Journal of Global Optimization **22:** 343–364, 2002.
© 2002 *Kluwer Academic Publishers. Printed in the Netherlands.*

Using concave envelopes to globally solve the nonlinear sum of ratios problem

HAROLD P. BENSON
University of Florida, Warrington College of Business Administration, Gainesville, FL, 32611, USA (E-mail: benson@dale.cba.ufl.edu)

Abstract. This article presents a branch and bound algorithm for globally solving the nonlinear sum of ratios problem (P). The algorithm works by globally solving a sum of ratios problem that is equivalent to problem (P). In the algorithm, upper bounds are computed by maximizing concave envelopes of a sum of ratios function over intersections of the feasible region of the equivalent problem with rectangular sets. The rectangular sets are systematically subdivided as the branch and bound search proceeds. Two versions of the algorithm, with convergence results, are presented. Computational advantages of these algorithms are indicated, and some computational results are given that were obtained by globally solving some sample problems with one of these algorithms.

Key words: Sum of ratios problem; Global optimization; Concave envelope; Fractional functions

1. Introduction

Consider the nonlinear sum of ratios problem

$$\text{(P)}\quad v = \max h(x) \overset{\triangle}{=} \sum_{i=1}^{p} \frac{n_i(x)}{d_i(x)}\,, \text{ subject to } x \in X\,,$$

where $p \geqslant 2$, $n_i: \mathcal{R}^n \to \mathcal{R}$ is a finite, concave function for each $i = 1, 2, \dots, p$, $d_i: \mathcal{R}^n \to \mathcal{R}$ is a finite, convex function for each $i = 1, 2, \dots, p$, and X is a compact, convex (possibly empty) set in $\mathcal{R}^n$. We assume also that for each $i = 1, 2, \dots, p$, $l_i \leqslant n_i(x) \leqslant u_i$ and $L_i \leqslant d_i(x) \leqslant U_i$ for all $x \in X$, where l_i, u_i, L_i and U_i are positive scalars that satisfy $l_i \leqslant u_i$ and $L_i \leqslant U_i$, $i = 1, 2, \dots, p$. Notice that if $L_i = U_i$, $i = 1, 2, \dots, p$, then problem (P) can be globally solved by any of a number of convex programming methods. Therefore, we assume for at least one $i \in \{1, 2, \dots, p\}$ that $L_i < U_i$. Sums of ratios problems have attracted the interest of practitioners and researchers for at least 30 years. During the past 10 years, interest in these problems has been especially intense. In part, this is because, from a practical point of view, sums of ratios problems have many important applications. Included among these are applications in areas such as transportation planning, government contracting, economics and finance [1, 7, 18, 20, 27, 28]. In these applications, the number of ratios summed in the objective function is usually less than four or five. Another reason for the strong interest in sums of ratios problems is that, from a research point of view, these problems pose significant theoretical and computational challenges. This is mainly because these problems are global optimization problems, i.e., they are known to generally possess multiple local optima that are not globally optimal [29].

During the past 35 years there has been significant progress in the development of deterministic algorithms for finding global optimal solutions to global optimization problems [14]. During the earlier years of this research, global optimization algorithms were developed for certain general classes of problems, including concave minimization problems, d.c. programming problems and others [13]. More recently, algorithms for special cases have been developed. Included among these are a number of algorithms for globally solving sums of ratios problems. Most of these algorithms, however, are limited to the linear case, i.e., to the case where $n_i(x)$ and $d_i(x)$ are affine functions for each $i = 1, 2, \ldots, p$, and X is a polyhedron [6, 9, 16, 17, 21–23]. There are at least four notable exceptions, however. In [19], Konno et al. develop an algorithm for sums of ratios problems in which the numerators and denominators are affine functions and the feasible region is a compact, convex set. In addition, three algorithms have been proposed for nonlinear sums of ratios problems. The first, by Quesada and Grossmann [25], applies underestimating functions in a branch and bound framework to a problem that includes problem (P) as a special case. The second algorithm, by Dur et al. [8], converts problem (P) to a parametric nonlinear program which is solved by branch and bound. More recently, Freund and Jarre [11] have presented an algorithm for solving problem (P) that works by underestimating an associated optimal value function.

One of the classic ways of solving a global optimization problem is to use a branch and bound approach wherein bounds are obtained by solving convex or linear programming subproblems that maximize concave envelopes or minimize convex envelopes of the objective function over subregions of the feasible region [14]. This approach has been successfully applied to a number of global optimization problems. Included among these, for instance, are the concave minimization problem [2], separable concave minimization problems [5, 14], bilinear programming problems [12] and concave quadratic minimization problems [15, 26]. However, with one notable exception, this approach has not been applied to the solution of sums of ratios problems. The exception is an algorithm by Kuno [23] for the linear sum of ratios problem. It can be shown that this algorithm uses concave envelopes over trapezoids to generate upper bounds [3], although Kuno did not stop to show this in his article.

In this article, we present a branch and bound algorithm for globally solving problem (P). The algorithm works by solving a sum of ratios problem (P_H) that is equivalent to problem (P), where H is a rectangle in $\mathcal{R}^{2p}$. In the algorithm, upper bounds are obtained by maximizing concave envelopes of the objective function of problem (P_H) over subregions of its feasible region that are obtained by repeatedly subviding H into subrectangles that belong to $\mathcal{R}^{2p}$. Two versions of the algorithm are presented. In one version, the rectangles are subdivided by a method called *bisection of ratio* α [30]. In the second version, this subdivision is accomplished by a method called *ω-subdivision* [30]. In both versions, one of the important advantages is the somewhat surprising fact that, although the branch and bound search involves rectangles defined in a space of dimension $2p$, branching takes place in a space of only dimension p. Another key advantage is that the upper bounding subproblems are convex programming problems that differ from one another only in the coefficients of certain linear constraints and in the bounds that describe their associated rectangles. The algorithms in this article were motivated by the seminal works of Horst and Tuy [14] and Tuy [30] on using branch and bound for global optimization, and by the recent examples of this use by Kuno [23] for linear sums of ratios problems.

The organization of this article is as follows. In Section 2, preliminary results of two types are presented. First, the problem (P_H) equivalent to the nonlinear sum of ratios problem (P) is derived. Second, concave and convex envelope functions for the objective function of problem (P_H) are given. In Section 3, a prototype branch and bound algorithm for globally solving problem (P_H) is presented, and properties of this algorithm are delineated. Two implementations of this algorithm, with convergence results, are presented in Section 4. Section 5 discusses some computational considerations and reports some computational results obtained by solving some example problems with one of the implementations of the prototype algorithm. Concluding remarks are given in the last section.

2. Preliminaries

Let $H = \{(t, s) \in \mathcal{R}^{2p} \mid l_i \leqslant t_i \leqslant u_i,\ L_i \leqslant s_i \leqslant U_i,\ i = 1, 2, \ldots, p\}$ and consider the sum of ratios problem

$$
\begin{aligned}
(P_H)\quad v_H = \max \sum_{i=1}^{p} \frac{t_i}{s_i}& \\
\text{s.t.}\quad & n_i(x) - t_i \geqslant 0,\ i = 1, 2, \ldots, p\ , \\
& -d_i(x) + s_i \geqslant 0,\ i = 1, 2, \ldots, p\ , \\
& x \in X,\ (t, s) \in H\ .
\end{aligned}
$$

Notice that the feasible region $Z(H) \stackrel{\triangle}{=} \{(x, t, s) \in \mathcal{R}^{n+2p} \mid n_i(x) - t_i \geqslant 0,\quad i = 1, 2, \ldots, p,\ -d_i(x) + s_i \geqslant 0,\ i = 1, 2, \ldots, p,\ x \in X,\ (t, s) \in H\}$ of problem (P_H) is a compact, convex set, and that $Z(H) = \emptyset$ if and only if $X = \emptyset$. To globally solve problem (P), the branch and bound algorithm will globally solve problem (P_H). The validity of this approach follows from the following result.

THEOREM 1. *If (x^*, t^*, s^*) is a global optimal solution for problem (P_H), then $t_i^* = n_i(x^*)$, $s_i^* = d_i(x^*)$, $i = 1, 2, \ldots, p$, and x^* is a global optimal solution for problem (P). Conversely, if x^* is a global optimal solution for problem (P), then*

(x^*, t^*, s^*) is a global optimal solution for problem (P_H), where $t_i^* = n_i(x^*)$, $s_i^* = d_i(x^*)$, $i = 1, 2, \ldots, p$.

