# A practical but rigorous approach to sum-of-ratios optimization in geometric applications

Takahito Kuno* and Toshiyuki Masaki

*Graduate School of Systems and Information Engineering*

*University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan*

April 13, 2012

**Abstract**

In this paper, we develop an algorithm for minimizing the $L_q$ norm of a vector whose components are linear fractional functions, where $q$ is an arbitrary positive integer. The problem is a kind of sum-of-ratios optimization problem, and often occurs in computer vision. In that case, it is characterized by a large number of ratios and a small number of variables. The algorithm we propose here exploits this feature and generates a globally optimal solution in a practical amount of computational time.

**Key words:** Global optimization, sum-of-ratios optimization, branch-and-bound, computer vision, multiple-view geometry.

## 1 Introduction

Fractional optimization problems have been studied in order to achieve optimal economic performance, as evidenced by the fact that many of economic indicators such as rate of return and productivity are represented by fractional functions. Sum-of-ratios optimization, i.e., optimization of a sum of fractional functions, arises in problems of stochastic nature, where the objective is to maximize the expectation of economic performance (see e.g., [1, 14, 15]). Recently, in spite of its theoretical difficulty [9, 20], sum-of-ratios optimization has attracted much attention in multiple-view geometry of computer vision, without any direct relation to economic performance. Since multiple-view geometry is developed in projective spaces, fractional functions play an essential role as mediator between Euclidean and projective spaces.

*The author was partially supported by a Grant-in-Aid for Challenging Exploratory Research (22651057) from the Japan Society for the Promotion of Sciences. E-mail: takahito@cs.tsukuba.ac.jp

A variety of problems, e.g., triangulation, camera resectioning, homography estimation, and so forth (see e.g., [11, 13]), can be formulated into a class of sum-of-ratios optimization problems, where the objective is to minimize a norm of a vector of linear fractional functions. Problems of this class are characterized by a small number of variables but a large number of ratios, and also occur in other applications of geometric optimization such as layered manufacturing and material layout [2, 7, 18]. Unfortunately, except for some heuristics [7], all existing algorithms are totally inadequate to solve such kind of problems because those are designed for economic applications with only a few ratios [4, 5, 16, 17]. The purpose of this paper is to propose an algorithm for computing a globally optimal solution of sum-of-ratios optimization problems sharing the above mentioned feature, in a practical amount of computational time.

In the next section, we give a formal definition of the target sum-of-ratios optimization problem. We also illustrate how the problem arises in computer vision, and reveal its characteristics as a geometric problem. Based on this observation, we develop a special-purpose branch-and-bound algorithm in Section 3. We first devise an linear programming relaxation for bounding, and then propose a convergent subdivision rule for branching. After providing a detailed description of the algorithm, we prove its correctness in the rest of the section. Lastly, we report some numerical results of the algorithm, and conclude the paper in Section 4.

## 2 Sum-of-ratios optimization problem

The problem considered in this paper is a class of fractional optimization problems, often called the *sum-of-ratios optimization problem*:

$$\left|\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{r} \left|\frac{\mathbf{c}_i^{\top}\mathbf{x}+\gamma_i}{\mathbf{d}_i^{\top}\mathbf{x}+\delta_i}\right|^q \\ \text{subject to} & \mathbf{A}\mathbf{x} \geq \mathbf{b}, \quad \mathbf{0} \leq \mathbf{x} \leq \mathbf{v}, \end{array}\right. \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{b} \in \mathbb{R}^m$, $\mathbf{c}_i, \mathbf{d}_i, \mathbf{v} \in \mathbb{R}^n$, $\gamma_i, \delta_i \in \mathbb{R}^1$, and $q$ is a positive integer. Let us denote the feasible set by

$$D = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{A}\mathbf{x} \geq \mathbf{b},\ \mathbf{0} \leq \mathbf{x} \leq \mathbf{v}\},$$

and assume throughout the paper that $D \neq \emptyset$ and

$$\mathbf{d}_i^{\top}\mathbf{x} + \delta_i > 0, \quad i = 1, \ldots, q, \quad \forall \mathbf{x} \in D. \tag{2}$$

If $q = 1$, then (1) is a linear sum-of-ratios optimization problem, for which branch-and-bound algorithms have been proposed in [16, 17]. When $q = 2$, problem (1) is a special case of nonlinear sum-of-ratios optimization problem:

$$\left|\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{r} \frac{f_i(\mathbf{x})}{g_i(\mathbf{x})} \\ \text{subject to} & \mathbf{x} \in D, \end{array}\right. \tag{3}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}^1$ is a convex function, and $g_i : \mathbb{R}^n \to \mathbb{R}^1$ is concave and positive on $D$. As shown in [13], these conditions are satisfied by setting

$$g_i(\mathbf{x}) = \mathbf{d}_i^\top \mathbf{x} + \delta_i, \quad f_i(\mathbf{x}) = \frac{(\mathbf{c}_i^\top \mathbf{x} + \gamma_i)^2}{\mathbf{d}_i^\top \mathbf{x} + \delta_i}, \quad i = 1, \dots, q.$$

For problem (3), branch-and-bound algorithms similar to the one in [16] have also been developed in [4, 5]. The number $q$ of ratios that can be handled by those existing deterministic algorithms is limited to only around ten, at the present time. The difficulty of (1) is attributed to the *sum* of ratios, not due to ratios themselves. To see this, consider the simplest case where $q = 1$. It is known that a linear ratio is a quasiconvex and quasiconcave function on the domain where the denominator is positive (see e.g., [19]). This characteristic leads to an efficient solution to linear fractional programming problems, as shown by Charnes and Cooper [6]. However, the sum of quasiconvex functions is not in general quasiconvex, and the sum of quasiconcave functions is not quasiconcave. These imply that the sum of linear ratios is neither a quasiconvex nor a quasiconcave function. In consequence, (1) can have multiple local minima different from global minima, not only at vertices of $D$, even when $q = 1$. From the viewpoint of computational complexity, (1) is known to be $\mathcal{NP}$-hard [9, 20].

### 2.1 Sum-of-ratios optimization in computer vision

The problem (1), although difficult to solve, has a wide variety of applications in computer vision dealing with geometric relations between the three-dimensional world and its projection onto a two-dimensional image plane. In this section, we take *triangulation* as a typical example and show how it can be formulated into (1). Essential to this formulation is the *pinhole camera model*.

**Pinhole camera model:** The pinhole camera model describes the relationship between the coordinates of a three-dimensional point and its projection onto the image plane of an ideal pinhole camera, where the camera aperture is a pinhole and no lenses are used to focus light. The geometry related to the mapping of a pinhole camera is illustrated in Figure 1. Let us denote the object of shooting by $\mathbf{x}' = (x'_1, x'_2, x'_3)^\top$ in the three-dimensional coordinate system with its origin at the camera aperture $\mathbf{o}$. Light emanating from $\mathbf{x}'$ passes through $\mathbf{o}$ and projects

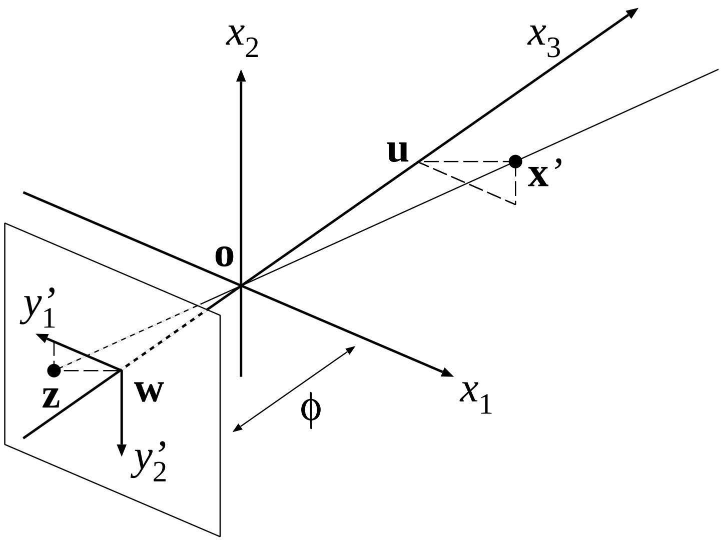


Figure 1: Geometry of a pinhole camera.

an inverted image $\mathbf{y}' = (y_1', y_2')^\top$ on the image plane, which is parallel to the $x_1$-$x_2$ plane and located at the focal length $\phi$ ($> 0$) from $\mathbf{o}$ in the negative direction of the $x_3$ axis. Let $\mathbf{u} = (0, 0, x_3')^\top$, $\mathbf{w} = (0, 0, -\phi)^\top$ and $\mathbf{z} = (y_1', y_2', -\phi)^\top$. Since the triangle connecting three points $\mathbf{o}, \mathbf{u}$ and $\mathbf{x}'$ is similar to that connecting $\mathbf{o}, \mathbf{w}$ and $\mathbf{z}$, we have $(y_1', y_2')^\top = (\phi/x_3')(x_1', x_2')^\top$, or equivalently

$$\begin{bmatrix} y_1' \\ y_2' \\ 1 \end{bmatrix} = \frac{\phi}{x_3'} \begin{bmatrix} x_1' \\ x_2' \\ x_3'/\phi \end{bmatrix}$$

in homogeneous coordinates. It should also be noted that the image $\mathbf{y}'$ is invariant under scaling of $\mathbf{x}'$. We denote this by

$$\begin{bmatrix} y_1' \\ y_2' \\ 1 \end{bmatrix} \sim \begin{bmatrix} x_1' \\ x_2' \\ x_3'/\phi \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/\phi & 0 \end{bmatrix} \begin{bmatrix} x_1' \\ x_2' \\ x_3' \\ 1 \end{bmatrix}, \tag{4}$$

and say that $(y_1', y_2', 1)^\top$ is *equivalent, or proportional, to* $(x_1', x_2', x_3'/\phi)^\top$. The $3 \times 4$ matrix in (4) is called the *camera matrix*.

