数理解析研究所講究録
第 1726 巻 2011 年 112-122

Global Optimization in Computer Vision

Takahito Kuno*

Graduate School of Systems end Information Engineering
University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan

Abstract

In this paper, we develop an algorithm for minimizing the L_p norm of a vector whose components are linear fractional functions, where p is an arbitrary positive integer. The problem is included in the sum-of-ratios 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 exploits those special structures and generates a globally optimal solution in a practical amount of computational time.

Key words: Global optimization, fractional programming, sum-of-ratios, branch-and-bound, computer vision, multiview geometry.

1 Introduction

Fractional optimization problems have been studied in order to achieve optimal economic performance, as evidenced by the fact that a time rate of earnings or profit is usually represented by a fractional function. 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, 10, 11]). Recently, however, fractional optimization has attracted much attention in multiview geometry of computer vision, without any direct relation to economic performance. Since multiview geometry is developed in projective spaces, fractional functions play an essential rule there. A variety of problems, e.g., triangulation, camera resectioning, homography estimation, and so forth (see e.g., [7, 9]), are 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 also characterized by a small number of variables but a large number of ratios in the objective function. Unfortunately, however, existing algorithms are not adequate to solve such a kind of problems because those were designed for problems with only a few ratios [2, 3, 12, 13]. The purpose of this paper is to propose a deterministic algorithm for solving sum-of-ratios optimization problems with these features in a practical amount of computational time.

*The author was partially supported by a Grant-in-Aid for Scientific Research (B 20310082) and 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

In the next section, we gives a formal definition of the target problem, and illustrate how it arises in computer vision, in Section 3. In Section 4, we develop a convergent branch-and-bound algorithm to generate a globally optimal solution, and report numerical results in Section 5.

2 Sum-of-ratios optimization

The problem considered in this paper is a class of fractional optimization problems:

$$\left| \begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{q} \left| \frac{\mathbf{d}^i\mathbf{x}+\delta^i}{\mathbf{c}^i\mathbf{x}+\gamma^i} \right|^p \\ \text{subject to} & \mathbf{A}\mathbf{x} \geq \mathbf{b}, \quad \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}, \end{array} \right. \tag{1}$$

where $\mathbf{A} \in \mathbf{R}^{m\times n}$, $\mathbf{b} \in \mathbf{R}^m$, $\mathbf{c}^i, \mathbf{d}^i, \mathbf{u} \in \mathbf{R}^n$, $\gamma^i, \delta^i \in \mathbf{R}^1$, and p is a positive integer. Let

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

and assume through the paper that

$$\mathbf{c}^i\mathbf{x} + \gamma^i > 0, \quad i = 1, \ldots, q, \quad \forall \mathbf{x} \in D. \tag{2}$$

If $q = 1$, then (1) is a linear sum-of-ratios problem, for which branch-and-bound algorithms have been proposed in [12, 13]. When $q = 2$, problem (1) is a special case of nonlinear sum-of-ratios problem:

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

where $d_i : \mathbf{R}^n \to \mathbf{R}^1$ is a convex function, $c_i : \mathbf{R}^n \to \mathbf{R}^1$ is a concave and positive valued function. As shown in [9], these conditions are satisfied if we set

$$c_i(\mathbf{x}) = \mathbf{c}^i\mathbf{x} + \gamma^i, \quad d_i(\mathbf{x}) = \frac{(\mathbf{d}^i\mathbf{x} + \delta^i)^2}{\mathbf{c}^i\mathbf{x} + \gamma^i}, \quad i = 1, \ldots, q.$$

For (3), branch-and-bound algorithms similar to the one in [12] have also been developed in [2, 3]. The number q of ratios that can be handled by those existing algorithms is limited to only around ten. The difficulty of (1) is attributed to the *sum* of ratios, not due to ratios themselves. To see this, consider the simplest case where $p = 1$. It is known that a linear ratio is a quasiconvex and quasiconcave function on the domain where the denominator is positive (see e.g., [14]). However, the sum of quasiconvex functions is not in general quasiconvex, and the sum of quasiconcave functions is not quasiconcave. This implies that the sum of linear ratios is neither quasiconvex nor quasiconcave. As a result, (1) has multiple local minima which are not globally optimal, even when $p = 1$. From the viewpoint of computational complexity, (1) is also known to be $\mathscr{NP}$-hard [5, 15].