Proof. Let (x^*, t^*, s^*) be a global optimal solution for problem (P_H). Then, for each $i = 1, 2, \ldots, p$, $n_i(x^*) \geqslant t_i^* \geqslant l_i > 0$ and $s_i^* \geqslant d_i(x^*) \geqslant L_i > 0$, $i = 1, 2, \ldots, p$. This implies that for each $i = 1, 2, \ldots, p$,

$$\frac{n_i(x^*)}{d_i(x^*)} \geqslant \frac{t_i^*}{s_i^*},$$

so that

$$h(x^*) \geqslant \sum_{i=1}^{p} \frac{t_i^*}{s_i^*}. \tag{1}$$

For each $i = 1, 2, \ldots, p$, let $\hat{t}_i = n_i(x^*)$ and $\hat{s}_i = d_i(x^*)$. Then $(x^*, \hat{t}, \hat{s})$ is a feasible solution for problem (P_H) and, in this problem, has an objective function value equal to $h(x^*)$. Since (x^*, t^*, s^*) is a global optimal solution for problem (P_H), this implies that

$$h(x^*) \leqslant \sum_{i=1}^{p} \frac{t_i^*}{s_i^*}. \tag{2}$$

From (1) and (2),

$$h(x^*) = \sum_{i=1}^{p} \frac{t_i^*}{s_i^*}. \tag{3}$$

By definition of h, since, for each $i = 1, 2, \ldots, p$,

$$\frac{n_i(x^*)}{d_i(x^*)} \geqslant \frac{t_i^*}{s_i^*}, \qquad n_i(x^*) \geqslant t_i^* > 0, \qquad s_i^* \geqq d_i(x^*) > 0,$$

this implies that for each $i = 1, 2, \ldots, p$, $t_i^* = n_i(x^*)$ and $s_i^* = d_i(x^*)$. For any feasible solution x for problem (P), if we set $t_i = n_i(x)$ and $s_i = d_i(x)$, $i = 1, 2, \ldots, p$, then (x, t, s) is a feasible solution for problem (P_H) and, in this problem, has an objective function value equal to $h(x)$. Since (x^*, t^*, s^*) is a global optimal solution for problem (P_H), this implies that for any feasible solution x for problem (P),

$$h(x) \leqslant \sum_{i=1}^{p} \frac{t_i^*}{s_i^*}.$$

From (3), since $x^* \in X$, this implies that x^* is a global optimal solution for problem (P).

To show the converse statement, let x^* be a global optimal solution for problem (P), and let $t_i^* = n_i(x^*)$ and $s_i^* = d_i(x^*)$, $i = 1, 2, \ldots, p$. Then (x^*, t^*, s^*) is a feasible solution with objective function value $h(x^*)$ in problem (P_H). Let (x, t, s) be a feasible solution for problem (P_H). Then, for each $i = 1, 2, \ldots, p$, $n_i(x) \geqslant t_i \geqslant l_i > 0$ and $s_i \geqslant d_i(x) \geqslant L_i > 0$. This implies that for each $i = 1, 2, \ldots, p$,

$$\frac{n_i(x)}{d_i(x)} \geqslant \frac{t_i}{s_i},$$

so that

$$h(x) \geqslant \sum_{i=1}^{p} \frac{t_i}{s_i}. \tag{4}$$

Since $x \in X$ and x^* is a global optimal solution for problem (P), $h(x) \leqslant h(x^*)$. Together with (4), this implies that

$$h(x^*) \geqslant \sum_{i=1}^{p} \frac{t_i}{s_i}. \tag{5}$$

Since $t_i^* = n_i(x^*)$ and $s_i^* = d_i(x^*)$, $i = 1, 2, \ldots, p$,

$$\sum_{i=1}^{p} \frac{t_i^*}{s_i^*} = h(x^*). \tag{6}$$

From (5) and (6),

$$\sum_{i=1}^{p} \frac{t_i^*}{s_i^*} \geqslant \sum_{i=1}^{p} \frac{t_i}{s_i},$$

and the proof is complete. □

Notice that it follows immediately from Theorem 1 that $v = v_H$.

To help compute upper bounds, the branch and bound method to be presented relies upon the concept of a concave envelope, which may be defined as follows.

DEFINITION 1 [14]. Let $M \subseteq \mathscr{R}^q$ be a compact, convex set, and let $f: M \to \mathscr{R}$ be upper semicontinuous on M. Then $f^M: M \to \mathscr{R}$ is called the concave envelope of f on M when

(i) $f^M(x)$ is a concave function on M,
(ii) $f^M(x) \geqslant f(x)$ 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 $\bar{x} \in M$.

The convex envelope of a function f on M is defined in a similar manner. Notice that the concave envelope f^M is the 'tightest' concave function that majorizes f on M. Closed-form formulas for f^M frequently cannot be found. But for the case of a simple fractional function, we have the following result.

THEOREM 2. *Let*

$$RC = \{(x_1, x_2) \in \mathscr{R}^2 \mid \underline{a} \leqslant x_1 \leqslant \bar{a},\ \underline{b} \leqslant x_2 \leqslant \bar{b}\},$$

where $\underline{a}$, $\bar{a}$, $\underline{b}$ *and* $\bar{b}$ *are scalars satisfying* $\underline{a} \leqslant \bar{a}$, $\underline{b} \leqslant \bar{b}$. *For any* $(x_1, x_2) \in \mathscr{R}^2$ *such that* $x_2 \neq 0$, *let* $ff(x_1, x_2) = x_1/x_2$.

(i) *Suppose that either* $\underline{a}, \underline{b} > 0$ *or* $\bar{a}$, $\bar{b} < 0$. *Then the concave envelope* $ff^{RC}: RC \to \mathscr{R}$ *of* ff *on* RC *is given by*

$$ff^{RC}(x_1, x_2) = \min\left\{\frac{1}{\underline{b}}x_1 - \left(\frac{\underline{a}}{\underline{b}\bar{b}}\right)x_2 + \frac{\underline{a}}{\bar{b}},\ \frac{1}{\bar{b}}x_1 - \left(\frac{\bar{a}}{\underline{b}\bar{b}}\right)x_2 + \frac{\bar{a}}{\underline{b}}\right\}.$$

(ii) *Suppose that either* $\bar{a} < 0$, $\underline{b} > 0$ *or* $\underline{a} > 0$, $\bar{b} < 0$. *Then the convex envelope* $ff_{RC}: RC \to \mathscr{R}$ *of* ff *on* RC *is given by*

$$ff_{RC}(x_1, x_2) = \max\left\{\frac{1}{\underline{b}}x_1 - \left(\frac{\bar{a}}{\underline{b}\bar{b}}\right)x_2 + \frac{\bar{a}}{\bar{b}},\ \frac{1}{\bar{b}}x_1 - \left(\frac{\underline{a}}{\underline{b}\bar{b}}\right)x_2 + \frac{\underline{a}}{\underline{b}}\right\}.$$

Proof. This result is essentially shown in Benson [3] as a corollary of a more general result. However, to provide a self-contained presentation, and because this result is central to this article, we provide a direct proof for (i) in the case where $\underline{a}, \underline{b} > 0$. Proofs for the other cases are similar.

Assume that $\underline{a}, \underline{b} > 0$. When $\underline{a} = \bar{a}$, $x_1 = \underline{a} = \bar{a}$ for all $(x_1, x_2) \in RC$, and $ff(x_1, x_2) = \underline{a}/x_2$ for all $(x_1, x_2) \in RC$. Therefore, in this case, $ff(x_1, x_2)$ is a convex function of one variable on a line segment, and the result easily follows from Theorem IV. 7 in [14]. Therefore, assume in the remainder of the proof that $\underline{a} < \bar{a}$. For each $(x_1, x_2) \in RC$, let

$$h_1(x_1, x_2) = \frac{1}{\underline{b}}x_1 - \left(\frac{\underline{a}}{\underline{b}\bar{b}}\right)x_2 + \frac{\underline{a}}{\bar{b}}.$$

$$h_2(x_1, x_2) = \frac{1}{\bar{b}}x_1 - \left(\frac{\bar{a}}{\underline{b}\bar{b}}\right)x_2 + \frac{\bar{a}}{\underline{b}}.$$

For $(x_1, x_2) \in RC$, $ff^{RC}(x_1, x_2)$ is defined as the minimum of $h_1(x_1, x_2)$ and $h_2(x_1, x_2)$. Since $h_1(x_1, x_2)$ and $h_2(x_1, x_2)$ are linear functions on RC, this implies that $ff^{RC}(x_1, x_2)$ is a concave function on RC.