**Triangulation:** Triangulation (or *reconstruction*) is the process of determining the three-dimensional coordinates of the object $\mathbf{x}'$, given its projections onto two, or more, images captured by pinhole cameras. In theory, the triangulation problem is quite trivial. Each image $\mathbf{y}'$ of $\mathbf{x}'$ corresponds to a half-line in the three-dimensional space such that all points on the

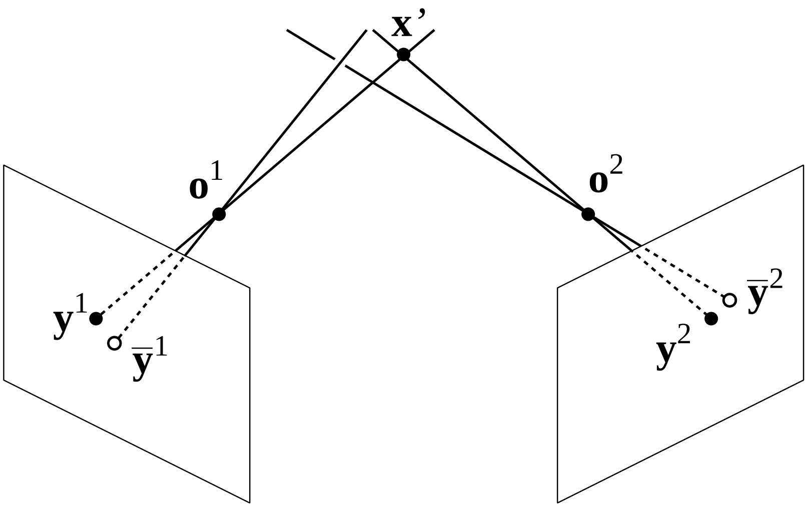


Figure 2: Triangulation from two images.

line are projected to $\mathbf{y}'$. Therefore, $\mathbf{x}'$ must lie on the intersection of those lines, and we must be able to calculate its coordinates analytically from a pair of different images. In practice, however, various types of noise, such as geometric noise from lens distortion or interest point detection error, lead to inaccuracies in the measured image coordinates. As a result, lines associated with different images of $\mathbf{x}'$ do not always intersect in the three-dimensional space, as is shown in Figure 2.

Suppose that $\mathbf{x}' = (x'_1, x'_2, x'_3)^\top$ is in an arbitrary three-dimensional coordinate system, and that there are $N$ images $\mathbf{y}^i = (y^i_1, y^i_2)^\top$ of $\mathbf{x}'$ captured by cameras $i = 1, \ldots, N$. Let us denote the $i$th camera matrix by

$$\mathbf{C}^i_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/\phi^i & 0 \end{bmatrix},$$

where $\phi^i$ ($> 0$) is the focal length of camera $i$. Note that $\mathbf{x}'$ is denoted as $\mathbf{R}^i\mathbf{x}' + \mathbf{t}^i$ for some rotation matrix $\mathbf{R}^i$ and a translation vector $\mathbf{t}^i$ in the three-dimensional coordinate system with the origin at the focal point $\mathbf{o}^i$ of camera $i$. Hence, from (4), we have

$$\begin{bmatrix} \mathbf{y}^i \\ 1 \end{bmatrix} \sim \mathbf{C}^i_0 \begin{bmatrix} \mathbf{R}^i & \mathbf{t}^i \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}' \\ 1 \end{bmatrix}, \quad i = 1, \ldots, N.$$

Let

$$\mathbf{C}^i = \begin{bmatrix} \mathbf{c}^i_1 & \gamma^i_1 \\ \mathbf{c}^i_2 & \gamma^i_2 \\ \mathbf{c}^i_3 & \gamma^i_3 \end{bmatrix} = \mathbf{C}^i_0 \begin{bmatrix} \mathbf{R}^i & \mathbf{t}^i \\ \mathbf{0} & 1 \end{bmatrix},$$

which is referred to as the *normalized camera matrix*. The coordinates of the image $\mathbf{y}^i$ is then given as

$$y_1^i = \frac{\mathbf{c}_1^i \mathbf{x}' + \gamma_1^i}{\mathbf{c}_3^i \mathbf{x}' + \gamma_3^i}, \quad y_2^i = \frac{\mathbf{c}_2^i \mathbf{x}' + \gamma_2^i}{\mathbf{c}_3^i \mathbf{x}' + \gamma_3^i},$$

if there is no noise. As mentioned above, however, this is not the case in practice, and we need to determine the coordinates $(x_1, x_2, x_3)^\top$ of $\mathbf{x}'$ so as to minimize the *reprojection residual*, defined below, between each $\mathbf{y}^i$ and the measurement $\bar{\mathbf{y}}^i$:

$$r_j^i(\mathbf{x}) = \left| \frac{\mathbf{c}_j^i \mathbf{x} + \gamma_j^i}{\mathbf{c}_3^i \mathbf{x} + \gamma_3^i} - \bar{y}_j^i \right|, \quad i = 1, \dots, N; \ j = 1, 2.$$

If we adopt the $L^1$ or $L^2$ norm criterion, the problem to be solved is as follows:

$$\left| \begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^N \sum_{j=1}^2 \left( r_j^i(\mathbf{x}) \right)^q \\ \text{subject to} & \mathbf{c}_3^i \mathbf{x} + \gamma_3^i \geq 0, \quad i = 1, \dots, N, \end{array} \right. \tag{5}$$

where $q = 1$ or 2, depending on the adopted norm. Since

$$\sum_{i=1}^N \sum_{j=1}^2 \left( r_j^i(\mathbf{x}) \right)^q = \sum_{i=1}^N \sum_{j=1}^2 \left| \frac{(\mathbf{c}_j^i - \bar{y}_j^i \mathbf{c}_3^i)\mathbf{x} + \gamma_j^i - \bar{y}_j^i \gamma_3^i}{\mathbf{c}_3^i \mathbf{x} + \gamma_3^i} \right|^q,$$

problem (5) is apparently a special case of (1). Besides triangulation, there are a number of problems formulated into (1), in computer vision, especially in connection with multiple-view geometry. For more details, see e.g. [11, 13].

## 3 Practical branch-and-bound algorithm

Geometric applications of (1), such as in computer vision and in [2, 7, 18], share a common characteristic that the number of ratios is large but the number of variables is small. For example, triangulation assumed in [3] uses more than a hundred cameras, and in that case, (1) has more than two hundreds ratios in the objective function whereas the number of variables is only three. By exploiting this feature, we will develop below a special-purpose branch-and-bound algorithm, which performs branching in the variable space and converges to a globally optimal solution of (1). First, we will derive a linear programming relaxation for the bounding operation.

