots, p$, let

$$
r_i < \min\left\{\min_{x\in X}[\langle a^i, x\rangle + \alpha_i], \min_{x\in X}[\langle b^i, x\rangle + \beta_i]\right\}.
$$

Next, in problem (P), let

$$
\begin{aligned}
g(x) &= h(x) + \sum_{i=1}^{p} [r_i\langle a^i, x\rangle + r_i\langle b^i, x\rangle + r_i(\alpha_i + \beta_i) - r_i^2], \\
f_i(x) &= \langle a^i, x\rangle + (\alpha_i - r_i), \quad i = 1, 2, \ldots, p,
\end{aligned}
$$

and

$$
g_i(x) = \langle b^i, x\rangle + (\beta_i - r_i), \quad i = 1, 2, \ldots, p.
$$

Then problem (P) is identical to problem (Q), and all of the required assumptions of problem (P) are fulfilled.

It follows that the applications of problem (P) include, for example, all of the applications of problem (Q). Since problem (Q) encompasses, for instance, general quadratic programming, bilinear programming and linear zero-one programming as special cases, these applications are quite numerous.

The general quadratic programming problem may be written

$$
\begin{aligned}
\max \quad & \frac{1}{2}y^T My + \langle d, y\rangle + \gamma, \\
\text{s.t.} \quad & y \in Y,
\end{aligned}
$$

where $Y \subseteq \mathbb{R}^n$ is a nonempty polytope, M is an $n \times n$ symmetric matrix of rank p, $d \in \mathbb{R}^n$ and $\gamma \in \mathbb{R}$ (see, e.g., Cambini and Sodini [2]). Given M, Tuy [3], for example, gives a simple constructive method for finding linearly independent vectors $v^1, v^2, \dots, v^p \in \mathbb{R}^n$ and vectors $w^1, w^2, \dots, w^p \in \mathbb{R}^n$ such that, for all $y \in \mathbb{R}^n$,

$$\frac{1}{2} y^T M y = \frac{1}{2} \sum_{i=1}^{p} \langle v^i, y \rangle \langle w^i, y \rangle.$$

Thus, the general quadratic programming problem can be expressed in the form of problem (Q), and the applications of problem (P) include all of the applications of general quadratic programming. Included among the latter, for example, are quadratic zero-one programming problems [4], quadratic assignment problems [4], problems in economies of scale [5], the constrained linear regression problem [6], VLSI chip design problems [7], the linear complementarity problem [5], and portfolio analysis problems [6].

Problem (Q) also subsumes the bilinear programming problem as a special case. Let $f \in \mathbb{R}^q$, $u \in \mathbb{R}^r$, and let F and U be $q \times s$ and $r \times t$ matrices, respectively.

Assume that

$$Y = \{y \in \mathbb{R}^s | Fy \leq f, y \geq 0\}$$

and

$$Z = \{z \in \mathbb{R}^t | Uz \leq u, z \geq 0\}$$

are nonempty compact polyhedra. Then, the bilinear programming problem is given by

$$\begin{aligned} \min \quad & \langle a, y \rangle + \langle b, z \rangle + y^T C z, \\ \text{s.t.} \quad & y \in Y, \quad z \in Z, \end{aligned}$$

where $a \in \mathbb{R}^s$, $b \in \mathbb{R}^t$ and C is an $s \times t$ matrix of rank p. From Konno and Yajima [8], using a constructive procedure, the bilinear programming problem can be written in the form

$$\begin{aligned} \max \quad & \langle -a, y \rangle + \langle -b, z \rangle + \sum_{i=1}^{p} \langle v^i, y \rangle \langle w^i, z \rangle, \\ \text{s.t.} \quad & y \in Y, \quad z \in Z, \end{aligned}$$

where $v^i \in \mathbb{R}^s$, $w^i \in \mathbb{R}^t$, $i = 1, 2, \dots, p$. The latter problem is a special case of problem (Q) with $x^T = [y^T, z^T]$, $h(x) = \langle -a, y \rangle + \langle -b, z \rangle$, $(a^i)^T = [(v^i)^T, 0^T]$ and $(b^i)^T = [0^T, (w^i)^T]$, $i = 1, 2, \dots, p$, $\alpha_i = \beta_i = 0$, $i = 1, 2, \dots, p$, and

$$X = \{(y, z) \in \mathbb{R}^{s+t} | Fy \leq f, Uz \leq u, y, z \geq 0\}.$$

Therefore, included among the applications of problem (P) are all of the applications of bilinear programming. These include, for example, location-allocation problems [9], constrained bimatrix games [10], the three-dimensional assignment problem [11], certain linear max-min problems [12], and many problems in engineering design, economic management and operations research.