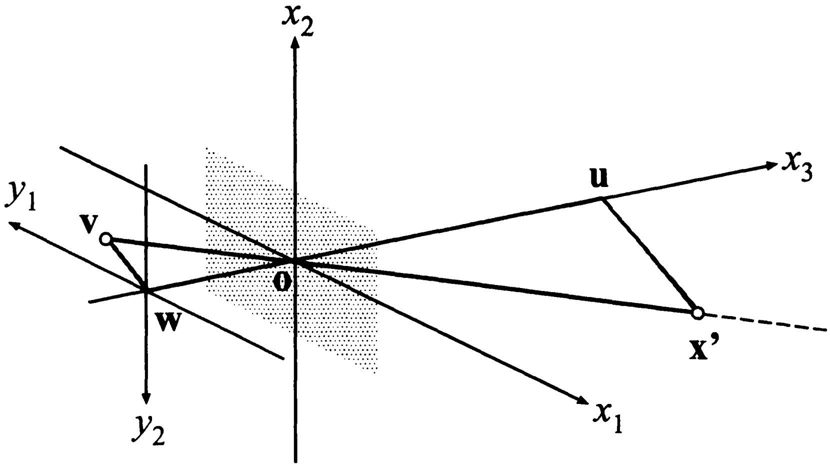

Figure 1: Geometry of a pinhole camera.

3 Example in computer vision

The sum-of-ratios problem (1), although difficult to solve, has a wide variety of applications within computer vision dealing with geometric relations between the 3D world and its projection onto a 2D image plane. In this section, we take triangulation as a typical example of application and show how it can be formulated into (1). The basis for this formulation is the pinhole camera model.

Pinhole camera model

The pinhole camera model describes the relationship between the coordinates of a 3D 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 subject of this camera by $\mathbf{x}' = (x'_1, x'_2, x'_3)^\top$ in the 3D coordinate system with its origin at the camera aperture $\mathbf{0}$. Light emanating from $\mathbf{x}'$ passes through $\mathbf{0}$ and projects 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 f from $\mathbf{0}$ in the negative direction of the x_3 axis. Let $\mathbf{u} = (0,0,x'_3)^\top$, $\mathbf{v} = (y'_1, y'_2, -f)^\top$ and $\mathbf{w} = (0,0,-f)^\top$. Since the triangle connecting three points $\mathbf{0}, \mathbf{x}'$ and $\mathbf{u}$ is similar to that connecting $\mathbf{0}, \mathbf{v}$ and $\mathbf{w}$, we have $(y'_1, y'_2)^\top = (f/x'_3)(x'_1, x'_2)^\top$, or equivalently

$$\begin{bmatrix} y'_1 \\ y'_2 \\ 1 \end{bmatrix} = \frac{f}{x'_3} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3/f \end{bmatrix}$$

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

$$\begin{bmatrix} y'_1 \\ y'_2 \\ 1 \end{bmatrix} \sim \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3/f \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \begin{bmatrix} x'_1 \\ x'_2 \\ x'_3 \\ 1 \end{bmatrix}, \tag{4}$$

and say that $(y'_1, y'_2, 1)^\top$ is *equivalent, or proportional, to* $(x'_1, x'_2, x'_3/f)^\top$. The 3×4 matrix in (4) is called the *camera matrix.*

TRIANGULATION

Triangulation is the process of determining the 3D coordinates of $\mathbf{x}'$ given its projections onto two, or more, image planes. In theory, the triangulation problem is trivial. Each image $\mathbf{y}'$ of $\mathbf{x}'$ corresponds to a line in the 3D space such that all points on the line are projected to $\mathbf{y}'$. Therefore, $\mathbf{x}'$ must lie on the intersection of those lines, and we must be able to compute it 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, the lines associated with the images of $\mathbf{x}'$ do not always intersect in the 3D space.