Suppose that $(x_1, x_2) \in RC$. Then, since $0 < \underline{a} \leqslant x_1$ and $0 < \underline{b} \leqslant x_2 \leqslant \bar{b}$, it follows that

$$\bar{b}x_1 - \underline{a}x_2 \geqslant 0$$

and

$$x_2 - \underline{b} \geqslant 0.$$

Therefore,

$$\begin{aligned} 0 &\leqslant (\bar{b}x_1 - \underline{a}x_2)(x_2 - \underline{b}) \\ &= \bar{b}x_1x_2 - \underline{a}x_2^2 - \underline{b}\bar{b}x_1 + \underline{a}\,\underline{b}x_2. \end{aligned}$$

By dividing the right-hand side of this inequality by $\underline{b}\bar{b}x_1 > 0$ and rearranging terms, we obtain

$$\frac{1}{\underline{b}}x_2 - \left(\frac{\underline{a}}{\underline{b}\bar{b}}\right)\left(\frac{x_2^2}{x_1}\right) + \left(\frac{\underline{a}}{\bar{b}}\right)\left(\frac{x_2}{x_1}\right) \geqslant 1\,.$$

By multiplying both sides of this inequality by $(x_1/x_2) > 0$, it follows that

$$h_1(x_1, x_2) \geqslant ff(x_1, x_2)\,.$$

In a similar fashion, it can be shown that

$$h_2(x_1, x_2) \geqslant ff(x_1, x_2)\,.$$

Therefore,

$$ff^{RC}(x_1, x_2) \geqslant ff(x_1, x_2)$$

for all $(x_1, x_2) \in RC$.

Now suppose that $w(x_1, x_2)$ is a concave function on RC such that

$$w(x_1, x_2) \geqslant ff(x_1, x_2) \quad \text{for all } (x_1, x_2) \in RC\,, \tag{7}$$

and, for some $(\bar{x}_1, \bar{x}_2) \in RC$

$$w(\bar{x}_1, \bar{x}_2) < \min\{h_1(\bar{x}_1, \bar{x}_2), h_2(\bar{x}_1, \bar{x}_2)\}\,. \tag{8}$$

Denote the vertices of RC by $A = (\underline{a}, \bar{b})$, $B = (\bar{a}, \bar{b})$, $C = (\bar{a}, \underline{b})$ and $D = (\underline{a}, \underline{b})$. Then, since $(\bar{x}_1, \bar{x}_2) \in RC$, $(\bar{x}_1, \bar{x}_2)$ belongs either to the convex hull of $\{A, B, C\}$ or to the convex hull of $\{A, C, D\}$ or both. Assume that $(\bar{x}_1, \bar{x}_2)$ belongs to the convex hull of $\{A, B, C\}$. The proof for the case where $(\bar{x}_1, \bar{x}_2)$ belongs to the convex hull of $\{A, D, C\}$ is similar. Then we may choose nonnegative scalars α_1, α_2 and α_3 that sum to 1.0 such that

$$(\bar{x}_1, \bar{x}_2) = \alpha_1 A + \alpha_2 B + \alpha_3 C\,. \tag{9}$$

Then

$$\begin{aligned} w(\bar{x}_1, \bar{x}_2) &\geqslant \alpha_1 w(A) + \alpha_2 w(B) + \alpha_3 w(C) \\ &\geqslant \alpha_1\left(\frac{\underline{a}}{\bar{b}}\right) + \alpha_2\left(\frac{\bar{a}}{\bar{b}}\right) + \alpha_3\left(\frac{\bar{a}}{\underline{b}}\right), \end{aligned}$$

where the first inequality follows by the concavity of w on RC, and the second inequality follows from (7). Since $h_2(A) = ff(A)$, $h_2(B) = ff(B)$ and $h_2(C) = ff(C)$, this implies that

$$w(\bar{x}_1, \bar{x}_2) \geqslant \alpha_1 h_2(A) + \alpha_2 h_2(B) + \alpha_3 h_2(C)\,.$$

By (9) and the linearity of $h_2(x_1, x_2)$, this implies that

$$w(\bar{x}_1, \bar{x}_2) \geqslant h_2(\bar{x}_1, \bar{x}_2)\,. \tag{10}$$

To complete the proof, we will derive a contradiction to (10). Towards this end, notice that since $(\bar{x}_1, \bar{x}_2)$ belongs to the convex hull of $\{A, B, C\}$, by considering the line through points A and C, it follows that

$$\bar{x}_2 \geqslant \left(\frac{\bar{b} - \underline{b}}{\underline{a} - \bar{a}}\right)\bar{x}_1 + \left(\frac{\underline{a}\underline{b} - \bar{a}\bar{b}}{\underline{a} - \bar{a}}\right).$$

By multiplying both sides of this inequality by $[(\underline{a} - \bar{a})/\underline{b}\bar{b})] < 0$ and rearranging terms, we obtain

$$\frac{1}{\bar{b}}\bar{x}_1 - \left(\frac{\bar{a}}{\underline{b}\bar{b}}\right)\bar{x}_2 + \frac{\bar{a}}{\underline{b}} \leqslant \frac{1}{\underline{b}}\bar{x}_1 - \left(\frac{\underline{a}}{\underline{b}\bar{b}}\right)\bar{x}_2 + \frac{\underline{a}}{\bar{b}}.$$

By the definitions of h_1 and h_2, this is equivalent to

$$h_2(\bar{x}_1, \bar{x}_2) \leqslant h_1(\bar{x}_1, \bar{x}_2).$$

Therefore,

$$ff^{RC}(\bar{x}_1, \bar{x}_2) = h_2(\bar{x}_1, \bar{x}_2).$$

From (8),

$$w(\bar{x}_1, \bar{x}_2) < ff^{RC}(x_1, x_2).$$

Taken together, that latter two relationships imply that

$$w(\bar{x}_1, \bar{x}_2) < h_2(\bar{x}_1, \bar{x}_2).$$

Since this contradicts (10), the proof is complete. □

The following is an immediate consequence of Theorem 2,

COROLLARY 1. *Let $\underline{a}$, $\bar{a}$ and $ff(x_1, x_2)$ be defined as in Theorem 2. Consider the horizontal line segment S given by*

$$S = \{(x_1, x_2) \in \mathcal{R}^2 \mid \underline{a} \leqslant x_1 \leqslant \bar{a},\ x_2 = b\},$$

where $b \in \mathcal{R}$.
(i) If either $\underline{a}, b > 0$ or $\bar{a}, b < 0$, then for all $(x_1, x_2) \in S$, $ff^S(x_1, x_2) = ff(x_1, x_2)$.
(ii) If either $\bar{a} < 0$, $b > 0$ or $\underline{a} > 0$, $b < 0$, then for all $(x_1, x_2) \in S$, $ff_S(x_1, x_2) = ff(x_1, x_2)$.

3. Prototype Algorithm

To search for a global optimal solution for problem (P_H), the prototype algorithm uses a branch and bound procedure. There are two fundamental processes in this procedure, a branching process and an upper bounding process.

The branching process in the prototype algorithm iteratively subdivides the rectangle H into subrectangles. This branching process helps the algorithm identify a location in the feasible region $Z(H)$ of problem (P_H) that contains a global optimal solution to the problem.

During each iteration of the algorithm, the branching process creates a more refined partition of a portion of $H^1 \triangleq H$ that cannot yet be excluded from consideration in the search for a global optimal solution for problem (P_H). The initial partition PR_1 consists simply of H^1.