A linear zero-one programming problem may be written as

$$\begin{aligned}
\text{(I)} \quad & \max \quad \langle c, y\rangle, \\
& \text{s.t.} \quad y_j \in \{0,1\}, \quad j = 1,2,\dots,n, \\
& \qquad\; y \in Y,
\end{aligned}$$

where $c \in \mathbb{R}^n$ and $Y \subseteq \mathbb{R}^n$ is a nonempty polytope. From Raghavachari [13], for $M > 0$ sufficiently large, y^* is an optimal solution to problem (I) if and only if y^* is an optimal solution to the problem

$$\begin{aligned}
\text{(IC)} \quad & \max \quad \langle c, y\rangle + M\sum_{j=1}^{n}(-y_j)(1-y_j), \\
& \text{s.t.} \quad 0 \le y_j \le 1, \quad j = 1,2,\dots,n, \\
& \qquad\; y \in Y.
\end{aligned}$$

Since problem (IC) is a special case of problem (Q), it follows that all of the numerous applications of linear zero-one programming are included among the applications of problem (P). For an overview of some of these applications, see Nemhauser and Wolsey [14].

1.3 Problem Classification

For each $i = 1,2,\dots,p$, from [15] each term $f_i(x)g_i(x)$ in the objective function $f(x)$ of problem (P) is quasiconcave on X. However, since $f(x)$ involves the sum of these terms, $f(x)$ is, in general, not a quasiconcave function. Therefore, problem (P) may, in general, have local optimal solutions that are not global optimal solutions. For example, consider the following instance of problem (Q), which, as we have seen, is a special case of problem (P):

$$\begin{aligned}
\max \quad & (x_1 + x_2) + 4(-x_1)(x_2), \\
\text{s.t.} \quad & 2x_1 + x_2 \ge 12, \\
& x_1 + 2x_2 \ge 12, \\
& x_1, x_2 \le 6.
\end{aligned}$$

It can be shown that $x^0 = (6,3)$ and $x^1 = (3,6)$ are local optimal solutions for this problem, each with objective function value -63, but that x^0 and x^1 are not global optimal solutions. The global optimal value is -56 and is achieved at $x^2 = (4,4)$. It follows that problem (P) is a global optimization problem.

1.4 Purpose, Significance and Content

The purpose of this article is to present an algorithm for globally solving the generalized concave multiplicative programming problem (P). Since problem (P) does not seem to have been studied previously, the algorithm to be presented is apparently the first to be proposed for this problem.

The algorithm implements a rectangular, branch-and-bound search that finds a global optimal solution to a problem that is equivalent to problem (P). During the search, the required upper bounds are computed by solving ordinary convex programming problems. Each of these problems is guaranteed to have an optimal solution.

The algorithm has a number of potentially-attractive characteristics. First, the branch-and-bound search takes place in $\mathbb{R}^p$ rather than in $\mathbb{R}^n$ or $\mathbb{R}^{2p}$. In many applications, p is significantly smaller than n, so that this characteristic of the algorithm is expected to considerably shorten the length of the search. Second, the main work of the algorithm is confined to solving ordinary convex programming problems for which many efficient codes are available. Third, these convex programs differ from one another in only a small subset of their data. As a result, to speed their solution times, an optimal solution to one problem can be used as a starting solution for solving the next problem.

The content of this article is as follows. In Sect. 2 we present the problem equivalent to problem (P) that the algorithm solves. In Sect. 3 we explain the operations of the algorithm and give a precise algorithm statement. Convergence properties of the algorithm and computational considerations for implementing the algorithm are given in Sect. 4. Section 5 reports the solution of a sample problem using the algorithm. Some concluding remarks are given in Sect. 6. For brevity, proofs have been omitted. These proofs can be found in [16].

2 Problem Reformulation

To globally solve problem (P), the branch-and-bound algorithm globally solves a problem (P1) equivalent to problem (P). To help to understand the derivation of problem (P1), we will need the following definition.

Definition 2.1 Let $w : C \to \mathbb{R}$, where $C \subseteq \mathbb{R}^n$. Then w is called a semistrictly quasiconcave function on C when

$$w(x^2) > w(x^1) \quad \text{implies that } w[\lambda x^1 + (1-\lambda)x^2] > w(x^1),$$

for all $x^2, x^2 \in C$ and λ such that $0 < \lambda < 1$.

Let $I = \{1, 2, \ldots, p\}$. For each $i \in I$, let

$$U_i^0 = \max_{x \in X}[\sqrt{f_i(x)}][g_i(x)].$$

Choose any $i \in I$. It is easy to show that, since $f_i(x)$ is concave on X, $c_i(x) = \sqrt{f_i(x)}$ is also concave on X. From p. 163 of Avriel et al. [15], this implies that $q_i(x) = [\sqrt{f_i(x)}][g_i(x)]$ is a semistrictly quasiconcave function on X. It is known (see, e.g., p. 88 of Avriel et al. [15]) that every local maximum of a semistrictly quasiconcave function over a convex set is also a global maximum. Therefore, U_i^0 can be found by any of a number of convex programming algorithms.

Let $H^0 = \{u \in \mathbb{R}^p | 0 \le u_i \le U_i^0, i \in I\}$. Then H^0 is a full-dimensional rectangle in $\mathbb{R}^p$. The objective function of problem (P1) is the function $G : H^0 \to \mathbb{R}$ given, for each $u \in H^0$, by

$$G(u) = \max_{x \in X} \left\{ g(x) + \sum_{i \in I} [2u_i \sqrt{f_i(x)} - u_i^2 (1/g_i(x))] \right\}.$$

Although G is generally neither concave nor convex on H^0, by the following result, it is continuous on H^0.