Suppose that $\mathbf{x}' = (x'_1, x'_2, x'_3)^\top$ is in an arbitrary 3D coordinate system, and that there are N images $\mathbf{y}^k = (y^k_1, y^k_2)^\top$ of $\mathbf{x}'$ captured by cameras $k = 1, \ldots, N$. Let us denote the kth camera matrix by

$$\mathbf{C}^k_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f^k & 0 \end{bmatrix},$$

where f^k is the focal length of camera k. Note that $\mathbf{x}'$ is denoted as $\mathbf{R}^k\mathbf{x}' + \mathbf{t}^k$ for some rotation matrix $\mathbf{R}^k$ and translation vector $\mathbf{t}^k$ in the 3D coordinate system with the origin at the focal point of camera k. Hence, from (4), we have

$$\begin{bmatrix} \mathbf{y}^k \\ 1 \end{bmatrix} \sim \mathbf{C}^k_0 \begin{bmatrix} \mathbf{R}^k & \mathbf{t}^k \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}' \\ 1 \end{bmatrix}, \quad k = 1, \ldots, N.$$

Let

$$\mathbf{C}^k = \begin{bmatrix} \mathbf{c}^k_1 & \gamma^k_1 \\ \mathbf{c}^k_2 & \gamma^k_2 \\ \mathbf{c}^k_3 & \gamma^k_3 \end{bmatrix} = \mathbf{C}^k_0 \begin{bmatrix} \mathbf{R}^k & \mathbf{t}^k \\ \mathbf{0} & 1 \end{bmatrix},$$

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

$$y^k_1 = \frac{\mathbf{c}^k_1\mathbf{x}' + \gamma^k_1}{\mathbf{c}^k_3\mathbf{x}' + \gamma^k_3}, \quad y^k_2 = \frac{\mathbf{c}^k_2\mathbf{x}' + \gamma^k_2}{\mathbf{c}^k_3\mathbf{x}' + \gamma^k_3},$$

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 $\mathbf{y}^k$ and the measurement $\bar{\mathbf{y}}^k$:

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

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

$$\left| \begin{array}{ll} \text{minimize} & \sum_{k=1}^{N} \sum_{j=1}^{2} \left(\mathbf{r}_j^k(\mathbf{x})\right)^p \\ \text{subject to} & \mathbf{c}_3^k\mathbf{x} + \gamma_3^k \geq 0, \quad k = 1,\ldots,N, \end{array} \right. \tag{5}$$

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

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

problem (5) is a special case of our target (1). Besides this 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. [7, 9].

4 Practical but rigorous algorithm

The problem (1) arising in computer vision is characterized generally by a large number of ratios and a small number of variables. Exploiting this special structure, we develop a practical branch-and-bound algorithm for generating a globally optimal solution. First, we will derive a relaxation of (1), which is solved iteratively for the bounding operation.

LP RELAXATION

Let us apply the Charnes-Cooper transformation [4] to (1) by introducing auxiliary variables:

$$\mathbf{y}_i = \eta_i\mathbf{x}, \quad \eta_i = \frac{1}{\mathbf{c}^i\mathbf{x} + \gamma^i}, \quad i = 1,\ldots,q.$$

Then we have

$$\left| \begin{array}{ll} \text{minimize} & \sum_{i=1}^{q} \left|\mathbf{d}^i\mathbf{y}_i + \delta^i\eta_i\right|^p \\ \text{subject to} & \left. \begin{array}{l} \mathbf{A}\mathbf{y}_i - \mathbf{b}\eta_i \geq \mathbf{0} \\ \mathbf{c}^i\mathbf{y}_i + \gamma^i\eta_i = 1 \\ \mathbf{y}_i = \eta_i\mathbf{x} \\ \mathbf{y}_i \geq \mathbf{0}, \quad \eta_i \geq 0 \end{array} \right\} \quad i = 1,\ldots,q \\ & \mathbf{0} \leq \mathbf{x} \leq \mathbf{u}, \end{array} \right. \tag{6}$$

which is equivalent to (1) in the following sense. If $\mathbf{x}^*$ is an optimal solution of (1), then $(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\eta}^*)$ with $\mathbf{y}_i^* = \eta_i^* \mathbf{x}^*$ and $\eta_i^* = 1/(\mathbf{c}^i \mathbf{x}^* + \gamma^i)$ 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 (1).