### 3.1 LINEAR PROGRAMMING RELAXATION

Consider the following subproblem of (1):

$$\mathrm{P}(\mathbf{l},\mathbf{u}) \left| \begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{r} \left| \frac{\mathbf{c}_i^{\mathsf{T}}\mathbf{x}+\gamma_i}{\mathbf{d}_i^{\mathsf{T}}\mathbf{x}+\delta_i} \right|^q \\ \text{subject to} & \mathbf{A}\mathbf{x} \geq \mathbf{b}, \quad \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, \end{array} \right.$$

where $\mathbf{0} \leq \mathbf{l} \leq \mathbf{u} \leq \mathbf{v}$. Let us try applying the Charnes-Cooper transformation [6] to $\mathrm{P}(\mathbf{l},\mathbf{u})$, by introducing auxiliary variables:

$$\mathbf{y}_i = \eta_i \mathbf{x}, \quad \eta_i = \frac{1}{\mathbf{d}_i^{\mathsf{T}}\mathbf{x}+\delta_i}, \quad i = 1, \ldots, r.$$

Then we have

$$\left| \begin{array}{lll} \text{minimize} & \displaystyle\sum_{i=1}^{r} |\mathbf{c}_i^{\mathsf{T}}\mathbf{y}_i + \gamma_i\eta_i|^q & \\ \text{subject to} & \left. \begin{array}{l} \mathbf{A}\mathbf{y}_i - \mathbf{b}\eta_i \geq \mathbf{0} \\ \mathbf{d}_i^{\mathsf{T}}\mathbf{y}_i + \delta_i\eta_i = 1 \\ \mathbf{y}_i = \eta_i\mathbf{x} \\ \mathbf{y}_i \geq \mathbf{0}, \quad \eta_i \geq 0 \end{array} \right\} & i = 1, \ldots, r \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, & \end{array} \right. \tag{6}$$

which is equivalent to $\mathrm{P}(\mathbf{l},\mathbf{u})$ in the following sense.

**Proposition 3.1.** *If* $\mathbf{x}^*$ *is an optimal solution of* $\mathrm{P}(\mathbf{l},\mathbf{u})$*, then* $(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\eta}^*)$ *with* $\mathbf{y}_i^* = \eta_i^*\mathbf{x}^*$ *and* $\eta_i^* = 1/(\mathbf{d}_i^{\mathsf{T}}\mathbf{x}^* + \delta_i)$ *is an optimal solution of (6). Conversely, if* $(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\eta}^*)$ *is an optimal solution of (6), then* $\mathbf{x}^*$ *is an optimal solution of* $\mathrm{P}(\mathbf{l},\mathbf{u})$.

*Proof.* Let $\mathbf{x}^*$ be an optimal solution of $\mathrm{P}(\mathbf{l},\mathbf{u})$. By assumption (2), we have $\mathbf{d}_i^{\mathsf{T}}\mathbf{x}^* + \delta_i > 0$ for $i = 1, \ldots, q$. We can also see that $(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\eta}^*)$ with $\mathbf{y}_i^* = \eta_i^*\mathbf{x}^*$ and $\eta_i^* = 1/(\mathbf{d}_i^{\mathsf{T}}\mathbf{x}^* + \delta_i)$ is a feasible solution of (6). Suppose there is a feasible solution $(\mathbf{x}', \mathbf{y}', \boldsymbol{\eta}')$ of (6) such that

$$\sum_{i=1}^{r} |\mathbf{c}_i^{\mathsf{T}}\mathbf{y}_i' + \gamma_i\eta_i'|^q < \sum_{i=1}^{r} |\mathbf{c}_i^{\mathsf{T}}\mathbf{y}_i^* + \gamma_i\eta_i^*|^q. \tag{7}$$

However, we have

$$\sum_{i=1}^{r} |\mathbf{c}_i^{\mathsf{T}}\mathbf{y}_i' + \gamma_i\eta_i'|^q = \sum_{i=1}^{r} \left| \frac{\mathbf{c}_i^{\mathsf{T}}\mathbf{x}'+\gamma_i}{\mathbf{d}_i^{\mathsf{T}}\mathbf{x}'+\delta_i} \right|^q, \quad \sum_{i=1}^{r} |\mathbf{c}_i^{\mathsf{T}}\mathbf{y}_i^* + \gamma_i\eta_i^*|^q = \sum_{i=1}^{r} \left| \frac{\mathbf{c}_i^{\mathsf{T}}\mathbf{x}^*+\gamma_i}{\mathbf{d}_i^{\mathsf{T}}\mathbf{x}^*+\delta_i} \right|^q,$$

and besides $\mathbf{x}'$ is a feasible solution of $\mathrm{P}(\mathbf{l},\mathbf{u})$. Therefore, (7) contradicts the optimality of $\mathbf{x}^*$ for $\mathrm{P}(\mathbf{l},\mathbf{u})$. The converse can be proven similarly. □

While the feasible set of $\mathrm{P}(\mathbf{l},\mathbf{u})$ is a polyhedron, this is no longer the case for problem (6). Yet it can easily be relaxed into a polyhedral set as follows, by eliminating $\mathbf{x}$ from (6):

$$\widetilde{\mathrm{P}}(\mathbf{l},\mathbf{u}) \left| \begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{r} \left|\mathbf{c}_i^{\mathsf{T}}\mathbf{y}_i + \gamma_i\eta_i\right|^q \\ \text{subject to} & \left.\begin{array}{l} \mathbf{A}\mathbf{y}_i - \mathbf{b}\eta_i \geq \mathbf{0} \\ \mathbf{d}_i^{\mathsf{T}}\mathbf{y}_i + \delta_i\eta_i = 1 \\ \mathbf{l}\eta_i \leq \mathbf{y}_i \leq \mathbf{u}\eta_i \\ \mathbf{y}_i \geq \mathbf{0}, \quad \eta_i \geq 0 \end{array}\right\} \quad i = 1,\ldots,r. \end{array}\right.$$

**Proposition 3.2.** *If* $\mathrm{P}(\mathbf{l},\mathbf{u})$ *has an optimal solution* $\mathbf{x}^*$*, then* $\widetilde{\mathrm{P}}(\mathbf{l},\mathbf{u})$ *also has an optimal solution* $(\widetilde{\mathbf{y}},\widetilde{\boldsymbol{\eta}})$*, which satisfies*

$$\sum_{i=1}^{r} \left|\mathbf{c}_i^{\mathsf{T}}\widetilde{\mathbf{y}}_i + \gamma_i\widetilde{\eta}_i\right|^q \leq \sum_{i=1}^{r} \left|\frac{\mathbf{c}_i^{\mathsf{T}}\mathbf{x}^* + \gamma_i}{\mathbf{d}_i^{\mathsf{T}}\mathbf{x}^* + \delta_i}\right|^q. \tag{8}$$

*Moreover, if* $(\widetilde{\mathbf{y}},\widetilde{\boldsymbol{\eta}})$ *satisfies*

$$\frac{1}{\widetilde{\eta}_1}\widetilde{\mathbf{y}}_1 = \cdots = \frac{1}{\widetilde{\eta}_r}\widetilde{\mathbf{y}}_r, \tag{9}$$

*then* $(1/\widetilde{\eta}_i)\widetilde{\mathbf{y}}_i$ *is an optimal solution of* $\mathrm{P}(\mathbf{l},\mathbf{u})$ *for any* $i$*.*

*Proof.* Let $\mathbf{x}^*$ be an optimal solution of $\mathrm{P}(\mathbf{l},\mathbf{u})$. Letting $\mathbf{y}_i^* = \eta_i^*\mathbf{x}^*$ and $\eta_i^* = 1/(\mathbf{d}_i^{\mathsf{T}}\mathbf{x}^* + \delta_i)$, we have $\mathbf{l}\eta_i^* \leq \mathbf{x}^*\eta_i^* \leq \mathbf{u}\eta_i^*$, $\mathbf{y}_i^* = \mathbf{x}^*\eta_i^*$, and hence $(\mathbf{y}^*,\boldsymbol{\eta}^*)$ is a feasible solution of $\widetilde{\mathrm{P}}(\mathbf{l},\mathbf{u})$. Since the objective function has an obvious lower bound, zero, $\widetilde{\mathrm{P}}(\mathbf{l},\mathbf{u})$ must have an optimal solution $(\widetilde{\mathbf{y}},\widetilde{\boldsymbol{\eta}})$, which satisfies (8). Note that $\widetilde{\boldsymbol{\eta}} > \mathbf{0}$; otherwise, $\widetilde{\eta}_i = 0$, $\widetilde{\mathbf{y}}_i = \mathbf{0}$, and hence $\mathbf{d}_i^{\mathsf{T}}\widetilde{\mathbf{y}}_i + \delta\widetilde{\eta}_i \neq 1$ for some $i$. If $(\widetilde{\mathbf{y}},\widetilde{\boldsymbol{\eta}})$ satisfies (9), then $(1/\widetilde{\eta}_i)\widetilde{\mathbf{y}}_i$ is a feasible solution of $\mathrm{P}(\mathbf{l},\mathbf{u})$, and its optimality follows from (8). $\square$