Lemma 2.1 *The function G is continuous on H^0.*

We now define the problem (P1) by

$$\text{(P1)} \quad v1 = \max_{u \in H^0} G(u).$$

Notice by Lemma 2.1 that problem (P1), is well defined and always has a global optimal solution. For any $u \in \mathbb{R}^p$, $u \ge 0$, define the problem (S_u) by

$$(S_u) \quad \max_{x \in X} \left\{ g(x) + \sum_{i \in I} [2u_i \sqrt{f_i(x)} - u_i^2 (1/g_i(x))] \right\}.$$

Then, problems (P) and (P1) are equivalent in the sense of the following result.

Theorem 2.1 *If u^* is a global optimal solution to problem* (P1), *then any point x^* that solves problem (S_u) with $u = u^*$ is a global optimal solution for problem* (P), *and $v1 = G(u^*) = f(x^*) = v$. If x^* is a global optimal solution for problem* (P), *then $(u^*)^T = (u_1^*, u_2^*, \ldots, u_p^*)$ is a global optimal solution for problem* (P1), *where*

$$u_i^* = [\sqrt{f_i(x^*)}][g_i(x^*)], \quad i \in I.$$

The branch-and-bound algorithm to be presented finds a global optimal solution u^* to problem (P1). Given u^*, to recover a global optimal solution for problem (P), by Theorem 2.1, we may solve problem (S_u) with $u = u^*$ for an optimal solution. Notice for any fixed $u \ge 0$ that problem (S_u) is a convex programming problem. This is because for each $i \in I$, $\sqrt{f_i(x)}$ is concave on X and, from Corollary 5.18(b) of Avriel et al. [15], $1/g_i(x)$ is convex on X. Since $H^0 \subseteq \{u \in \mathbb{R}^p | u \ge 0\}$, it follows that given a global optimal solution u^* to problem (P1), a global optimal solution for problem (P) can be recovered by solving the convex program (S_u) with $u = u^*$.

Although problem (P1) is a global optimization problem, we can state the following simple necessary condition for a global optimal solution to this problem.

Proposition 2.1 *If u^* is a global optimal solution to problem* (P1), *then $u_i^* > 0$ for all $i \in I$.*

3 Algorithm

To globally solve problem (P1), the algorithm to be presented uses a branch-and-bound approach. There are three fundamental processes in the algorithm, a branching process, an upper bounding process, and a lower bounding process.

The algorithm performs a branching process in $\mathbb{R}^p$ that iteratively subdivides the p-dimensional rectangle H^0 of problem (P1) into smaller rectangles that are also of dimension p. This process helps the algorithm identify a location in H^0 of a point that is a global optimal solution for problem (P1). At each stage of the process, the subdivision yields a more refined partition (Horst and Tuy [17]) of a portion of H^0 that is guaranteed to contain a global optimal solution. The initial partition Q^0 consists simply of $\{H^0\}$.

During a typical iteration k of the algorithm, $k \geq 1$, a rectangle H^{k-1} available from iteration $k-1$ is subdivided into two p-dimensional rectangles by a process called bisection of ratio α where α is a prechosen parameter that satisfies $0.0 < \alpha \leq 0.5$. Let $H^{k-1} = \{u \in \mathbb{R}^p | L_i^{k-1} \leq u_i \leq U_i^{k-1}, i \in I\}$, where $L_i^{k-1} < U_i^{k-1}$ for all $i \in I$, and let α satisfy $0.0 < \alpha \leq 0.5$. The procedure for forming a bisection of ratio α of H^{k-1} into two subrectangles H_1^{k-1} and H_2^{k-1} can be described as follows (Tuy [3]).

Step 1. Let $U_j^{k-1} - L_j^{k-1} = \max_{i \in I}\{U_i^{k-1} - L_i^{k-1}\}$.
Step 2. Let v_j satisfy

$$\min\{v_j - L_j^{k-1}, U_j^{k-1} - v_j\} = \alpha(U_j^{k-1} - L_j^{k-1}).$$

Step 3. Let

$$H_1^{k-1} = \{u \in \mathbb{R}^p | L_j^{k-1} \leq u_j \leq v_j, L_i^{k-1} \leq u_i \leq U_i^{k-1}, i \neq j\},$$
$$H_2^{k-1} = \{u \in \mathbb{R}^p | v_j \leq u_j \leq U_j^{k-1}, L_i^{k-1} \leq u_i \leq U_i^{k-1}, i \neq j\}.$$

The new partition Q^k of the portion of H^0 remaining under consideration is then given by

$$Q^k = Q^{k-1} \backslash \{H^{k-1}\} \cup \{H_1^{k-1}, H_2^{k-1}\}.$$

The second fundamental process of the algorithm is the upper bounding process. For each rectangle $H \subseteq \mathbb{R}^p$ created by the branching process, this process gives an upper bound $\mathrm{UB}(H)$ for the optimal value $v(H)$ of the problem.