Consider a subproblem of (6):

$$
\mathrm{P}(\mathbf{s},\mathbf{t}) \left|
\begin{array}{ll}
\text{minimize} & \displaystyle\sum_{i=1}^{q} |\mathbf{d}^i \mathbf{y}_i + \delta^i \eta_i|^p \\
\text{subject to} & \left.\begin{array}{l} \mathbf{A}\mathbf{y}_i - \mathbf{b}\eta_i \geq \mathbf{0} \\ \mathbf{c}^i \mathbf{y}_i + \gamma^i \eta_i = 1 \\ \mathbf{y}_i = \eta_i \mathbf{x} \\ \mathbf{y}_i \geq \mathbf{0}, \quad \eta_i \geq 0 \end{array}\right\} \quad i = 1, \ldots, q \\
& \mathbf{s} \leq \mathbf{x} \leq \mathbf{t},
\end{array}\right.
$$

where $\mathbf{0} \leq \mathbf{s} \leq \mathbf{t} \leq \mathbf{u}$. The feasible set of this problem is not convex, but can be relaxed into a convex one:

$$
\overline{\mathrm{P}}(\mathbf{s},\mathbf{t}) \left|
\begin{array}{ll}
\text{minimize} & \displaystyle\sum_{i=1}^{q} |\mathbf{d}^i \mathbf{y}_i + \delta^i \eta_i|^p \\
\text{subject to} & \left.\begin{array}{l} \mathbf{A}\mathbf{y}_i - \mathbf{b}\eta_i \geq \mathbf{0} \\ \mathbf{c}^i \mathbf{y}_i + \gamma^i \eta_i = 1 \\ \mathbf{s}\eta_i \leq \mathbf{y}_i \leq \mathbf{t}\eta_i \\ \mathbf{y}_i \geq \mathbf{0}, \quad \eta_i \geq 0 \end{array}\right\} \quad i = 1, \ldots, q.
\end{array}\right.
$$

Proposition 4.1. *If $P(\mathbf{s},\mathbf{t})$ has an optimal solution $(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\eta}^*)$, then $\overline{P}(\mathbf{s},\mathbf{t})$ also has an optimal solution $(\overline{\mathbf{y}}, \overline{\boldsymbol{\eta}})$, which satisfies*

$$\sum_{i=1}^{q} |\mathbf{d}^i \overline{\mathbf{y}}_i + \delta^i \overline{\eta}_i|^p \leq |\mathbf{d}^i \mathbf{y}_i^* + \delta^i \eta_i^*|^p .$$

In addition, if $(\overline{\mathbf{y}}, \overline{\boldsymbol{\eta}})$ satisfies

$$\overline{y}_{1j}/\overline{\eta}_1 = \cdots = \overline{y}_{qj}/\overline{\eta}_q, \quad j = 1, \ldots, n,$$

then $(\overline{\mathbf{x}}, \overline{\mathbf{y}}, \overline{\boldsymbol{\eta}})$ with $\overline{x}_j = \overline{y}_{1j}/\overline{\eta}_1$ is an optimal solution of $P(\mathbf{s},\mathbf{t})$.

One more thing to be noted on $\overline{\mathrm{P}}(\mathbf{s},\mathbf{t})$ is that it is decomposable into q problems, each of which is of the form:

$$
\left|
\begin{array}{ll}
\text{minimize} & |\mathbf{d}^i \mathbf{y}_i + \delta^i \eta_i|^p \\
\text{subject to} & \mathbf{A}\mathbf{y}_i - \mathbf{b}\eta_i \geq \mathbf{0} \\
& \mathbf{c}^i \mathbf{y}_i + \gamma^i \eta_i = 1 \\
& \mathbf{s}\eta_i \leq \mathbf{y}_i \leq \mathbf{t}\eta_i \\
& \mathbf{y}_i \geq \mathbf{0}, \quad \eta_i \geq 0.
\end{array}\right. \tag{7}
$$

Introducing another auxiliary variable $\zeta_i = \left|\mathbf{d}^i\mathbf{y}_i + \delta^i\eta_i\right|$, we can rewrite (7) into

$$\left|\begin{array}{ll} \text{minimize} & \zeta_i^p \\ \text{subject to} & -\zeta_i \leq \mathbf{d}^i\mathbf{y}_i + \delta^i\eta_i \leq \zeta_i \\ & \mathbf{A}\mathbf{y}_i - \mathbf{b}\eta_i \geq \mathbf{0} \\ & \mathbf{c}^i\mathbf{y}_i + \gamma^i\eta_i = 1 \\ & \mathbf{s}\eta_i \leq \mathbf{y}_i \leq \mathbf{t}\eta_i \\ & \mathbf{y}_i \geq \mathbf{0}, \quad \eta_i \geq 0, \quad \zeta_i \geq 0. \end{array}\right.$$