During iteration k of the algorithm, $k \geqslant 1$, the branching process is used to help create a new partition PR_{k+1}. First, a screening procedure is used to remove any rectangles from P_k that can, at this point of the search, be excluded from further consideration, and PR_{k+1} is temporarily set equal to the set of rectangles that remain. Later in iteration k, a rectangle H^k in PR_{k+1} is identified for further examination. The branching process is then invoked to subdivide H^k into two subrectangles H^{2k}, H^{2k+1}. The only assumption in the branching process is that H^{2k} and H^{2k+1} create a partition of H^k, i.e., that their union is H^k and they intersect only on their relative boundaries [14]. The new partition PR_{k+1} of the portion of H^1 remaining under consideration is then given by

$$PR_{k+1} = (PR_{k+1} \backslash \{H^k\}) \cup \{H^{2k}, H^{2k+1}\} .$$

The second fundamental process in the prototype algorithm is the upper bounding process. Given $\hat{H} = H^1$ or a subrectangle $\hat{H}$ of H^1, it is assumed that the upper bounding process can accomplish two things. First, it is assumed to be able to compute an upper bound $UB(\hat{H})$ for the optimal objective function value $v_{\hat{H}}$ of the sum of ratios problem

$$\begin{aligned}(P_{\hat{H}})\ v_{\hat{H}} = \max & \sum_{i=1}^{p} \frac{t_i}{s_i} \\ \text{s.t. } & n_i(x) - t_i \geqslant 0 , \quad i = 1, 2, \dots , p , \\ & -d_i(x) + s_i \geqslant 0 , \quad i = 1, 2, \dots , p , \\ & x \in X, (t, s) \in \hat{H}\end{aligned}$$

that satisfies $UB(\hat{H}) = -\infty$ when the feasible region $Z(\hat{H})$ of problem $(P_{\hat{H}})$ is empty. Second, the upper bounding process is assumed to be capable of identifying a *distinguished point* $\omega(\hat{H})$ in $Z(\hat{H})$ when $Z(\hat{H}) \neq \emptyset$.

We may state the prototype branch and bound algorithm for globally solving problem (P_H) as follows.

3.1. PROTOTYPE ALGORITHM

Step 0 (Initialization): Let $\hat{H} = H = H^1$. If a feasible solution $(x^1, t^1, s^1) \in Z(\hat{H})$ is available, let $(\bar{x}^1, \bar{t}^1, \bar{s}^1)$ denote the available feasible solution of maximum objective function value in problem $(P_{\hat{H}})$, and let

$$v^1 = \sum_{i=1}^{p} \frac{\bar{t}_i^1}{\bar{s}_i^1}\,. \tag{11}$$

Otherwise, let $\{(\bar{x}^1, \bar{t}^1, \bar{s}^1)\} = \emptyset$, and let $v^1 = E$, where $E \in [-\infty, \infty)$ is a lower bound for $v_{\hat{H}}$ in problem $(P_{\hat{H}})$. Compute an upper bound $UB(\hat{H})$ for $v_{\hat{H}}$. Set $PR_1 = \{H^1\}$ and $k = 1$.

Step 1 (Screening): Set

$$R_k = \{\hat{H} \in PR_k \mid UB(\hat{H}) > v^k\}\,,$$

and set

$$PR_{k+1} = \{\hat{H} \in PR_k \mid \hat{H} \in R_k\}\,.$$

Step 2 (Termination Criterion): If $R_k = \emptyset$, then terminate. Problem (P_H) is infeasible if $v^k = E$. Otherwise, $(\bar{x}^k, \bar{t}^k, \bar{s}^k)$ is a global optimal solution for problem (P_H).

Step 3 (Branching): Let $H^k \in R_k$ satisfy

$$H^k \in \operatorname{argmax}\{UB(\hat{H}) \mid \hat{H} \in R_k\}\,.$$

Partition H^k into two subrectangles H^{2k}, H^{2k+1}.

Step 4 (New Partition): Let

$$PR_{k+1} = (PR_{k+1} \backslash \{H^k\}) \cup \{H^{2k}, H^{2k+1}\}\,.$$

Step 5 (Evaluation): Set $k = k+1$. For each $\hat{H} \in PR_k$, determine $UB(\hat{H})$ and, if $Z(\hat{H}) \neq \emptyset$, determine the distinguished point $\omega(\hat{H}) \in Z(\hat{H})$.

Step 6 (Incumbent): Define $(\bar{x}^k, \bar{t}^k, \bar{s}^k)$ to be the point that, among $(\bar{x}^{k-1}, \bar{t}^{k-1}, \bar{s}^{k-1})$ and all points $\omega(\hat{H})$ found in Step 5, maximizes the objective function value in problem (P_H). Let

$$v^k = \sum_{i=1}^{p} \frac{\bar{t}_i^k}{\bar{s}_i^k} \tag{12}$$

and go to Step 1.

For each $k \geqslant 1$, the value v^k provides a lower bound for the optimal value v_H of problem (P_H), and the point $(\bar{x}^k, \bar{t}^k, \bar{s}^k) \in Z(H)$ such that (11) or (12) holds is called the *incumbent solution*. Notice for each $k \geqslant 1$ that the screening process eliminates rectangles $\hat{H}$ from further consideration for which $UB(\hat{H}) \leqslant v^k$.

By the construction of the algorithm, when the algorithm is finite, it either finds a global optimal solution for problem (P_H) or detects that problem (P_H) is infeasible. It is also possible for the algorithm to be infinite. The following definition and result help to analyze this case.

DEFINITION 2. The prototype algorithm for problem (P_H) is said to be convergent if it is infinite and $\lim_{k\to\infty} v^k = v_H$ or if it is finite.

PROPOSITION 1. *If the prototype algorithm for problem* (P_H) *is infinite and convergent, then any accumulation point of* $\{(\bar{x}^k, \bar{t}^k, \bar{s}^k)\}_{k=1}^{\infty}$ *is a global optimal solution for problem* (P_H).

Proof. Suppose that the prototype algorithm for problem (P_H) is infinite and convergent. Then, by (11)–(12) and the definition of convergence,

$$\lim_{k\to\infty} v^k = \lim_{k\to\infty}\left(\sum_{i=1}^{p} \frac{\bar{t}_i^k}{\bar{s}_i^k}\right) \tag{13}$$

$$= v_H \,. \tag{14}$$

Let $(\bar{x}, \bar{t}, \bar{s})$ be an accumulation point of $\{(\bar{x}^k, \bar{t}^k, \bar{s}^k\}_{k=1}^{\infty}$. Then for some $K \subseteq \{1, 2, \ldots\}$,

$$\lim_{k\in K}(\bar{x}^k, \bar{t}^k, \bar{s}^k) = (\bar{x}, \bar{t}, \bar{s})\,. \tag{15}$$

By (13)–(14), since

$$\left\{\sum_{i=1}^{p} \frac{\bar{t}_i^k}{\bar{s}_i^k}\right\}_{k\in K}$$

is a subsequence of

$$\begin{gathered}\left\{\sum_{i=1}^{p} \frac{\bar{t}_i^k}{\bar{s}_i^k}\right\}_{k=1}^{\infty}, \\ \lim_{k\in K}\left(\sum_{i=1}^{p} \frac{\bar{t}_i^k}{\bar{s}_i^k}\right) = v_H \,.\end{gathered} \tag{16}$$

From (15) and the continuity of the objective function of problem (P_H) over H,

$$\lim_{k\in K}\left(\sum_{i=1}^{p} \frac{\bar{t}_i^k}{\bar{s}_i^k}\right) = \sum_{i=1}^{p} \frac{\bar{t}_i}{\bar{s}_i}\,. \tag{17}$$

From (16) and (17),

$$\sum_{i=1}^{p} \frac{\bar{t}_i}{\bar{s}_i} = v_H \,. \tag{18}$$

By (15), since the feasible region $Z(H)$ of problem (P_H) is a closed set, $(\bar{x}, \bar{t}, \bar{s}) \in Z(H)$. Together with (18), this implies that $(\bar{x}, \bar{t}, \bar{s})$ is a global optimal solution for problem (P_H). □

4. Two Implementations

To implement the prototype algorithm of Section 3 in ways that yield convergent algorithms for problem (P_H), the upper bounding and branching processes of the prototype algorithm must be implemented appropriately. In this section, we describe two of the convergent algorithms that can be obtained from the prototype algorithm in this way. These two algorithms use the same upper bounding process, but different branching processes.