$$(\mathrm{P1}(H)) \quad \max_{u \in H} G(u).$$

For each rectangle H created by the branching process, $\mathrm{UB}(H)$ is found by solving a single convex program $\mathrm{PR1}(H)$. To derive this convex program, we first need to rewrite the function $G : H^0 \to \mathbb{R}$. Toward this end, for each $i \in I$, let

$$\bar{t}_i = \max_{x \in X} \sqrt{f_i(x)},$$

and let $\underline{s}_i$ satisfy

$$0 < \underline{s}_i \le \min_{x \in X} g_i(x).$$

As noted previously, for each $i \in I, c_i(x) = \sqrt{f_i(x)}$ is a concave function on X. Therefore, for each $i \in I, \bar{t}_i$ can be found by solving a convex programming problem. For each $i \in I, \underline{s}_i$ can be chosen to be a sufficiently small positive number or by methods that we will discuss in Sect. 4.

Consider now the function $F : H^0 \to \mathbb{R}$ which is given for any $u \in H^0$ by

$$\begin{aligned}
(\mathrm{P}(u)) \quad F(u) = \max\ & g(x) + \sum_{i \in I}[2u_i t_i - u_i^2(1/s_i)],\\
\text{s.t.}\quad & t_i - \sqrt{f_i(x)} \le 0, \quad i \in I,\\
& s_i - g_i(x) \le 0, \quad i \in I,\\
& 0 \le t_i \le \bar{t}_i, \quad i \in I,\\
& \underline{s}_i \le s_i, \quad i \in I,\\
& x \in X.
\end{aligned}$$

Notice that for each $u \in H^0$, the objective function in the problem $\mathrm{P}(u)$ defining $F(u)$ is continuous over the nonempty, compact feasible region of the problem. Therefore, $F : H^0 \to \mathbb{R}$ is well defined.

Lemma 3.1 *For each $u \in H^0, G(u) = F(u)$. In addition, if $u \in H^0$ and (x^*, t^*, s^*) is an optimal solution to problem* $\mathrm{P}(u)$, *then* $G(u) = g(x^*) + \sum_{i \in I}[2u_i\sqrt{f_i(x^*)} - u_i^2(1/g_i(x^*))]$.

Let $H = \{u \in \mathbb{R}^p | L \le u \le U\}$ be a typical rectangle created by the branching process, where $L, U \in \mathbb{R}^p$ and $0 \le L_i < U_i$ for all $i \in I$. Then by Lemma 3.1, problems $\mathrm{P1}(H)$ and $\mathrm{PE1}(H)$ have the same optimal value $v(H)$, where problem $\mathrm{PE1}(H)$ is given by

$$\begin{aligned}
(\mathrm{PE1}(H)) \quad \max \quad & g(x) + \sum_{i \in I}[2u_i t_i - u_i^2(1/s_i)],\\
\text{s.t.} \quad & t_i - \sqrt{f_i(x)} \le 0, \quad i \in I, && \text{(1a)}\\
& s_i - g_i(x) \le 0, \quad i \in I, && \text{(1b)}\\
& 0 \le t_i \le \bar{t}_i, \quad i \in I, && \text{(1c)}\\
& \underline{s}_i \le s_i, \quad i \in I, && \text{(1d)}\\
& L_i \le u_i \le U_i, \quad i \in I, && \text{(1e)}\\
& x \in X. && \text{(1f)}
\end{aligned}$$

To find an upper bound $\mathrm{UB}(H)$ for $v(H)$, the algorithm solves a relaxed version of problem $\mathrm{PE1}(H)$ which is identical to problem $\mathrm{PE1}(H)$, except that in the objective function of the relaxed problem, for each $i \in I$, a concave envelope of $h_i(u_i, t_i) \triangleq u_i t_i$ is substituted for $h_i(u_i, t_i)$.

Definition 3.1 (Horst and Tuy [17]) Let $M \subseteq \mathbb{R}^q$ be a compact convex set, and let $f : M \to \mathbb{R}$ be upper semicontinuous on M. Then $f^M : M \to \mathbb{R}$ is called the concave envelope of f on M when

(i) $f^M(x)$ is a concave function on M.
(ii) $f^M(x) \geq f(x)$ for all $x \in M$.
(iii) There is no function $w(x)$ satisfying (i) and (ii) such that $w(\bar{x}) < f^M(\bar{x})$ for some point $\bar{x} \in M$.

For each $i \in I$, let

$$M_i = \{(u_i, t_i) \in \mathbb{R}^2 | L_i \leq u_i \leq U_i, 0 \leq t_i \leq \bar{t}_i\},$$

where $H = \{u \in \mathbb{R}^p | L_i \leq u_i \leq U_i, i \in I\}$ is a typical rectangle created by the branching process. Then, for each $i \in I, 0 \leq L_i < U_i$, and, from Al-Khayyal and Falk [18], the concave envelope of $h_i^{M_i}$ of h_i on M_i is given, for each $(u_i, t_i) \in M_i$ by

$$h_i^{M_i}(u_i, t_i) = \min\{\bar{t}_i u_i + L_i t_i - \bar{t}_i L_i, U_i t_i\}. \tag{2}$$