A further point to note on $\widetilde{\mathrm{P}}(\mathbf{l},\mathbf{u})$ is that it is decomposable into $r$ problems, each of which is of the form:

$$\left| \begin{array}{ll} \text{minimize} & \left|\mathbf{c}_i^{\mathsf{T}}\mathbf{y}_i + \gamma_i\eta_i\right|^q \\ \text{subject to} & \mathbf{A}\mathbf{y}_i - \mathbf{b}\eta_i \geq \mathbf{0} \\ & \mathbf{d}_i^{\mathsf{T}}\mathbf{y}_i + \delta_i\eta_i = 1 \\ & \mathbf{l}\eta_i \leq \mathbf{y}_i \leq \mathbf{u}\eta_i \\ & \mathbf{y}_i \geq \mathbf{0}, \quad \eta_i \geq 0. \end{array}\right. \tag{10}$$

Introducing another auxiliary variable $\zeta_i = \left|\mathbf{c}_i^\mathsf{T}\mathbf{y}_i + \gamma_i\eta_i\right|$, we can rewrite (10) into

$$
\left|\begin{array}{ll}
\text{minimize} & \zeta_i^q \\
\text{subject to} & \mathbf{A}\mathbf{y}_i - \mathbf{b}\eta_i \geq \mathbf{0} \\
& \mathbf{d}_i^\mathsf{T}\mathbf{y}_i + \delta_i\eta_i = 1 \\
& -\zeta_i \leq \mathbf{c}_i^\mathsf{T}\mathbf{y}_i + \gamma_i\eta_i \leq \zeta_i \\
& \mathbf{l}\eta_i \leq \mathbf{y}_i \leq \mathbf{u}\eta_i \\
& \mathbf{y}_i \geq \mathbf{0}, \quad \eta_i \geq 0, \quad \zeta_i \geq 0.
\end{array}\right.
$$

To minimize $\zeta_i^q$, we only need to minimize $\zeta_i$, regardless of the magnitude of $q$, because $\zeta_i$ is restricted to be nonnegative. Therefore, to solve (10), we may solve the following:

$$
\mathrm{Q}_i(\mathbf{l},\mathbf{u}) \left|\begin{array}{ll}
\text{minimize} & \zeta_i \\
\text{subject to} & \mathbf{A}\mathbf{y}_i - \mathbf{b}\eta_i \geq \mathbf{0} \\
& \mathbf{d}_i^\mathsf{T}\mathbf{y}_i + \delta_i\eta_i = 1 \\
& -\zeta_i \leq \mathbf{c}_i^\mathsf{T}\mathbf{y}_i + \gamma_i\eta_i \leq \zeta_i \\
& \mathbf{l}\eta_i \leq \mathbf{y}_i \leq \mathbf{u}\eta_i \\
& \mathbf{y}_i \geq \mathbf{0}, \quad \eta_i \geq 0, \quad \zeta_i \geq 0.
\end{array}\right.
$$

This implies that, even though $\widetilde{\mathrm{P}}(\mathbf{l},\mathbf{u})$ is a nonlinear optimization problem with $(n+1)r$ variables, it can be solved by solving $r$ linear programming problems, each with $n+2$ variables.

**Proposition 3.3.** *The relaxed problem* $\widetilde{\mathrm{P}}(\mathbf{l},\mathbf{u})$ *has an optimal solution* $(\widetilde{\mathbf{y}},\widetilde{\boldsymbol{\eta}})$ *if and only if the linear programming problem* $\mathrm{Q}_i(\mathbf{l},\mathbf{u})$ *has an optimal solution* $(\widetilde{\mathbf{y}}_i,\widetilde{\eta}_i,\widetilde{\zeta}_i)$ *for* $i = 1,\ldots,r$.

*Proof.* Obvious from the above observation. □

Thus, we can decide whether the subproblem $\mathrm{P}(\mathbf{l},\mathbf{u})$ is worth solving or not, by solving linear programming problems $\mathrm{Q}_i(\mathbf{l},\mathbf{u})$ for $i = 1,\ldots,r$. More precisely, if $\sum_{i=1}^r \widetilde{\zeta}_i^q \geq z^*$ holds for the value $z^*$ of the best feasible solution obtained so far, we can leave $\mathrm{P}(\mathbf{l},\mathbf{u})$ out of consideration because there are no better solutions in the intersection of $D$ with

$$[\mathbf{l},\mathbf{u}] = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\}.$$

Otherwise, we have to examine subproblems of the subproblem $\mathrm{P}(\mathbf{l},\mathbf{u})$ after an adequate branching operation.

### 3.2 Convergent subdivision rule

One way to generate subproblems of $\mathrm{P}(\mathbf{l},\mathbf{u})$ is naturally to subdivide the rectangle $[\mathbf{l},\mathbf{u}]$ into a set of subrectangles. If we divide $[\mathbf{l},\mathbf{u}]$ along $x_k = (l_k+u_k)/2$ for $k \in \arg\max\{u_j - l_j \mid j =$

$1,\dots,n\}$, the algorithm is guaranteed to be convergent just like the usual rectangular branch-and-bound algorithm for concave minimization based on the bisection rule [8, 12, 22]. Instead of such an exhaustive method, we will propose here a more sophisticated subdivision rule for branching.

For any optimal solution $(\widetilde{\mathbf{y}}, \widetilde{\boldsymbol{\eta}})$ of $\widetilde{\mathrm{P}}(\mathbf{l}, \mathbf{u})$, let

$$\widetilde{\mathbf{x}}_i = \frac{1}{\widetilde{\eta}_i}\widetilde{\mathbf{y}}_i, \quad i = 1,\dots,r. \tag{11}$$

If $\widetilde{\mathbf{x}}_1 = \cdots = \widetilde{\mathbf{x}}_r$, then $\mathbf{x}_i$ is an optimal solution of $\mathrm{P}(\mathbf{l}, \mathbf{u})$ for any $i$, by Proposition 3.2. Even if not, $\widetilde{\mathbf{x}}_i$'s are all feasible for $\mathrm{P}(\mathbf{l}, \mathbf{u})$, and so is their centroid

$$\boldsymbol{\omega} = \frac{1}{r}\sum_{i=1}^{r}\widetilde{\mathbf{x}}_i. \tag{12}$$

By means of this point $\boldsymbol{\omega} \in [\mathbf{l}, \mathbf{u}]$, we can subdivide the rectangle, just as under the $\omega$-subdivision rule for the rectangular branch-and-bound algorithm [8, 12, 22]. Namely, let

$$\rho_j = \min\{u_j - \omega_j, \omega_j - l_j\}, \quad j = 1,\dots,n, \tag{13}$$

and select

$$j' \in \arg\max\{\rho_j \mid j = 1,\dots,n\}. \tag{14}$$

Then we may divide $[\mathbf{l}, \mathbf{u}]$ into $[\mathbf{l}', \mathbf{u}]$ and $[\mathbf{l}, \mathbf{u}']$, where

$$l'_j = \begin{cases} \omega_j, & \text{if } j = j' \\ l_j, & \text{otherwise,} \end{cases} \qquad u'_j = \begin{cases} \omega_j, & \text{if } j = j' \\ u_j, & \text{otherwise.} \end{cases} \tag{15}$$

If we apply the same operation to both subrectangles $[\mathbf{l}', \mathbf{u}]$ and $[\mathbf{l}, \mathbf{u}']$ recursively, the following sequences are generated:

$$\mathbf{l}^k \le \mathbf{l}^{k+1} \le \boldsymbol{\omega}^{k+1} \le \mathbf{u}^{k+1} \le \mathbf{u}^k, \quad k = 1,2,\dots, \tag{16}$$

where $[\mathbf{l}^1, \mathbf{u}^1] = [\mathbf{l}, \mathbf{u}]$, and $\omega^k_{j^k}$ is equal to either $l^{k+1}_{j^k}$ or $u^{k+1}_{j^k}$. For each $k$, equations (11) and (12) yield $\widetilde{\mathbf{x}}^k_i$'s and $\boldsymbol{\omega}^k$ from an optimal solution $(\widetilde{\mathbf{y}}^k, \widetilde{\boldsymbol{\eta}}^k)$ of $\widetilde{\mathrm{P}}(\mathbf{l}^k, \mathbf{u}^k)$.