4.1. THE UPPER BOUNDING PROCESS

Let $\hat{H}$ denote either H or a subrectangle of H that is generated by the prototype algorithm. Then $\hat{H}$ may be written

$$\hat{H} = \hat{H}_1 \times \hat{H}_2 \times \cdots \times \hat{H}_p , \tag{19}$$

where, for each $i = 1, 2, \ldots, p$,

$$\hat{H}_i = \{(t_i, s_i) \in \mathscr{R}^2 \mid \hat{l}_i \leqslant t_i \leqslant \hat{u}_i, \hat{L}_i \leqslant s_i \leqslant \hat{U}_i\} .$$

Here, $\hat{l}_i$, $\hat{u}_i$, $\hat{L}_i$ and $\hat{U}_i$ are positive scalars such that $\hat{l}_i \leqslant \hat{u}_i$, $\hat{L}_i \leqslant \hat{U}_i$, $i = 1, 2, \ldots, p$, and, for at least one $i \in \{1, 2, \ldots, p\}$, $\hat{L}_i < \hat{U}_i$. For each $i \in \{1, 2, \ldots, p\}$, let

$$ff_i(t_i, s_i) = \frac{t_i}{s_i} \tag{20}$$

for all $(t_i, s_i) \in \hat{H}_i$. Then, for each $i = 1, 2, \ldots, p$, by Theorem 2, since $\hat{l}_i, \hat{L}_i > 0$, the concave envelope $ff_i^{\hat{H}_i}$ of ff_i on $\hat{H}_i$ is given by

$$ff_i^{\hat{H}_i}(t_i, s_i) = \min\left\{\frac{1}{\hat{L}_i} t_i - \left(\frac{\hat{l}_i}{\hat{L}_i\hat{U}_i}\right)s_i + \frac{\hat{l}_i}{\hat{U}_i}, \frac{1}{\hat{U}_i} t_i - \left(\frac{\hat{u}_i}{\hat{L}_i\hat{U}_i}\right)s_i + \frac{\hat{u}_i}{\hat{L}_i}\right\} . \tag{21}$$

By Theorem IV. 8 in [14] and (19), (20), the concave envelope $g^{\hat{H}}(t, s)$ of $g(t, s) \overset{\triangle}{=} \sum_{i=1}^{p} t_i/s_i$ on $\hat{H}$ is then given, for each $(t, s) \in \hat{H}$, by

$$g^{\hat{H}}(t, s) = \sum_{i=1}^{p} ff_i^{\hat{H}_i}(t_i, s_i) . \tag{22}$$

To calculate an upper bound $UB(\hat{H})$ for $v_{\hat{H}}$, the upper bounding process maximizes the concave envelope $g^{\hat{H}}(t, s)$ of $g(t, s)$ over the feasible region $Z(\hat{H})$ of problem $(P_{\hat{H}})$. To accomplish this, the convex programming problem $(PUB_{\hat{H}})$ given by

$$(PUB_{\hat{H}}) \max \sum_{i=1}^{p} r_i$$

$$s.t.$$

$$r_i \leqslant \frac{1}{\hat{L}_i} t_i - \left(\frac{\hat{l}_i}{\hat{L}_i\hat{U}_i}\right)s_i + \frac{\hat{l}_i}{\hat{U}_i}, \quad i = 1, 2, \ldots, p ,$$

$$
\begin{aligned}
& r_i \leqslant \frac{1}{\hat{U}_i} t_i - \left(\frac{\hat{u}_i}{\hat{L}_i \hat{U}_i} \right) s_i + \frac{\hat{u}_i}{\hat{L}_i} , \quad i = 1, 2, \ldots, p , \\
& n_i(x) - t_i \geqslant 0 , \quad i = 1, 2, \ldots, p , \\
& -d_i(x) + s_i \geqslant 0 , \quad i = 1, 2, \ldots, p , \\
& x \in X , \quad (t, s) \in \hat{H} ,
\end{aligned}
$$

is solved, and $UB(\hat{H})$ is set equal to the optimal value of this problem. From (21)–(22), this optimal value does, in fact, equal the maximum of $g^{\hat{H}}(t, s)$ over $Z(\hat{H})$. When $Z(\hat{H})$ is empty, $UB(\hat{H})$ is set equal to $-\infty$. When $Z(\hat{H})$ is not empty, the upper bounding process chooses any optimal solution found for problem $(PUB_{\hat{H}})$ as the distinguished point $\omega(\hat{H})$.

4.2. TWO BRANCHING PROCESSES

Without loss of generality, let $\hat{H}$ denote a rectangle H^k that is to be partitioned into two subrectangles by the branching process of the prototype algorithm, where $k \geqslant 1$ and $\hat{H}$ is defined as in Section 4.1. At least two suitable branching processes are available. These processes are adaptations of two common branching processes from global optimization [30]. In both processes, although $\hat{H} \subseteq \mathcal{R}^{2p}$, the branching process takes place in a space of only dimension p.

4.2.1. *Bisection of Ratio α*

Let α be a prechosen parameter that satisfies $0.0 < \alpha \leqslant 0.5$. The procedure for forming a bisection of ratio α of $\hat{H}$ into two subrectangles $\hat{H}'$ and $\hat{H}''$ is as follows.

Step 1: Let $(\hat{U}_j - \hat{L}_j) = \max_{i=1,2,\ldots,p}\{\hat{U}_i - \hat{L}_i\}$.
Step 2: Let v_j satisfy

$$\min\{v_j - \hat{L}_j, \hat{U}_j - v_j\} = \alpha(\hat{U}_j - \hat{L}_j) . \tag{23}$$

Step 3: Let

$$
\begin{aligned}
\hat{H}_j' &= \{(t_j, s_j) \in \mathcal{R}^2 \mid \hat{l}_j \leqslant t_j \leqslant \hat{u}_j, \hat{L}_j \leqslant s_j \leqslant v_j\} , \\
\hat{H}_j'' &= \{(t_j, s_j) \in \mathcal{R}^2 \mid \hat{l}_j \leqslant t_j \leqslant \hat{u}_j, v_j \leqslant s_j \leqslant \hat{U}_j\} .
\end{aligned}
$$

Step 4: Let

$$
\begin{aligned}
\hat{H}' &= \hat{H}_1 \times \hat{H}_2 \times \cdots \times \hat{H}_{j-1} \times \hat{H}_j' \times \hat{H}_{j+1} \times \cdots \times \hat{H}_p , \\
\hat{H}'' &= \hat{H}_1 \times \hat{H}_2 \times \cdots \times \hat{H}_{j-1} \times \hat{H}_j'' \times \hat{H}_{j+1} \times \cdots \times \hat{H}_p .
\end{aligned}
$$

Notice from (23) that when $\alpha < 0.5$, the ratio of the length of the smaller of the two edges in $\hat{H}_j'$ and $\hat{H}_j''$ that correspond to s_j to the length of the edge of $\hat{H}_j$ that corresponds to s_j is α. When $\alpha = 0.5$, the bisection of ratio α is accomplished by bisecting the interval $[\hat{L}_j, \hat{U}_j]$ at its midpoint $v_j = 0.5(\hat{L}_j + \hat{U}_j)$.

4.2.2. *The ω-Subdivision Rule*

Let $\omega(\hat{H}) = (\hat{x}, \hat{t}, \hat{s})$ be the distinguished point determined by the prototype algorithm for the rectangle $\hat{H} = H^k$. For each $i = 1, 2, \ldots, p$, recall that the concave envelope $ff_i^{\hat{H}_i}$ of ff_i on $\hat{H}_i$ is given by (21). The procedure for subdividing $\hat{H}$ into two subrectangles $\hat{H}'$ and $\hat{H}''$ by ω-subdivision is as follows.

Step 1: Let

$$\theta_j = \max_{i=1,2,\ldots,p} \{ ff_i^{\hat{H}_i}(\hat{t}_i, \hat{s}_i) - ff_i(\hat{t}_i, \hat{s}_i) \} .$$

Step 2: Let

$$\hat{H}_j' = \{(t_j, s_j) \in \mathscr{R}^2 \mid \hat{l}_j \leqslant t_j \leqslant \hat{u}_j, \hat{L}_j \leqslant s_j \leqslant \hat{s}_j \} ,$$

$$\hat{H}_j'' = \{(t_j, s_j) \in \mathscr{R}^2 \mid \hat{l}_j \leqslant t_j \leqslant \hat{u}_j, \hat{s}_j \leqslant s_j \leqslant \hat{U}_j \} .$$

Step 3: Let

$$\hat{H}' = \hat{H}_1 \times \hat{H}_2 \times \cdots \times \hat{H}_{j-1} \times \hat{H}_j' \times \hat{H}_{j+1} \times \cdots \times \hat{H}_p ,$$

$$\hat{H}'' = \hat{H}_1 \times \hat{H}_2 \times \cdots \times \hat{H}_{j-1} \times \hat{H}_j'' \times \hat{H}_{j+1} \times \cdots \times \hat{H}_p .$$

The ω-subdivision rule attempts to subdivide $\hat{H}$ in such a way so as to maximize the improvement in the quality of the concave envelope overestimations for $\hat{H}'$ and $\hat{H}''$ as compared to that for $\hat{H}$.