The upper bound UB(H) for $v(H)$ used in the algorithm is given by

$$\begin{aligned} \text{UB}(H) = \max g(x) + \sum_{i \in I} [2h_i^{M_i}(u_i, t_i) - u_i^2(1/s_i)], \\ \text{s.t.} \quad \text{(1a)–(1f)}, \end{aligned}$$

where, for each $i \in I$, $h_i^{M_i}(u_i, t_i)$ is given by (2). To calculate UB(H), the algorithm finds an optimal solution and the optimal value to the problem PR1(H) given by

$$(\text{PR1}(H)) \quad \text{UB}(H) = \max g(x) + \sum_{i \in I} [2r_i - u_i^2(1/s_i)],$$

$$\begin{aligned} \text{s.t.} \quad & r_i \leq \bar{t}_i u_i + L_i t_i - \bar{t}_i L_i, \quad i \in I, && \text{(3a)} \\ & r_i \leq U_i t_i, \quad i \in I, && \text{(3b)} \\ & t_i - \sqrt{f_i(x)} \leq 0, \quad i \in I, \\ & s_i - g_i(x) \leq 0, \quad i \in I, \\ & 0 \leq t_i \leq \bar{t}_i, \quad i \in I, \\ & \underline{s}_i \leq s_i, \quad i \in I, \\ & L_i \leq u_i \leq U_i, \quad i \in I, && \text{(3c)} \\ & r_i \geq 0, \quad i \in I, \\ & x \in X. \end{aligned}$$

Notice that the optimal value $\mathrm{UB}(H)$ of problem $\mathrm{PR1}(H)$ satisfies $\mathrm{UB}(H) \geq v(H)$. This is because problems $\mathrm{P1}(H)$ and $\mathrm{PE1}(H)$ have the same optimal value $v(H)$, and because, by using Definition 3.1 and (2), it is easily verified that problem $\mathrm{PR1}(H)$ is equivalent to the problem obtained from problem $\mathrm{PE1}(H)$ by substituting the concave envelope $h_i^{M_i}(u_i, t_i)$ of $h_i(u_i, t_i)$ on M_i for $h_i(u_i, t_i)$ in the objective function of problem $\mathrm{PE1}(H)$ for all $i \in I$. It is also easy to see that the feasible region of problem $\mathrm{PR1}(H)$ is a nonempty, compact set. Since the objective function of problem $\mathrm{PR1}(H)$ is continuous over this set, problem $\mathrm{PR1}(H)$ always has an optimal solution. Finally, it is easy to see that problem $\mathrm{PR1}(H)$ involves the maximization of a concave function over a convex set. This is because, for each $i \in I$, as observed earlier, the function $c_i(x) = \sqrt{f_i(x)}$ is concave on X and, from p. 119 of Bazaraa, Sherali and Shetty [19], the function

$$d_i(u_i, s_i) = u_i^2/s_i$$

is convex over the feasible region of the problem. Therefore, problem $\mathrm{PR1}(H)$ is an ordinary convex program and always has an optimal solution.

During each iteration $k \geq 0$, the upper bounding process computes an upper bound for the optimal value $v1$ of problem (P1). For each $k \geq 0$, this upper bound UB_k is given by

$$\mathrm{UB}_k = \max\{\mathrm{UB}(H) | H \in Q^k\}.$$

The lower bounding process is the third fundamental process of the branch and bound algorithm. In each iteration of the algorithm, this process finds a lower bound for $v1$. For each $k \geq 0$, this lower bound LB_k is given by

$$\mathrm{LB}_k = G(\hat{u}^k),$$

where $\hat{u}^k$ is the incumbent feasible solution for problem (P1); i.e., among all of the optimal solutions (r, s, t, u, x) for problems of the form $\mathrm{PR1}(H)$ found through iteration k, $u = \hat{u}^k$ achieves the largest value of G.

Based upon the results and algorithmic processes discussed in this section, the branch-and-bound algorithm for globally solving problem (P1) may be stated as follows.

3.1 Branch-and-Bound Algorithm

Initialization:

(i) Choose $\alpha \in \mathbb{R}$ such that $0 < \alpha \leq 1/2$.
(ii) Determine an optimal solution $(r^0, s^0, t^0, u^0, x^0)$ and the optimal value $\mathrm{UB}(H^0)$ to problem $\mathrm{PR1}(H^0)$. Set $\mathrm{UB}_0 = \mathrm{UB}(H^0)$, $\mathrm{LB}_0 = G(u^0)$, and $\hat{u}^0 = u^0$.
(iii) Set $Q^0 = \{H^0\}$ and $k = 1$, and go to iteration k.

Iteration k:

(i) If $\mathrm{LB}_{k-1} = \mathrm{UB}_{k-1}$, then terminate. The point $\hat{u}^{k-1}$ is a global optimal solution for problem (P1) and $v1 = \mathrm{LB}_{k-1}$. If $\mathrm{LB}_{k-1} \neq \mathrm{UB}_{k-1}$, continue.