To minimize ζ_i^p, we only need to minimize ζ_i, despite the magnitude of p, because ζ_i is a nonnegative variable. We can therefore solve (7) by solving a linear programming problem:

$$\left|\begin{array}{ll} \text{minimize} & \zeta_i \\ \text{subject to} & -\zeta_i \leq \mathbf{d}^i\mathbf{y}_i + \delta^i\eta_i \leq \zeta_i \\ & \mathbf{A}\mathbf{y}_i - \mathbf{b}\eta_i \geq \mathbf{0} \\ & \mathbf{c}^i\mathbf{y}_i + \gamma^i\eta_i = 1 \\ & \mathbf{s}\eta_i \leq \mathbf{y}_i \leq \mathbf{t}\eta_i \\ & \mathbf{y}_i \geq \mathbf{0}, \quad \eta_i \geq 0, \quad \zeta_i \geq 0. \end{array}\right. \tag{8}$$

Proposition 4.2. *The relaxation $\overline{P}(\mathbf{s},\mathbf{t})$ has an optimal solution $(\overline{\mathbf{y}},\overline{\boldsymbol{\eta}})$ if and only if (8) has an optimal solution $(\overline{\mathbf{y}}_i,\overline{\eta}_i,\overline{\zeta}_i)$ for each $i = 1,\ldots,q$.*

Thus, we can see whether $P(\mathbf{s},\mathbf{t})$ is worth solving or not, by solving q linear programming problems with $(n+2)$ variables. If $\sum_{i=1}^q \overline{\zeta}_i^p \geq z^*$ holds for the value z^* of the best feasible solution obtained so far, we can discard $P(\mathbf{s},\mathbf{t})$ from further consideration. Otherwise, we have to examine subproblems of $P(\mathbf{s},\mathbf{t})$ in turn.

SUBDIVISION RULE

One way to generate subproblems of $P(\mathbf{s},\mathbf{t})$ is the subdivision of $[\mathbf{s},\mathbf{t}] = \{\mathbf{x} \in \mathbf{R}^n \mid \mathbf{s} \leq \mathbf{x} \leq \mathbf{t}\}$ into a set of subrectangles. If we divide $[\mathbf{s},\mathbf{t}]$ along $x_k = (s_k + t_k)/2$ for $k \in \arg\max\{t_j - s_j \mid j = 1,\ldots,n\}$, the algorithm is guaranteed to be convergent. Instead of such an exhaustive method, we will propose here a more sophisticated subdivision rule.

An optimal solution $(\overline{\mathbf{y}},\overline{\boldsymbol{\eta}})$ of $\overline{P}(\mathbf{s},\mathbf{t})$ naturally satisfies

$$\mathbf{s}\overline{\eta}_i \leq \overline{\mathbf{y}}_i \leq \mathbf{t}\overline{\eta}_i, \quad i = 1,\ldots,q.$$

Let

$$\boldsymbol{\omega} = \frac{1}{q}\sum_{i=1}^q \frac{1}{\overline{\eta}_i}\overline{\mathbf{y}}_i. \tag{9}$$

Then $\boldsymbol{\omega}$ is a point in the rectangle $[\mathbf{s},\mathbf{t}]$ We can use this point $\boldsymbol{\omega}$ like the subdivision point of the usual ω-subdivision rule for the rectangular branch-and-bound algorithm. Let

$$\delta_j = \min\{t_j - \omega_j, \omega_j - s_j\}, \quad j = 1,\ldots,n, \tag{10}$$

and let

$$j' \in \arg\max\{\delta_j \mid j = 1, \ldots, n\}. \tag{11}$$

We may divide $[\mathbf{s}, \mathbf{t}]$ into $[\mathbf{s}', \mathbf{t}]$ and $[\mathbf{s}, \mathbf{t}']$, where

$$s'_j = \begin{cases} \omega_j, & \text{if } j = j' \\ s_j, & \text{otherwise,} \end{cases} \qquad t'_j = \begin{cases} \omega_j, & \text{if } j = j' \\ t_j, & \text{otherwise.} \end{cases}$$

Then a sequence $\{k\}$ will be generated and satisfy

$$\mathbf{s}^k \le \mathbf{s}^{k+1} \le \boldsymbol{\omega}^{k+1} \le \mathbf{t}^{k+1} \le \mathbf{t}^k, \quad k = 1, 2, \ldots, \tag{12}$$

where $s_{j^k}^{k+1} = \omega_{j^k}^k$ and $t_{j^k}^{k+1} = \omega_{j^k}^k$.