**Lemma 3.4.** *There exist points $\bar{\mathbf{l}}$ and $\bar{\mathbf{u}}$ in $[\mathbf{l}^1, \mathbf{u}^1]$ such that $\bar{\mathbf{l}} \le \bar{\mathbf{u}}$, and as $k \to +\infty$,*

$$\mathbf{l}^k \to \bar{\mathbf{l}}, \quad \mathbf{u}^k \to \bar{\mathbf{u}}.$$

*The sequence $\{\boldsymbol{\omega}^k\}$ also has accumulation points, each of which lies on a corner of the limit rectangle $[\bar{\mathbf{l}}, \bar{\mathbf{u}}]$.*

*Proof.* We see from (16) that for each $j$ both sequences $\{l_j^k\}$ and $\{u_j^k\}$ are monotonic, bounded, and hence have limits $\bar{l}_j$ and $\bar{u}_j$, respectively, such that $\bar{l}_j \le \bar{u}_j$. As for $\{\boldsymbol{\omega}^k\}$, it is generated in the compact set $[\mathbf{l}^1, \mathbf{u}^1]$ and has at least one accumulation point. Let $\overline{\boldsymbol{\omega}}$ be an arbitrary accumulation point and $\{\boldsymbol{\omega}^{k_s}\}$ a subsequence converging to $\overline{\boldsymbol{\omega}}$. Since $\{1, \ldots, n\}$ is a finite set, there is an index $t \in \{1, \ldots, n\}$ such that $j^{k_s} = t$ for infinitely many $s$. Therefore, by noting that $\omega_t^{k_s} \in \{l_t^{k_s+1}, u_t^{k_s+1}\}$, we have $\omega_t^{k_s} \to \overline{\omega}_t \in \{\bar{l}_t, \bar{u}_t\}$ as $s \to +\infty$. We also see from (13) and (14) that

$$\min\{u_t^{k_s} - \omega_t^{k_s}, \omega_t^{k_s} - l_t^{k_s}\} \ge \min\{u_j^{k_s} - \omega_j^{k_s}, \omega_j^{k_s} - l_j^{k_s}\}, \quad j = 1, \ldots, n.$$

The left-hand side converges to zero, and so does the right-hand side. This implies that $\overline{\boldsymbol{\omega}}$ is a corner point of the rectangle $[\bar{\mathbf{l}}, \bar{\mathbf{u}}]$. □

**Lemma 3.5.** *Let $\overline{\boldsymbol{\omega}}$ be any accumulation point of $\{\boldsymbol{\omega}^k\}$, and $\{\boldsymbol{\omega}^{k_s}\}$ a subsequence converging to $\overline{\boldsymbol{\omega}}$. Then, as $s \to +\infty$,*

$$\widetilde{\mathbf{x}}_i^{k_s} \to \overline{\boldsymbol{\omega}}, \quad i = 1, \ldots, r.$$

*Proof.* As shown in the preceding lemma, $\overline{\omega}_j \in \{\bar{l}_j, \bar{u}_j\}$ for $j = 1, \ldots, n$. Since $l_j^{k_s} \le \widetilde{x}_{ij}^{k_s}$ for each $i$, we have

$$\omega_j^{k_s} - l_j^{k_s} = \frac{1}{r}\sum_{i=1}^{r}\left(\widetilde{x}_{ij}^{k_s} - l_j^{k_s}\right) \ge \frac{1}{r}\left(\widetilde{x}_{ij}^{k_s} - l_j^{k_s}\right) \ge 0,$$

which implies that $\widetilde{x}_{ij}^{k_s} - l_j^{k_s} \to 0$ if $\overline{\omega}_j = \bar{l}_j$. Similarly, $\widetilde{x}_{ij}^{k_s} - u_j^{k_s} \to 0$ if $\overline{\omega}_j = \bar{u}_j$. In either case, $\widetilde{x}_{ij}^{k_s} \to \overline{\omega}_j$, as $s \to +\infty$. □

### 3.3 Description of the algorithm

Starting from $[\mathbf{l}^1, \mathbf{u}^1] = [\mathbf{0}, \mathbf{v}]$, we solve the relaxed problem $\widetilde{\mathrm{P}}(\mathbf{l}^k, \mathbf{u}^k)$ successively for $k = 1, 2, \ldots$. The rectangle $[\mathbf{l}^k, \mathbf{u}^k]$ is discarded unless the value of $\widetilde{\mathrm{P}}(\mathbf{l}^k, \mathbf{u}^k)$ is less than the value of the incumbent $\mathbf{x}^k$, the best feasible solution of (1) obtained so far. Since $\boldsymbol{\omega}^k$ yielded by (11) and (12) from $(\widetilde{\mathbf{y}}^k, \widetilde{\boldsymbol{\eta}}^k)$ is feasible for $\mathrm{P}(\mathbf{l}^k, \mathbf{u}^k)$, and hence for (1), the incumbent $\mathbf{x}^k$ can be updated with $\boldsymbol{\omega}^k$ if necessary. This feasible solution $\boldsymbol{\omega}^k$ is also used to divide $[\mathbf{l}^k, \mathbf{u}^k]$ into two subrectangles, according to (13), (14) and (15). Let $\varepsilon \ge 0$ be a given tolerance. The algorithm is summarized as follows:

```
algorithm sum_ratio
begin
   for i = 1,...,r do begin
      solve Q_i(0,u) and obtain an optimal solution (ỹ_i, η̃_i, ζ̃_i);  x̃_i := (1/η̃_i)ỹ_i;
   end;
   β(0,u) := Σ_{i=1}^r ζ̃_i^q;  ω := (1/q) Σ_{i=1}^r x̃_i;
```

$\mathbf{x}^1 := \boldsymbol{\omega};\ z^1 := \sum_{i=1}^r |(\mathbf{c}^i\boldsymbol{\omega}+\gamma^i)/(\mathbf{d}^i\boldsymbol{\omega}+\delta^i)|^q;\ \mathscr{L} := \{[\mathbf{0},\mathbf{u}]\};\ k := 1;$
while $\mathscr{L} \neq \emptyset$ do begin
select a rectangle with the smallest value of $\beta$ from $\mathscr{L}$ and denote it by $[\mathbf{l}^k, \mathbf{u}^k]$;
if $z^k - \beta(\mathbf{l}^k,\mathbf{u}^k) \leq \varepsilon$ then $\mathscr{L} := \emptyset$;
else begin
let $\widetilde{\mathbf{x}}_i^k = (1/\widetilde{\eta}_i^k)\widetilde{\mathbf{y}}_i^k$ for the solution $(\widetilde{\mathbf{y}}_i^k, \widetilde{\eta}_i^k, \widetilde{\zeta}_i^k)$ of $\mathrm{Q}_i(\mathbf{l}^k,\mathbf{u}^k)$, and $\boldsymbol{\omega}^k = (1/q)\sum_{i=1}^r \widetilde{\mathbf{x}}_i^k$;
for $j = 1,\ldots,n$ do $\rho_j := \min\{u_j^k - \omega_j^k, \omega_j^k - l_j^k\}$;
select $j^k \in \arg\max\{\rho_j \mid j = 1,\ldots,n\}$;
for $p = 2k, 2k+1$ do $[\mathbf{l}^p,\mathbf{u}^p] := [\mathbf{l}^k,\mathbf{u}^k];\ l_{j^k}^{2k} := \omega_{j^k}^k;\ u_{j^k}^{2k+1} := \omega_{j^k}^k$;
for $p = 2k, 2k+1$ do begin
for $i = 1,\ldots,r$ do begin
solve $\mathrm{Q}_i(\mathbf{l}^p,\mathbf{u}^p)$ and obtain an optimal solution $(\widetilde{\mathbf{y}}_i, \widetilde{\eta}_i, \widetilde{\zeta}_i)$; $\widetilde{\mathbf{x}}_i := (1/\widetilde{\eta}_i)\widetilde{\mathbf{y}}_i$;
end;
$\beta(\mathbf{l}^p,\mathbf{u}^p) := \sum_{i=1}^r \widetilde{\zeta}_i^q;\ \boldsymbol{\omega}^p := (1/q)\sum_{i=1}^r \widetilde{\mathbf{x}}_i$;
$z^p := \sum_{i=1}^r |(\mathbf{c}^i\boldsymbol{\omega}^p+\gamma^i)/(\mathbf{d}^i\boldsymbol{\omega}^p+\delta^i)|^q$;
end;
select $p_k \in \arg\min\{z^p \mid p = k, 2k, 2k+1\};\ \mathbf{x}^{k+1} := \boldsymbol{\omega}^{p_k};\ z^{k+1} := z^{p_k}$;
$\mathscr{L} := \mathscr{L} \cup \{[\mathbf{l}^{2k},\mathbf{u}^{2k}], [\mathbf{l}^{2k+1},\mathbf{u}^{2k+1}]\} \setminus \{[\mathbf{l}^k,\mathbf{u}^k]\};\ k := k+1$
end
end
end;

**Theorem 3.6.** *Suppose $\varepsilon = 0$. If the algorithm* sum_ratio *terminates after k iterations, then $\mathbf{x}^k$ is an optimal solution of (1). If* sum_ratio *does not terminate, then every accumulation point of the sequence $\{\mathbf{x}^k \mid k = 1,2,\ldots\}$ is an optimal solution.*