(ii) Subdivide H^{k-1} into two rectangles H_1^{k-1} and H_2^{k-1} via the bisection of ratio α procedure.
(iii) For each $i = 1, 2$, find an optimal solution $(r_i^{k-1}, s_i^{k-1}, t_i^{k-1}, u_i^{k-1}, x_i^{k-1})$ and the optimal value $\mathrm{UB}(H_i^{k-1})$ to problem $\mathrm{PR1}(H_i^{k-1})$.
(iv) Set

$$\mathrm{LB}_k = \max\{G(\hat{u}^{k-1}), G(u_1^{k-1}), G(u_2^{k-1})\}$$

and choose $\hat{u}^k$ so that $\mathrm{LB}_k = G(\hat{u}^k)$.
(v) Set $Q^k = Q^{k-1} \backslash \{H^{k-1}\} \cup \{H_1^{k-1}, H_2^{k-1})\}$.
(vi) Delete from Q^k all rectangles H such that $\mathrm{UB}(H) \le \mathrm{LB}_k$.
(vii) If $Q^k = \emptyset$, set $\mathrm{UB}_k = \mathrm{LB}_k$, set $k = k + 1$ and go to iteration k. Otherwise, set

$$\mathrm{UB}_k = \max\{\mathrm{UB}(H) | H \in Q^k\}.$$

(viii) Choose a rectangle $H^k \in Q^k$ such that

$$\mathrm{UB}(H^k) = \mathrm{UB}_k.$$

(ix) Set $k = k + 1$ and go to iteration k.

For each $k \ge 1$, step (vi) of iteration k executes the fathoming process; i.e., it eliminates from further consideration rectangles H that need not be explored further for a global optimal solution to problem (P1).

4 Convergence and Computational Considerations

By construction, when the branch-and-bound algorithm of Sect. 3 is finite, it terminates for some $k \ge 1$ in step (i) of iteration k with a global optimal solution $\hat{u}^{k-1}$ and the optimal value LB_{k-1} to problem (P1) at hand. It is also possible for the algorithm to be infinite. In the latter case, the following result is a key to the convergence of the algorithm.

Theorem 4.1 *Suppose that the branch-and-bound algorithm is infinite. Let $\{H^q\}$ be an infinite subsequence of $\{H^k\}$ generated by the algorithm such that $H^{q'} \subseteq H^q$ for each rectangle H^q and its successor $H^{q'}$ in the subsequence. Let $\bigcap_q H^q = \lim_q H^q = \{u^*\}$. Then u^* is a global optimal solution for problem* (P1).

Remark 4.1 Note that when the algorithm is infinite, it follows from Tuy [3] that a subsequence $\{H^t\}$ of $\{H^k\}$ exists such that $H^{t'} \subseteq H^t$ for each rectangle H^t and its successor $H^{t'}$ in the subsequence, and, from Corollary 5.4 of Tuy [3], for any such subsequence $\{H^t\}$, $\bigcap_t H^t = \lim_t H^t = \{\tilde{u}\}$ for some $\tilde{u} \in \mathbb{R}^p$. Therefore, whenever the branch-and-bound algorithm is infinite, a subsequence $\{H^q\}$ as in the statement of Theorem 4.1 can be identified.

From Theorem 4.1 and its proof [16], we obtain the convergence properties of the algorithm given in the next result.

Corollary 4.1 *Suppose that the branch-and-bound algorithm is infinite. Then*

(i) *Each accumulation point of* $\{\hat{u}^k\}$ *is a global optimal solution for problem* (P1).
(ii) *Each accumulation point of* $\{u^k\}$ *is a global optimal solution for problem* (P1) *where, for each* k $(r^k, s^k, t^k, u^k, x^k)$ *denotes the optimal solution found by the algorithm to problem* $\mathrm{PR1}(H^k)$.
(iii) $\lim_{k\to\infty} \mathrm{LB}_k = \lim_{k\to\infty} \mathrm{UB}_k = v1$.

In practice, there are various computational considerations that may be taken into account when implementing the algorithm. To close this section, we discuss four of these.

First, users of the algorithm may want to employ a revised termination rule to guarantee finiteness of the algorithm. Let $\epsilon > 0$ be a prechosen, relatively-small real number. A point $\hat{u} \in \mathbb{R}^p$ is called an ϵ-global optimal solution and $G(\hat{u})$ is called an ϵ-global optimal value for problem (P1) when $\hat{u} \in H^0$ and $G(\hat{u}) + \epsilon \geq G(u)$ for all $u \in H^0$. An ϵ-global optimal solution for problem (P) is defined similarly. By part (iii) of Corollary 4.1, if for each $k \geq 1$, we replace step (i) of iteration k of the algorithm with

"(i) If $\mathrm{UB}_{k-1} - \mathrm{LB}_{k-1} \leq \epsilon$, then terminate. The point $\hat{u}^{k-1}$ is an ϵ-global optimal solution for problem (P1). Otherwise, continue",

then the algorithm will be finite and will remain valid. In this case, to recover an ϵ-global optimal solution for problem (P), the following result can be used.

Proposition 4.1 *Assume that* $\hat{u}$ *is an* ϵ*-global optimal solution for problem* (P1). *Then any point* $\hat{x}$ *that solves problem* (S_u) *with* $u = \hat{u}$ *is an* ϵ*-global optimal solution for problem* (P).