Notice that both bisection of ratio α and ω-subdivision yield branching processes for the prototype algorithm that, at each iteration, subdivide an interval $[L_j, U_j]$ that is associated with a denominator s_j of problem (P_H). Thus, these branching processes *never* subdivide any of the intervals $[l_i, u_i]$, $i = 1, 2, \ldots, p$, that are associated with the numerators t_i, $i = 1, 2, \ldots, p$, of problem (P_H).

4.3. TWO ALGORITHMS AND CONVERGENCE

We will refer to the algorithm obtained by executing the upper bounding and branching processes of the prototype algorithm by the bounding process given in Section 4.1 and by bisection of ratio α, respectively, as the Bisection Algorithm. Similarly, we will refer to the algorithm obtained by executing the upper bounding and branching processes of the prototype algorithm by the bounding process given in Section 4.1 and by the ω-subdivision rule, respectively, as the Omega Algorithm. The main goal of this section is to show that the Bisection and Omega Algorithms are convergent in the sense of Definition 2.

4.3.1. *Convergence of the Bisection Algorithm*

Notice that when the Bisection Algorithm is infinite, since $\{1, 2, \ldots, p\}$ is finite, there exists an infinite sequence $\{H^q\}_{q=1}^{\infty}$ of rectangles in $\mathscr{R}^{2p}$ generated by the algorithm such that for each $q = 1, 2, \ldots, H^{q+1} \subset H^q$ and H^{q+1} is formed from H^q

by the bisection of ratio α process, where, in step 1 of this process, for some fixed $j_0 \in \{1, 2, \ldots, p\}$,

$$(U^q_{j_0} - L^q_{j_0}) = \max_{i=1,2,\ldots,p} \{U^q_i - L^q_i\},$$

where $\{(t, s) \in \mathcal{R}^{2p} \mid l^q_i \leqslant t_i \leqslant u^q_i,\ L^q_i \leqslant s_i \leqslant U^q_i,\ i = 1, 2, \ldots, p\} = H^q$ for each q. Assume in the next result that $\{H^q\}^{\infty}_{q=1}$ is a sequence of rectangles of this type, and, for each q and each $i \in \{1, 2, \ldots, p\}$, let

$$H^q_i = \{(t_i, s_i) \in \mathcal{R}^2 \mid l^q_i \leqslant t_i \leqslant u^q_i, L^q_i \leqslant s_i \leqslant U^q_i\}.$$

LEMMA 1. *For some subsequence Q of $\{1, 2, \ldots\}$, the limit rectangle*

$$H^{\infty}_{j_0} = \bigcap_{q \in Q} H^q_{j_0}$$

is a line segment in $\mathcal{R}^2$ parallel to the t_{j_0}-axis.

Proof. By Lemma 5.4 in [30] and the bisection of ratio α rule, there exists a subsequence Q of $\{1, 2, \ldots\}$ such that

$$\lim_{q \in Q} L^q_{j_0} = L_{j_0} \in \mathcal{R},$$
$$\lim_{q \in Q} U^q_{j_0} = U_{j_0} \in \mathcal{R},$$

and

$$\lim_{q \in Q} v^q_{j_0} = \bar{v} \in \{L_{j_0}, U_{j_0}\},$$

where, for each $q \in Q$, $v^q_{j_0}$ denotes the point v_{j_0} chosen in step 2 of the bisection of ratio α rule. Therefore, $\bar{v} = L_{j_0}$ or $\bar{v} = U_{j_0}$. In either case, $\bar{v} \in \mathcal{R}$ is a single point, so that

$$\begin{aligned} H^{\infty}_{j_0} &= \bigcap_{q \in Q} H^q_{j_0} \\ &= \{(t_{j_0}, s_{j_0}) \mid l^q_{j_0} \leqslant t_{j_0} \leqslant u^q_{j_0}, s_{j_0} = \bar{v}\}, \end{aligned}$$

which is a line segment in $\mathcal{R}^2$ parallel to the t_{j_0}-axis. □

THEOREM 3. *Suppose that the Bisection Algorithm is infinite, and let $\{H^q\}^{\infty}_{q=1}$ denote a sequence of rectangles in $\mathcal{R}^{2p}$ generated by the algorithm such that for each $q = 1, 2, \ldots, H^{q+1} \subset H^q$. Then, for some subsequence Q of $\{1, 2, \ldots\}$,*

$$\lim_{q \in Q} \left[UB(H^q) - \sum_{i=1}^{p} \frac{\bar{t}^q_i}{\bar{s}^q_i} \right] = 0.$$

Proof. Since $\{H^q\}^{\infty}_{q=1}$ is infinite, by steps 1 and 2 of the Bisection Algorithm, we may choose a subsequence Q of $\{1, 2, \ldots\}$ such that for each $q \in Q$, $UB(H^q) \neq -\infty$. Because $\{H^q\}_{q \in Q}$ is infinite, we may assume without loss of generality that $\{H^q\}_{q \in Q}$

has the properties of the sequence $\{H^q\}_{q=1}^{\infty}$ of Lemma 1. For each $q \in Q$, since $UB(H^q) \neq -\infty$, the upper bounding process implies that

$$g^{H^q}(t^q, s^q) = UB(H^q)\,, \tag{24}$$

where g^{H^q} denotes the concave envelope of $g(t,s) = \sum_{i=1}^{p} t_i/s_i$ over H^q, and $\omega(H^q) = (x^q, t^q, s^q)$ is the distinguished point corresponding to H^q chosen by the upper bounding process. For each q, since $\omega(H^q) \in Z(H^q)$, $Z(H^q) \neq \emptyset$. By repeated application of Lemma 1, we may assume without loss of generality that

$$\lim_{q \in Q} H^q = H_1^{\infty} \times H_2^{\infty} \times \cdots \times H_p^{\infty} \stackrel{\triangle}{=} H^{\infty}\,, \tag{25}$$

where, for each $i = 1, 2, \ldots, p$, $H_i^{\infty} \subseteq \mathscr{R}^2$ is a line segment parallel to the t_i-axis. For each $q \in Q$,

$$\begin{aligned} UB(H^q) &= \max_{(x,t,s) \in Z(H^q)} g^{H^q}(t,s) \\ &\geqslant v_H \\ &\geqslant g(\bar{t}^q, \bar{s}^q) && (26) \\ &\geqslant g(t^q, s^q)\,, && (27) \end{aligned}$$

where the first equation follows, since $Z(H^q) \neq \emptyset$, from the definition of $UB(H^q)$ in the upper bounding process, the first inequality follows from step 3 of the Bisection Algorithm and the validity of the upper bounding process, the second inequality follows because $(\bar{x}^q, \bar{t}^q, \bar{s}^q) \in Z(H^q)$, and the third inequality holds by the choice of the incumbent solution in step 6 of the Bisection Algorithm. For each $q \in Q$, $(t^q, s^q) \in H^q \subseteq H$. Therefore, $\{(t^q, s^q)\}_{q \in Q}$ has a convergent subsequence, and, by (25), the limit point of this subsequence lies in H^{∞}. Assume without loss of generality that

$$\lim_{q \in Q} (t^q, s^q) = (\bar{t}, \bar{s}) \in H^{\infty}\,. \tag{28}$$

By the continuity of g on H, this implies that

$$\lim_{q \in Q} g(t^q, s^q) = g(\bar{t}, \bar{s})\,. \tag{29}$$

Since $(t^q, s^q) \in H^q$ for all $q \in Q$, by using (28), Theorem 2(i) and Corollary 1, it can be seen that

$$\lim_{q \in Q} g^{H^q}(t^q, s^q) = g(\bar{t}, \bar{s})\,. \tag{30}$$

Combining (24), (26), (27), (29) and (30), we obtain

$$g(\bar{t}, \bar{s}) = \lim_{q \in Q} UB(H^q) = \lim_{q \in Q} g(\bar{t}^q, \bar{s}^q) = g(\bar{t}, \bar{s})\,.$$