*Proof.* Let us assume that sum_ratio does not terminate; otherwise, the claim is obvious. The algorithm then generates at least one infinite sequence of nested rectangles, such as those in (16). Renumbering the indices if necessary, we have $\{\mathbf{l}^k\}$ and $\{\mathbf{u}^k\}$ converging to $\bar{\mathbf{l}}$ and $\bar{\mathbf{u}}$, respectively. From the description of sum_ratio, it holds that

$$0 \leq \beta(\mathbf{l}^k,\mathbf{u}^k) < z^k \leq \sum_{i=1}^r \left|\frac{\mathbf{c}^i\boldsymbol{\omega}^k+\gamma^i}{\mathbf{d}^i\boldsymbol{\omega}^k+\delta^i}\right|^q, \quad k = 1,2,\ldots.$$

Note that $\{z^k\}$ is nonincreasing, bounded from below, and hence converges to some $\bar{z} \geq 0$. Let $\overline{\boldsymbol{\omega}}$ be any accumulation point of $\{\boldsymbol{\omega}^k\}$, and $\{\boldsymbol{\omega}^{k_s}\}$ a subsequence converging to $\overline{\boldsymbol{\omega}}$. Then, as $s \to +\infty$, we have

$$\sum_{i=1}^r \left|\frac{\mathbf{c}^i\boldsymbol{\omega}^{k_s}+\gamma^i}{\mathbf{d}^i\boldsymbol{\omega}^{k_s}+\delta^i}\right|^q - \beta(\mathbf{l}^{k_s},\mathbf{u}^{k_s}) \to 0,$$

because

$$\beta(\mathbf{l}^{k_s},\mathbf{u}^{k_s}) = \sum_{i=1}^{r} \left| \mathbf{c}^i \widetilde{\mathbf{y}}_i^{k_s} + \gamma^i \widetilde{\eta}_i^{k_s} \right|^q = \sum_{i=1}^{r} \left| \frac{\mathbf{c}^i (1/\widetilde{\eta}_i^{k_s}) \widetilde{\mathbf{y}}_i^{k_s} + \gamma^i}{\mathbf{d}^i (1/\widetilde{\eta}_i^{k_s}) \widetilde{\mathbf{y}}_i^{k_s} + \delta^i} \right|^q = \sum_{i=1}^{r} \left| \frac{\mathbf{c}^i \widetilde{\mathbf{x}}_i^{k_s} + \gamma^i}{\mathbf{d}^i \widetilde{\mathbf{x}}_i^{k_s} + \delta^i} \right|^q,$$

and $\widetilde{\mathbf{x}}_i^{k_s} \to \overline{\boldsymbol{\omega}}$ by Lemma 3.5. We see therefore that, as $s \to +\infty$,

$$\beta(\mathbf{l}^{k_s},\mathbf{u}^{k_s}),\ \sum_{i=1}^{r} \left| \frac{\mathbf{c}^i \boldsymbol{\omega}^{k_s} + \gamma^i}{\mathbf{d}^i \boldsymbol{\omega}^{k_s} + \delta^i} \right|^q \to \bar{z}.$$

Suppose

$$\exists \mathbf{x}' \in D, \quad \sum_{i=1}^{r} \left| \frac{\mathbf{c}_i^{\top} \mathbf{x}' + \gamma_i}{\mathbf{d}_i^{\top} \mathbf{x}' + \delta_i} \right|^q < \bar{z}. \tag{17}$$

At iteration $k_s$ for each $s$, this feasible solution $\mathbf{x}'$ belongs to some rectangle $[\mathbf{l},\mathbf{u}] \in \mathscr{L}$, and we have

$$\beta(\mathbf{l}^{k_s},\mathbf{u}^{k_s}) \le \beta(\mathbf{l},\mathbf{u}) \le \sum_{i=1}^{r} \left| \frac{\mathbf{c}_i^{\top} \mathbf{x}' + \gamma_i}{\mathbf{d}_i^{\top} \mathbf{x}' + \delta_i} \right|^q,$$

because $[\mathbf{l}^{k_s},\mathbf{u}^{k_s}]$ is selected as a rectangle with the smallest value of $\beta$. However, $\beta(\mathbf{l}^{k_s},\mathbf{u}^{k_s}) \to \bar{z}$, which contradicts (17). Hence, we have

$$\bar{z} \le \sum_{i=1}^{r} \left| \frac{\mathbf{c}_i^{\top} \mathbf{x} + \gamma_i}{\mathbf{d}_i^{\top} \mathbf{x} + \delta_i} \right|^q, \quad \forall \mathbf{x} \in D. \tag{18}$$

Also note that $D$ is compact, and any accumulation point of $\{\mathbf{x}^k\}$, say $\bar{\mathbf{x}}$, belongs to $D$. For any subsequence $\{\mathbf{x}^{k_t}\}$ converging to $\bar{\mathbf{x}}$, we have

$$\sum_{i=1}^{r} \left| \frac{\mathbf{c}_i^{\top} \mathbf{x}^{k_t} + \gamma_i}{\mathbf{d}_i^{\top} \mathbf{x}^{k_t} + \delta_i} \right|^q = z^{k_t} \to \bar{z},$$

as $t \to +\infty$, because $\{z^k\}$ is a convergent sequence. This, together with (18), implies that $\bar{\mathbf{x}}$ is an optimal solution of (1). □

**Corollary 3.7.** *If $\varepsilon > 0$, the algorithm* sum_ratio *terminates after finitely many iterations and yields an approximate solution $\mathbf{x}^k$ of (1) such that*

$$z^k = \sum_{i=1}^{r} \left| \frac{\mathbf{c}^i \mathbf{x}^k + \gamma^i}{\mathbf{d}^i \mathbf{x}^k + \delta^i} \right|^q \le \sum_{i=1}^{r} \left| \frac{\mathbf{c}^i \mathbf{x} + \gamma^i}{\mathbf{d}^i \mathbf{x} + \delta^i} \right|^q + \varepsilon, \quad \forall \mathbf{x} \in D. \tag{19}$$

*Proof.* Let $\{\boldsymbol{\omega}^{k_s}\}$ be a subsequence of $\{\boldsymbol{\omega}^k\}$ converging to $\overline{\boldsymbol{\omega}}$. Then, as $p \to +\infty$, we have

$$z^{k_s} - \beta(\mathbf{l}^{k_s},\mathbf{u}^{k_s}) \to 0,$$

as seen in the proof of Theorem 3.6. Therefore, if $\varepsilon > 0$, it holds at some iteration $k$ that $z^k - \beta(\mathbf{l}^k, \mathbf{u}^k) \le \varepsilon$. This termination criterion implies (19) because $\beta(\mathbf{l}^k, \mathbf{u}^k)$ represents a lower bound of (1). $\square$

## 4 Numerical results

In this section, we present numerical results obtained with the algorithm sum_ratio. According to the description given in the previous section, sum_ratio was coded in GNU Octave (version 3.0.5) [10], a MATLAB-like programming environment, and run on an AMD Opteron 256 (3.0GHz) single core processor. The problem used as a benchmark is of the form:

$$\left|\begin{array}{lll} \text{minimize} & \displaystyle\sum_{i=1}^{r} \left| \frac{\mathbf{d}^i\mathbf{x} + \delta^i}{\mathbf{c}^i\mathbf{x} + \gamma^i} \right|^2 & \\ \text{subject to} & \mathbf{c}^i\mathbf{x} + \gamma^i \ge 0, & i = 1, \ldots, r \\ & 0 \le x_j \le 10.0, & j = 1, \ldots, n, \end{array}\right. \tag{20}$$

which imitates triangulation from $r/2$ images when $n = 3$. All $c^i_j$, $\gamma^i$, $d^i_j$ and $\delta^i$ were generated randomly in the interval $[-0.5, 0.5]$, and ten instances were selected for each size $(n, r)$. As for the subdivision rule, in addition to our proposed rule, we also tested the usual bisection rule mentioned at the beginning of Section 3.2 for comparison; their respective computer programs are referred to as *$\omega$-subdivision* and *bisection*. To solve the linear programming problem $\mathrm{Q}_i(\mathbf{l}, \mathbf{u})$, we coded a revised simplex procedure from scratch in Octave, without using any procedures available in the optimization toolbox. Also, to prevent the convergence from being affected by the magnitude of the optimal value, we replaced the termination criterion $z^k - \beta(\mathbf{l}^k, \mathbf{u}^k) \le \varepsilon$ in sum_ratio by

$$(1 - \varepsilon)z^k - \beta(\mathbf{l}^k, \mathbf{u}^k) \le 0.$$

The comparison results between *$\omega$-subdivision* and *bisection* are demonstrated in Figures 3 - 6. Figure 3 shows the variation of the average computational time (in seconds) required by each program with $r$ ranging from 50 to 350 in 50 increments when $n = 3$ and $\varepsilon = 0.05$. For the same set of instances, Figure 4 shows the average number of branching operations, which is equal to the number of $\widetilde{\mathrm{P}}(\mathbf{l}, \mathbf{u})$'s solved in the course of computation. The solid lines represent the results by *$\omega$-subdivision*, and the dashed lines are those by *bisection*. Both computer programs behave similarly, but it is quite obvious that *$\omega$-subdivision* is much superior to *bisection*; in fact, the former requires less than half the computational time and number of branching operations required by the latter for each $r$. Another thing to note about both programs is that there is no tendency for the number of branching operations to increase with