Recall from Sect. 2 for each fixed $u \geq 0$ that problem (S_u) is a convex programming problem. Therefore, from the discussion above and Proposition 4.1, if the branch-and-bound algorithm with the modified termination rule is used to find an ϵ-global optimal solution to problem (P1), the algorithm will be finite, and an ϵ-global optimal to problem (P) can be recovered by solving a single convex program.

A second computational consideration concerns the definition of H^0. For each $i \in I$, let

$$L_i^0 = \min_{x\in X} [\sqrt{f_i(x)}][g_i(x)]. \tag{4}$$

As noted earlier, for each $i \in I$, $q_i(x) = [\sqrt{f_i(x)}][g_i(x)]$ is a semistrictly quasiconcave function on X. From Avriel et al. [15], this implies that $q_i(x)$ is quasiconcave on X for each $i \in I$. Therefore, for each $i \in I$, finding L_i^0 requires solving a global optimization problem. However, in some cases, for at least some elements $i \in I$, computing L_i^0 may not require too much effort. For example, if X is a polytope, then, for each $i \in I$, $q_i(x)$ attains its minimum over X at one of the extreme points of X, and several relatively-efficient concave minimization algorithms are available for computing L_i^0 (see, e.g., Benson [20], Horst and Tuy [17]).

When $L_i^0, i \in I$, as defined by (4), can be computed, then, to globally solve problem (P), a user of the branch-and-bound algorithm may apply the algorithm to problem (P1) with H^0 redefined as

$$H^0 = \{u \in \mathbb{R}^p | L_i^0 \le u \le U_i^0, i \in I\}.$$

Since $L_i^0 > 0, i \in I$, this approach shrinks the domain of problem (P1), and the algorithm may therefore converge more rapidly.

A third computational consideration concerns finding a value for $\underline{s}_i$ for each $i \in I$ such that

$$0 < \underline{s}_i \le \min_{x \in X} g_i(x). \tag{5}$$

Let $i \in I$. If $g_i(x)$ happens to be an affine function, then a user of the algorithm may set

$$\underline{s}_i = \min_{x \in X} g_i(x), \tag{6}$$

since finding the minimum value of $g_i(x)$ over X amounts to solving a single convex program. In this case, the user may also substitute the constraint

$$s_i - g_i(x) = 0$$

for the constraint

$$s_i - g_i(x) \le 0$$

in the definition of problem PR1(H) for each rectangle H created by the algorithm. If $g_i(x)$ is not affine, then at least three possibilities exist for finding a value for $\underline{s}_i$ that satisfies (5). First, $\underline{s}_i$ may simply be chosen to be a positive number of much smaller magnitude than the values that $g_i(x)$ takes over X. Second, $\underline{s}_i$ may be chosen according to (6) by applying an appropriate concave minimization algorithm (see, e.g., [20] or Horst and Tuy [17]) to the problem

$$\min_{x \in X} g_i(x). \tag{7}$$

Third, a value for $\underline{s}_i$ may be found that satisfies (5) by initiating an outer approximation algorithm (see Horst and Tuy [17]) on (7) and terminating the algorithm prematurely. This approach is viable because each iteration of an outer approximation algorithm for solving (7) gives a lower bound for the optimal value of (7) and the sequence of lower bounds thereby generated is nondecreasing.

A fourth computational consideration concerns the solution of the convex programming problems PR1(H) for rectangles H generated by the algorithm. For each $k \ge 1$, in step (ii) of iteration k, a parent rectangle H^{k-1} is subdivided into two offspring rectangles H_1^{k-1} and H_2^{k-1} by the bisection of ratio α procedure. From the steps of this procedure and the definition of problem PR1(H), it is evident for each $i = 1, 2$ that problems PR1(H) and PR1(H_i^{k-1}) differ, at most, in three of the coefficients of the linear constraints (3a)–(3c). Therefore, an optimal solution to one problem can be used to good advantage as a starting solution for the next problem. A similar comment applies to the solutions of the convex programming problems (S_u). Recall that these problems must be solved during the lower bounding process in order to evaluate $G(u)$ as feasible solutions u for problem (P1) are found.

5 Sample Problem

With the aid of LINGO (LINDO Systems [21]), we have used the branch-and-bound algorithm to solve several sample problems. Below we describe one of these sample problems and solution results. In this problem, LINGO was used to solve the convex subproblems specified by the algorithm. The ϵ-global optimal solution and ϵ-global optimal value are given to the nearest ten-thousandth.