Since for each $q \in Q$, $g(\bar{t}^q, \bar{s}^q) = \sum_{i=1}^{p} \bar{t}_i^q / \bar{s}_i^q$, this completes the proof. □

COROLLARY 2. *The Bisection Algorithm is convergent.*

Proof. Suppose that the Bisection Algorithm is infinite. Then, as noted previously, we may choose a sequence of rectangles, which we denote without loss of generality by $\{H^q\}_{q=1}^{\infty}$, such that for each $q = 1, 2, \ldots, H^{q+1} \subset H^q$ and H^{q+1} is formed from H^q by the bisection of ratio α process. By the validity of the upper bounding process and steps 3 and 6 of the algorithm, for each $q = 1, 2, \ldots,$

$$UB(H^q) \geqslant v_H \geqslant g(\bar{t}^{q+1}, \bar{s}^{q+1}) \geqslant g(\bar{t}^q, \bar{s}^q) = v^q \,, \tag{31}$$

and $\{v^q\}_{q=1}^{\infty}$ is a nondecreasing sequence of real numbers. By Theorem 3, for some $Q \subseteq \{1, 2, \ldots\}$,

$$\lim_{q \in Q} UB(H^q) = \lim_{q \in Q} g(\bar{t}^q, \bar{s}^q) \,.$$

From (31), this implies that

$$\lim_{q \in Q} UB(H^q) = v_H = \lim_{q \in Q} v^q \,.$$

Since $\{v^q\}_{q=1}^{\infty}$ is a nondecreasing sequence, this implies that $\lim_{q \to \infty} v^q = v_H$. □

From Corollary 2 and Proposition 1, whenever the Bisection Algorithm is infinite, any accumulation point $(\bar{x}, \bar{t}, \bar{s})$ of the sequence $\{(\bar{x}^k, \bar{t}^k, \bar{s}^k)\}_{k=1}^{\infty}$ that it generates is a global optimal solution for problem (P_H). From Theorem 1, $\bar{x}$ is then a global optimal solution for problem (P).

4.3.2. *Convergence of the Omega Algorithm*

For the Omega Algorithm, counterparts to Lemma 1, Theorem 3, and Corollary 2 hold. The proofs of these results, except for the proof of the counterpart of Theorem 3, are virtually identical to the proofs for the case of the Bisection Algorithm. In the interest of brevity, these results are not shown here. Together, however, they imply that the Omega Algorithm is convergent and, by Proposition 1, whenever it is infinite, any accumulation point $(\bar{x}, \bar{t}, \bar{s})$ of the sequence $\{(\bar{x}^k, \bar{t}^k, \bar{s}^k)\}_{k=1}^{\infty}$ that it generates is a global optimal solution for problem (P_H). From Theorem 1, $\bar{x}$ is then a global optimal solution for problem (P).

5. Computational Issues and Solved Examples

There are at least three computational issues that may arise in using either of the two suggested implementations of the prototype algorithm.

The first computational issue concerns step 2 of the prototype algorithm, the termination criterion. In practice, even after many iterations, R_k may remain nonempty. However, by the convergence results, it follows that for any $\epsilon > 0$,

$$\left(UB^k - \sum_{i=1}^{p} \frac{\bar{t}_i^k}{\bar{s}_i^k} \right) \leqslant \epsilon \tag{32}$$

will hold for k sufficiently large. In practice, it is recommended that the algorithms be terminated if, for some prechosen, relatively-small value of $\epsilon > 0$, (32) holds.

When termination occurs in this way, it is easy to show that $\bar{x}^k$ is a global ϵ-optimal solution and $h(x^*)$ is a global ϵ-optimal value for problem (P) in the sense that $x^* \in X$ and

$$h(x^*) + \epsilon \geqslant v \ .$$

The second computational issue concerns the upper bounding process. From Section 4.1, each upper bound in the algorithms is computed by solving a nonlinear, convex program of the form of problem $(PUB_{\hat{H}})$. These problems differ from one another only in the coefficients of $2p$ linear constraints and in the bounds for $(t, s) \in \mathscr{R}^{2p}$. Therefore, an optimal solution to one problem can potentially be used to good advantage as a starting solution to the next problem. The ability to implement this idea will depend upon the particular convex programming algorithm and code used to solve these problems.

The third computational issue concerns the assumption for problem (P) that for each $i = 1, 2, \ldots p$, positive scalars l_i, u_i, L_i and U_i are available such that for all $x \in X$, $l_i \leqslant n_i(x) \leqslant u_i$ and $L_i \leqslant d_i(x) \leqslant U_i$, $i = 1, 2, \ldots, p$. In some cases, these scalars may not be available but, rather, must be computed or estimated. For each $i = 1, 2, \ldots, p$, the computation of u_i and L_i can be accomplished by maximizing $n_i(x)$ over X and minimizing $d_i(x)$ over X, respectively. Since these computations involve solving convex programs, they are not problematical. However, for each $i = 1, 2, \ldots, p$, finding values for l_i and U_i, if they are not readily available, will require a different approach. Of course, for each $i = 1, 2, \ldots, p$, if $n_i(x)$ and $d_i(x)$ are linear functions and X is a polytope, these values can be found by solving $2p$ linear programming problems. In other cases, a special procedure may be needed. One such procedure, which is based upon convex programming, is given in [4].

We have coded the Omega Algorithm and used it to globally solve a number of sample problems (P). The code was written in the AMPL language [10], where we chose the MINOS solver [24] to solve the convex upper bounding subproblems. The AMPL code uses the optimal solution of each convex upper bounding problem to help to provide a good starting solution for the next problem. The sample problems were solved with the code on a Pentium II, 400 MHz personal computer.

Below we describe some of these sample problems and solution results. In each case, the feasible regions were small enough so that for each $i = 1, 2, \ldots, p$, l_i and U_i were found by extreme point search. Global ϵ-optimal solutions and ϵ-optimal values are given to the nearest one-hundredth.

EXAMPLE 1. In this example $p = n = 2$. The numerator and denominator functions are given for each $(x_1, x_2) \in \mathscr{R}^2$ by

$$n_1(x_1, x_2) = -x_1^2 + 3x_1 - x_2^2 + 3x_2 + 3.5 \ ,$$
$$n_2(x_1, x_2) = x_2 \ ,$$
$$d_1(x_1, x_2) = x_1 + 1.0 \ ,$$
$$d_2(x_1, x_2) = x_1^2 - 2x_1 + x_2^2 - 8x_2 + 20.0 \ ,$$

and X consists of all $(x_1, x_2) \in \mathscr{R}^2$ that satisfy the inequalities

$$\begin{aligned} &2x_1 + x_2 \leqslant 6\,, \\ &3x_1 + x_2 \leqslant 8\,, \\ &x_1 - x_2 \leqslant 1\,, \\ &x_1, x_2 \geqslant 1\,. \end{aligned}$$

With $\epsilon = 0.001$, the algorithm found the global ϵ-optimal solution $(x_1, x_2) = (1.00, 1.74)$ with global ϵ-optimal value $v = 4.06$ after 17 iterations. The MINOS code's total subproblem solution time was 0.62 s. The initial upper bound was 4.33, and the ϵ-optimal value v was discovered during iteration number 9.

EXAMPLE 2. In this example, $p = 2$ and $n = 3$. The numerator and denominator functions are given for each $(x_1, x_2, x_3) \in \mathscr{R}^3$ by

$$\begin{aligned} n_1(x_1, x_2, x_3) &= -x_1^2 + 4x_1 - 2x_2^2 + 8x_2 - 3x_3^2 + 12x_3 + 56\,, \\ n_2(x_1, x_2, x_3) &= -2x_1^2 + 16x_1 - x_2^2 + 8x_2 + 2\,, \\ d_1(x_1, x_2, x_3) &= x_1^2 - 2x_1 + x_2^2 - 2x_2 + x_3 + 20\,, \\ d_2(x_1, x_2, x_3) &= 2x_1 + 4x_2 + 6x_3\,, \end{aligned}$$

and

$$X = \{(x_1, x_2, x_3) \in \mathscr{R}^3 \mid x_1 + x_2 + x_3 \leqslant 10, -x_1 - x_2 + x_3 \leqslant 4, x_j \geqslant 1, j = 1, 2, 3\}\,.$$

With $\epsilon = 0.01$, a global ϵ-optimal solution $(x_1, x_2, x_3) = (1.81, 1.00, 1.00)$ with ϵ-optimal value $v = 6.12$ was found after 24 iterations of the algorithm. The MINOS code's total subproblem solution time was 1.00 s. The initial upper bound was 7.36, and the ϵ-optimal value v was discovered during iteration number 2.