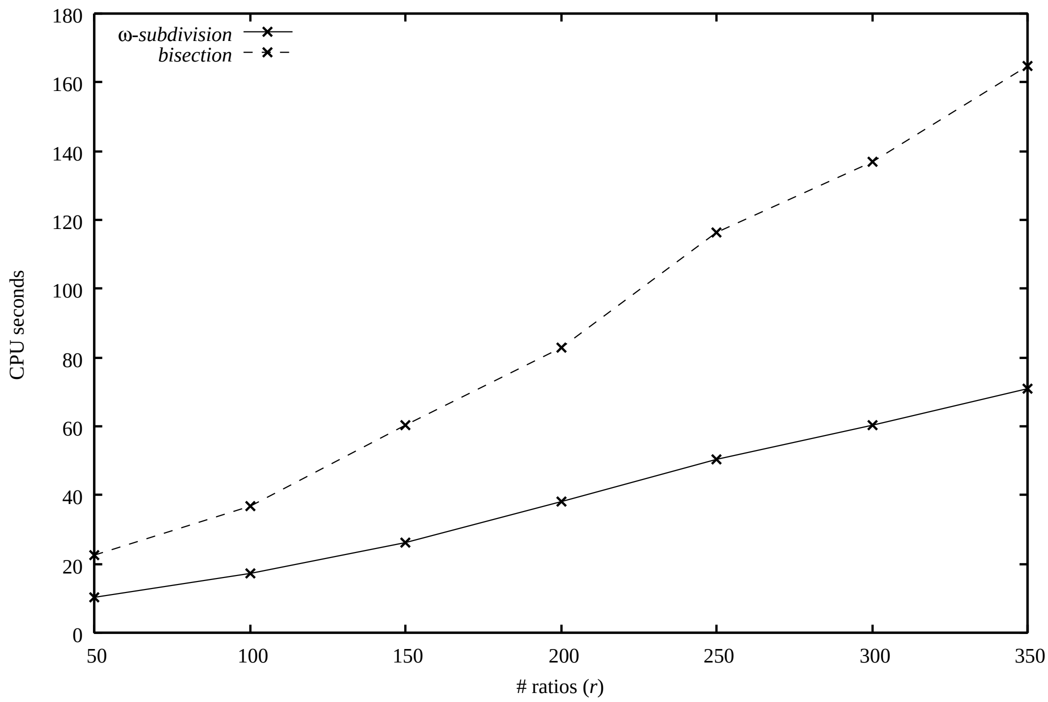


Figure 3: Average computational time in seconds when $n = 3$ and $\varepsilon = 0.05$.

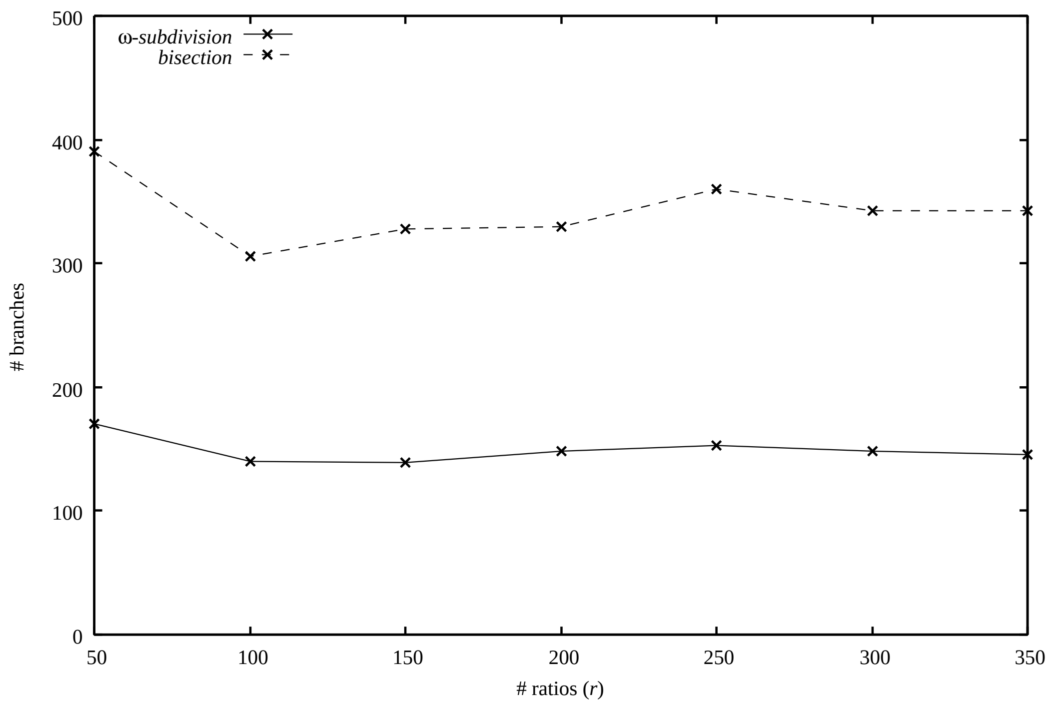


Figure 4: Average numbers of branching operations when $n = 3$ and $\varepsilon = 0.05$.

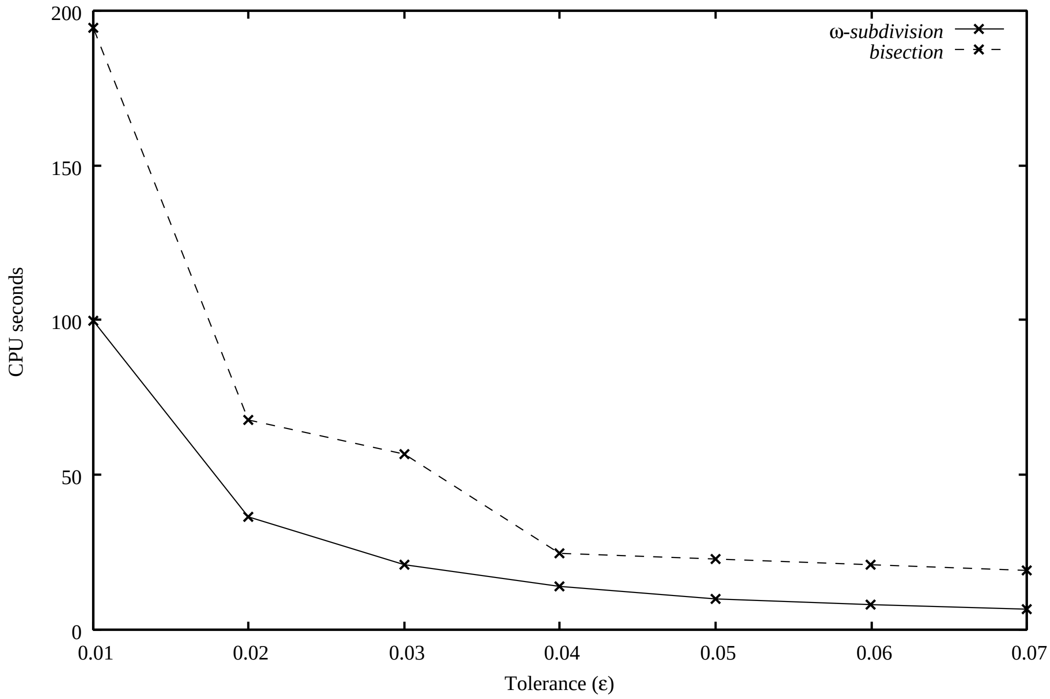


Figure 5: Average computational time in seconds when $(n, r) = (3, 50)$.