Example 5.1 This example is in the form of problem (P) and is given by

$$v = \max\, (x_1 - x_2) + (5 - 0.25x_1^2)(0.125x_2 + 1) + (0.25x_1 + 1)(4 - 0.125x_2^2),$$
$$\text{s.t.}\quad 5x_1 - 8x_2 \geq -24,$$
$$5x_1 + 8x_2 \leq 44,$$
$$6x_1 - 3x_2 \leq 15,$$
$$4x_1 + 5x_2 \geq 10,$$
$$x_1 \geq 0.$$

Let X^1 denote the feasible region of this problem. To solve this problem, we wrote it in the form of problem (P1). In this example, the set of extreme points of X^1 was easy to find, so we were able to compute $L_i^0, i = 1, 2$, as defined by (4) by extreme point search. The resulting problem (P1) in $p = 2$ variables is

$$v1 = \max\ G(u_1, u_2),$$
$$\text{s.t.}\quad 1.3750 \leq u_1 \leq 3.1671,$$
$$2.1337 \leq u_2 \leq 5.1567,$$

where, for each feasible solution (u_1, u_2) to the problem,

$$G(u_1, u_2) = \max_{x \in X^1} [(x_1 - x_2) + 2u_1\sqrt{-0.25x_1^2 + 5} - (u_1^2/(0.125x_2 + 1))$$
$$+ 2u_2\sqrt{0.25x_1 + 1} - (u_2^2/(-0.125x_2^2 + 4))].$$

With $\epsilon = 0.05$, the algorithm found the ϵ-global optimal solution $(\hat{u})^T = (1.8695, 5.1048)$ with ϵ-global optimal value 12.4373 after 77 iterations. The initial upper bound was 13.3286, and the ϵ-global optimal value was found during iteration number 67. Solving problem (S_u) with $u = \hat{u}$ yields the ϵ-global optimal solution $(\hat{x})^T = (2.5000, 0.0000)$ to the original formulation of the problem, with ϵ-global optimal value 12.4373.

6 Concluding Remarks

In this article, we have introduced the generalized concave multiplicative programming problem (P) for the first time. Problem (P) has numerous practical applications in economics, statistics, finance, engineering design, and in many other areas. Since this problem is a global optimization problem, it is particularly challenging to solve. We have also proposed what is apparently the first algorithm for globally solving problem (P). The algorithm implements a rectangular, branch-and-bound search and has several potentially-attractive characteristics. First, the branch-and-bound search takes place in a space which typically has much smaller dimension than the space of the problem's decision variables. Second, all subproblems that must be solved to implement the algorithm are convex programming problems, each of which is guaranteed to have an optimal solution. Third, to speed the solution of these convex programs, an optimal solution to one problem can be used as a starting solution to the next problem. It is hoped that this algorithm will offer a potentially useful tool for globally solving the generalized concave multiplicative programming problem and its applications.

References

1. Konno, H., Kuno, T., Yajima, Y.: Global minimization of a generalized convex multiplicative function. J. Glob. Optim. **4**, 47–62 (1994)
2. Cambini, R., Sodini, C.: Decomposition methods for solving nonconvex quadratic programs via branch and bound. J. Glob. Optim. **33**, 313–336 (2006)
3. Tuy, H.: Convex Analysis and Global Optimization. Kluwer Academic, Dordrecht (1998)
4. Pardalos, P.M., Rosen, J.B.: Constrained Global Optimization: Algorithms and Applications. Springer, Berlin (1987)
5. Floudas, C.A., Visweswaran, V.: Quadratic optimization. In: Horst, R., Pardalos, P.M. (eds.) Handbook of Global Optimization, pp. 217–269. Kluwer Academic, Dordrecht (1995)
6. Horst, R., Pardalos, P.M., Thoai, N.V.: Introduction to Global Optimization. Kluwer Academic, Dordrecht (1995)
7. Watanabe, H.: IC layout generation and compaction using mathematical programming. Ph.D. Thesis, University of Rochester (1984)
8. Konno, H., Yajima, Y.: Solving rank two bilinear programs by parametric simplex algorithms. Institute of Human and Social Sciences Working Paper IHSS 90–17, Tokyo Institute of Technology (1990)
9. Konno, H., Thach, P.T., Tuy, H.: Optimization on Low Rank Nonconvex Structures. Kluwer Academic, Dordrecht (1997)
10. Mangasarian, O.L.: Equilibrium points of bimatrix games. SIAM J. **12**, 778–780 (1964)
11. Freize, A.M.: A bilinear programming formulation of the 3-dimensional assignment problem. Math. Program. **7**, 376–379 (1979)
12. Falk, J.E.: A linear max-min problem. Math. Program. **5**, 169–188 (1973)
13. Raghavachari, M.: On connections between zero-one integer programming and concave programming under linear constraints. Oper. Res. **17**, 680–684 (1969)
14. Nemhauser, G.L., Wolsey, L.A.: Combinatorial Optimization. Wiley, New York (1998)
15. Avriel, M., Diewart, W.E., Schaible, S., Zang, I.: Generalized Concavity. Plenum, New York (1988)

16. Benson, H.P.: Global maximization of a generalized concave multiplicative function. Warrington College of Business Administration Working Paper, University of Florida (2007)
17. Horst, R., Tuy, H.: Global Optimization—Deterministic Approaches. Springer, Berlin (1993)
18. Al-Khayyal, F.A., Falk, J.E.: Jointly constrained biconvex programming. Math. Oper. Res. **8**, 273–286 (1983)
19. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming—Theory and Algorithms. Wiley, New York (1993)
20. Benson, H.P.: Deterministic algorithms for constrained concave minimization: a unified critical survey. Nav. Res. Logist. **43**, 765–795 (1996)
21. LINDO SYSTEMS: Optimization modeling with LINGO. LINDO Systems, Chicago (2000)