EXAMPLE 3. In this example, $p = 3$ and $n = 4$. The numerator and denominator functions on $\mathscr{R}^4$ are given by

$$\begin{aligned} n_1(x_1, x_2, x_3, x_4) &= \left(\sum_{j=1}^{4} -x_j^2 + 16x_j\right) - 214\,, \\ n_2(x_1, x_2, x_3, x_4) &= -x_1^2 + 16x_1 - 2x_2^2 + 20x_2 - 3x_3^2 + 60x_3 - 4x_4^2 + 56x_4 \\ &\quad - 586\,, \\ n_3(x_1, x_2, x_3, x_4) &= \left(\sum_{j=1}^{4} -x_j^2 + 20x_j\right) - 324\,, \\ d_1(x_1, x_2, x_3, x_4) &= 2x_1 - x_2 - x_3 + x_4 + 2\,, \\ d_2(x_1, x_2, x_3, x_4) &= -x_1 + x_2 + x_3 - x_4 + 10\,, \\ d_3(x_1, x_2, x_3, x_4) &= x_1^2 - 4x_4\,, \end{aligned}$$

and X consists of all $(x_1, x_2, x_3, x_4) \in \mathscr{R}^4$ such that

$$6 \leqslant x_1 \leqslant 10\,,$$
$$4 \leqslant x_2 \leqslant 6\,,$$
$$8 \leqslant x_3 \leqslant 12\,,$$
$$6 \leqslant x_4 \leqslant 8\,,$$
$$x_1 + x_2 + x_3 + x_4 \leqslant 34\,.$$

With $\epsilon = 0.01$, the algorithm found the global ϵ-optimal solution $(x_1, x_2, x_3, x_4) = (6.00, 6.00, 10.06, 8.00)$ with global ϵ-optimal value $v = 16.17$ after 37 iterations. The MINOS code's total subproblem solution time was 2.06 s. The initial upper bound was 27.04, and the ϵ-optimal value was discovered in iteration number 24.

We caution that while these examples may seem, in a very preliminary way, to indicate the potential viability of the Omega Algorithm for globally solving problem (P), considerable additional computational tests will be needed to draw precise conclusions concerning the computational capabilities of this algorithm (or of the Bisection Algorithm).

6. Concluding Remarks

Two versions of a branch and bound algorithm for globally solving the nonlinear sum of ratios problem (P) have been presented. In both algorithms, the branch and bound search creates rectangular regions that belong to $\mathscr{R}^{2p}$, where p is the number of ratios in the objective function of problem (P). However, the branching process in both algorithms takes place in $\mathscr{R}^p$, rather than $\mathscr{R}^{2p}$. In addition, the upper bounding subproblems in these two algorithms are convex programming problems that are quite similar to one another. These characteristics of the two algorithms offer computational advantages that can enhance the efficiencies of the algorithms. It is hoped that these algorithms and the ideas used to create these algorithms will offer useful tools for solving nonlinear sums of ratios problems.

Acknowledgement

The author gratefully acknowledges the helpful comments of an anonymous referee regarding an earlier draft of this article.

References

1. Almogy, Y. and Levin, O. (1970), Parametric analysis of a multi-stage stochastic shipping problem, *Operational Research '69*, Tavistok Publications, London.
2. Benson, H.P. (1996), Deterministic algorithms for constrained concave minimization: a unified critical survey, *Naval Research Logistics* 43: 765–795.
3. Benson, H.P. (2001), On the construction and utilization of convex and concave envelopes of bilinear and fractional functions, Working Paper, Department of Decision and Information Sciences, University of Florida, Gainesville, FL.
4. Benson, H.P. (1999), An outcome space branch and bound outer-approximation algorithm for convex multiplicative programming, *Journal of Global Optimization* 15: 315–342.

5. Benson, H.P. (1990), Separable concave minimization via partial outer approximation and branch and bound, *Operations Reserch Letters* 9: 389–394.
6. Cambini, A., Martein, L. and Schaible, S. (1989), On maximizing a sum of ratios, *Journal of Information and Optimization Sciences* 10: 65–79.
7. Colantoni, C.S., Manes, R.P. and Whinston, A. (1969), Programming, profit rates and pricing decisions, *The Accounting Review* 44: 467–481.
8. Dur, M., Horst, R. and Thoai, N.V. (2001), Solving sum-of-ratios fractional programs using efficient points, *Optimization*, to appear.
9. Falk, J.E. and Palocsay, S.W. (1994), Image space analysis of generalized fractional programs, *Journal of Global Optimization* 4: 63–88.
10. Fourer, R., Gay, D.M. and Kernighan, B.W. (1993), *AMPL, a Modeling Language for Mathematical Programming*, Boyd and Fraser Publishing Company, Danvers, MA.
11. Freund, R. and Jarre, F. (2001), Solving the sum-of-ratios problem by an interior point method, *Journal of Global Optimization* 19: 83–102.
12. Gallo, G. and Ulkucu, A. (1977), Bilinear programming: an exact algorithm, *Mathematical Programming* 12: 173–194.
13. Horst, R. and Pardalos, P.M. (eds) (1995), *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands.
14. Horst, R. and Tuy, H. (1993), *Global Optimization: Deterministic Approaches*, Springer Verlag, Berlin.
15. Kalantari, B. and Rosen, J.B. (1987), An algorithm for global minimization of linearly constrained concave quadratic functions, *Mathematics of Operations Research* 12: 544–561.
16. Konno, H. and Abe, N. (1999), Minimization of the sum of three linear fractional functions, *Journal of Global Optimization* 15: 419–432.
17. Konno, H. and Fukaishi, K. (2000), A branch and bound algorithm for solving low rank linear multiplicative and fractional programming problems, *Journal of Global Optimization* 18: 283–299.
18. Konno, H. and Inori, M. (1989), Bond portfolio optimization by bilinear fractional programming, *Journal of the Operations Research Society of Japan* 32: 143–158.
19. Konno, H., Kuno, T. and Yajima, Y. (1994), Global minimization of a generalized convex multiplicative function, *Journal of Global Optimization* 4: 47–62.
20. Konno, H. and Watanabe, H. (1996), Bond portfolio optimization problems and their application to index tracking: a partial optimization approach, *Journal of the Operations Research Society of Japan* 39: 295–306.
21. Konno, H., Yajima, Y. and Matsui, T. (1991), Parametric simplex algorithms for solving a special class of nonconvex minimization problems, *Journal of Global Optimization* 1: 65–81.
22. Konno, H. and Yamashita, H. (1999), Minimizing sums and products of linear fractional functions over a polytope, *Naval Research Logistics* 46: 583–596.
23. Kuno, T. (2000), A branch-and-bound algorithm for maximizing the sum of several linear ratios, University of Tsukuba, Research Report ISE-TR-00-175, Tsukuba, Japan.
24. Murtagh, B.A. and Saunders, M.A. (1998), MINOS 5.5 User's Guide, Stanford University, Technical Report SOL 83-20R, Stanford, CA.
25. Quesada, I. and Grossman, I. (1995), A global optimization algorithm for linear fractional and bilinear programs, *Journal of Global Optimization* 6: 39–76.
26. Rosen, J.B. and Pardalos, P.M. (1986), Global minimization of large-scale constrained concave quadratic problems by separable programming, *Mathematical Programming* 34: 163–174.
27. Schaible, S. (1995), Fractional programming, in: R. Horst and P.M. Pardalos (eds), *Handbook of Global Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 495–608.
28. Schaible, S. (1994), Fractional programming with sums of ratios, University of Pisa, Report No. 83, Pisa, Italy.

29. Schaible, S. (1977), A note on the sum of a linear and linear-fractional function, *Naval Research Logistics Quarterly* 24: 691–693.
30. Tuy, H. (1998), *Convex Analysis and Global Optimization*, Kluwer Academic Publishers, Dordrecht, The Netherlands.