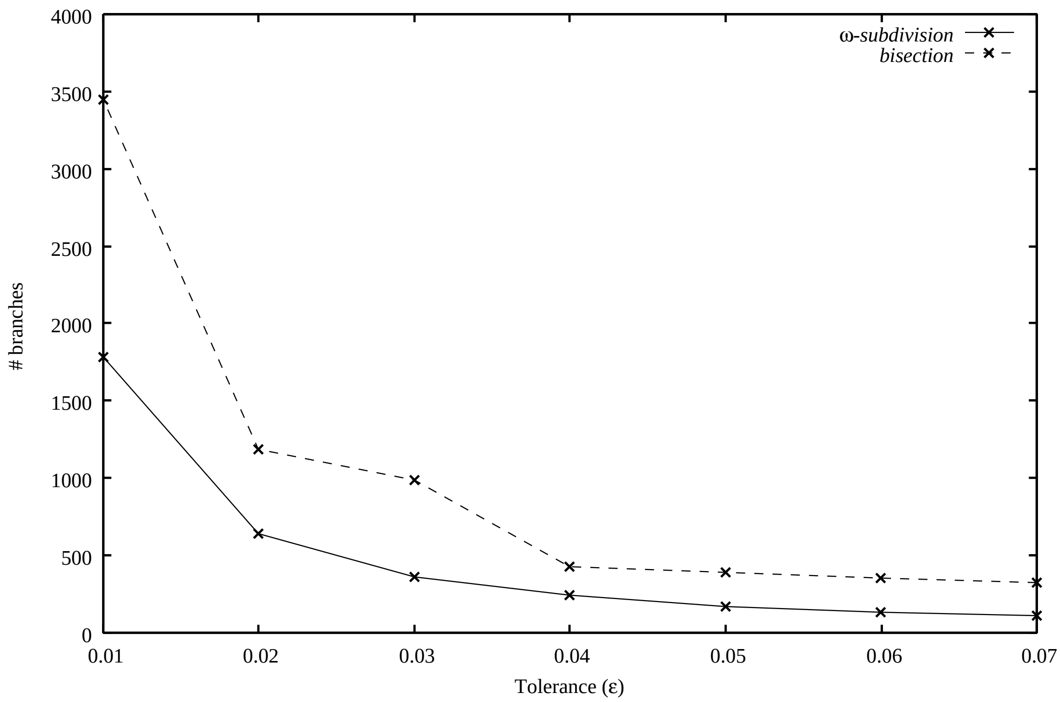


Figure 6: Average numbers of branching operations when $(n, r) = (3, 50)$.

Table 1: Average performance of $\omega$-*subdivision* when $n = 3$ and $\varepsilon = 0.05$.

| $r$ | 400 | 600 | 800 | 1,000 | 1,200 |
|---|---|---|---|---|---|
| CPU time | 80.61 | 133.3 | 197.3 | 262.4 | 354.9 |
| | (8.89) | (17.5) | (32.3) | (32.3) | (41.4) |
| # branches | 139.4 | 135.6 | 132.8 | 124.8 | 125.0 |
| | (15.8) | (18.4) | (20.0) | (15.7) | (14.9) |

increasing $r$, except for a few ups and downs. Accordingly, the computational time is dominated only by the time taken to solve $r$ linear programming problems with five variables, and increases moderately in an almost linear fashion with increase in $r$. In contrast to this, both programs are rather sensitive to changes in $\varepsilon$. Figures 5 and 6 show the variation of the average computational time and number of branching operations, respectively, required by each program when $(n, r)$ was fixed at $(3, 50)$ and $\varepsilon$ was changed in the interval $[0.01, 0.07]$. The number of branching operations increases in proportion to the reciprocal of $\varepsilon$, and so does the computational time. Still, we see that $\omega$-*subdivision* requires at most half the computational time and number of branching operations required by *bisection* for each $\varepsilon$.

The results of solving instances with larger $r$ by $\omega$-*subdivision* are summarized in Table 1. It contains the average computational time (in seconds), the average number of branching operations, and their standard deviations in parentheses when $n = 3$, $\varepsilon = 0.05$, and $r$ ranged from 400 to $1,200$ in 200 increments. Again, the number of branching operations shows no increase with increasing $r$, and remains less than 140 for all values of $r$. Eventually, it took around six minutes to solve (20) of size $(n, r) = (3, 1200)$, which corresponds to triangulation from 600 images. We can conclude that the algorithm sum_ratio has performance more than enough, at least for three-dimensional triangulation in computer vision. In that case, the computational time will be further improved if we use the linear-time algorithm for linear programming problems in a fixed dimension [21].

How does the algorithm behave for instances with $n > 3$? Unfortunately, the performance of $\omega$-*subdivision* rapidly deteriorates with increasing $n$, as is shown in Table 2, which contains the same statistics as in Table 1. This table seems to indicate that sum_ratio is a poor replacement for the algorithms in [16, 17], which are reported to solve sum-of-ratios optimization problems with up to a hundred variables. However, we should remember that the application areas for those algorithms are completely different from those for sum_ratio. Unlike sum_ratio, it would be impossible for them to solve any instance with $1,200$ ratios in a practical amount of time, even if the number of variables is only three. What is important is that one should choose the right algorithm for one's particular application. There is no doubt that the algorithm sum_ratio widens the application range of sum-of-ratios optimization.

Table 2: Average performance of $\omega$-*subdivision* when $r = 50$ and $\varepsilon = 0.05$.

| $n$ | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|
| CPU time | 2.443 | 10.19 | 34.38 | 180.8 | 949.8 |
| | (0.558) | (4.06) | (20.57) | (100.0) | (519.7) |
| # branches | 43.0 | 170.6 | 538.6 | $2,675$ | $13,106$ |
| | (10.2) | (68.3) | (323.9) | $(1,479)$ | $(7,142)$ |

## References


[1] Almogy, Y., and O. Levin, "Parametric analysis of a multi-stage stochastic shipping problem", *Proceedings of the Fifth IFORS Conference* (Venice, 1969), 359–370.

[2] Arkin, E.M., Y.-J. Chiang, M. Held, J.S.B. Mitchell, V. Sacristan, S.S. Skiena, and T.-C. Yang, "On minimum-area hulls", *Algorithmica* **21** (1998), 119–136.

[3] Bartoli, A., and J.-T. Lapreste, "Triangulation for points on lines", *Image and Vision Computing* **26** (2008), 315–324.

[4] Benson, H.P., "Global optimization algorithm for the nonlinear sum of ratios problem", *Journal of Optimization Theory and Applications* **112** (2002), 1–29.

[5] Benson, H.P., "Using concave envelopes to globally solve the nonlinear sum of ratios problem", *Journal of Global Optimization* **22** (2002), 343–364.

[6] Charnes, A., and W.W. Cooper, "Programming with linear fractional functionals", *Naval Research Logistics Quarterly* **9** (1962), 181–186.

[7] Chen, D.Z., O. Daescu, Y. Dai, N. Katoh, X. Wu, and J. Xu, "Efficient algorithms and implementations for optimizing the sum of linear fractional functions, with applications", *Journal of Combinatorial Optimization* **9** (2005), 69–90.

[8] Falk, J.E., and R.M. Soland, "An algorithm for separable nonconvex programming problems", *Management Science* **15** (1969), 550–569.

[9] Freund, R.W., and F. Jarre, "Solving the sum-of-ratios problem by an interior-point method", *Journal of Global Optimization* **19** (2001), 83–102.

[10] GNU Octave, http://www.gnu.org/software/octave/.

[11] Hartley, R., and F. Kahl, "Optimal algorithms in multiview geometry", *Proceedings of the Asian conference on computer vision* **1** (2007), 13–34.

[12] Horst, R., and H. Tuy, *Global Optimization: Deterministic Approaches*, 2nd ed., Springer (Berlin, 1993).

[13] Kahl, F., S. Agarwal, M.K. Chandraker, D. Kriegman, and S. Belongie, "Practical global optimization for multiview geometry", *International Journal of Computer Vision* **79** (2008), 271–284.

[14] Konno, H., and M. Inori, "Bond portfolio optimization by bilinear fractional programming", *Journal of the Operations Research of Japan* **32** (1989), 143–158.

[15] Konno, H., and H. Watanabe, "Bond portfolio optimization problems and their applications to index tracking: a partial optimization approach", *Journal of the Operations Research Society of Japan* **39** (1996), 295–306.

[16] Kuno, T., "A branch-and-bound algorithm for maximizing the sum of several linear ratios", *Journal of Global Optimization* **22** (2002), 155–174.

[17] Kuno, T., "A revision of the trapezoidal branch-and-bound algorithm for linear sum-of-ratios problems", *Journal of Global Optimization* **33** (2005), 435–464.

[18] Majhi, J., R. Janardan, J. Schwerdt, M. Smid, and P. Gupta, "Minimizing support structures and trapped area in two-dimensional layered manufacturing", *Computational Geometry* **12** (1999), 241–267.

[19] Mangasarian, O.L., *Nonlinear Programming*, Krieger (Florida, 1969).

[20] Matsui, T., "NP-hardness of linear multiplicative programming and related problems", *Journal of Global Optimization* **9** (1996), 113–119.

[21] Megiddo, N., "Linear programming in linear time when the dimension is fixed", *Journal of the Association of Computing Machinery* **31** (1984), 114–127.

[22] Tuy, H., *Convex Analysis and Global Optimization*, Kluwer Academic Publishers (Dordrecht, 1998).