Lemma 4.3. *There exist some points $\mathbf{s}^0 \le \mathbf{t}^0$ such that*

$$\mathbf{s}^k \to \mathbf{s}^0, \quad \mathbf{t}^k \to \mathbf{t}^0, \quad \textit{as } k \to +\infty,$$

and $\{\boldsymbol{\omega}^k\}$ accumulates at a corner of the rectangle $[\mathbf{s}^0, \mathbf{t}^0]$.

Proof. We see from (12) that for each j the sequences $\{s_j^k\}$ and $\{t_j^k\}$ are monotonic, bounded, and hence have limits s_j^0 and t_j^0, respectively. Since $\{1, \ldots, n\}$ is a finite set, there is a subsequence $\{k_\ell\}$ such that $j^{k_\ell} = r \in \{1, \ldots, n\}$ for every ℓ. We can also assume that $\omega_r^{k_\ell} \in \{s_r^{k_\ell+1}, t_r^{k_\ell+1}\}$. Hence, we have $\omega_r^{k_\ell} \to \omega_r^0 \in \{s_r^0, t_r^0\}$, as $\ell \to +\infty$, by taking a further subsequence if necessary. For each ℓ in this subsequence, we can assume from (10) and (11) that

$$\min\{t_r^{k_\ell} - \omega_r^{k_\ell}, \omega_r^{k_\ell} - s_r^{k_\ell}\} \ge \min\{t_j^{k_\ell} - \omega_j^{k_\ell}, \omega_j^{k_\ell} - s_j^{k_\ell}\}, \quad j = 1, \ldots, n.$$

The left-hand side converges to zero, and so does the right-hand side. This implies that a corner of $[\mathbf{s}^0, \mathbf{t}^0]$ is an accumulation point of $\{\boldsymbol{\omega}^k\}$. □

Lemma 4.4. *Let $\{k_\ell\}$ be a subsequence such that $\{\boldsymbol{\omega}^{k_\ell}\}$ converges to a corner of $[\mathbf{s}^0, \mathbf{t}^0]$. Then, as $\ell \to +\infty$,*

$$\rho\left((1/\overline{\eta}_i^{k_\ell})\overline{\mathbf{y}}_i^{k_\ell}, \boldsymbol{\omega}^{k_\ell}\right) \to 0, \quad i = 1, \ldots, q,$$

where $\rho(\cdot, \cdot)$ represents the distance between the two points.

Proof. Assume on the contrary that there exist indices r, j and a positive number ε such that, for infinitely many ℓ,

$$\left|\overline{y}_{rj}^{k_\ell}/\overline{\eta}_r^{k_\ell} - \omega_j^{k_\ell}\right| > \varepsilon. \tag{13}$$

If $\omega_j^{k_\ell} \to s_j^0$, then we have

$$\omega_j^{k_\ell} - s_j^{k_\ell} = \frac{1}{q}\sum_{i=1}^{q}\left(\overline{y}_{ij}^{k_\ell}/\overline{\eta}_i^{k_\ell} - s_j^{k_\ell}\right) \ge \frac{1}{q}\left(\overline{y}_{rj}^{k_\ell}/\overline{\eta}_r^{k_\ell} - s_j^{k_\ell}\right) \ge 0,$$

because $s_j^{k_\ell}\overline{\eta}_i^{k_\ell} \le \overline{y}_{ij}^{k_\ell}$ holds for each i. Hence, $\omega_j^{k_\ell} \to s_j^0$ and $\overline{y}_{rj}^{k_\ell}/\overline{\eta}_r^{k_\ell} \to s_j^0$, which contradicts (13). Even if $\{\omega_j^{k_\ell}\} \to t_j^0$, we reach a similar contradiction. □

OUTLINE OF THE ALGORITHM

Starting from $[\mathbf{s}^1, \mathbf{t}^1] = [\mathbf{0}, \mathbf{u}]$, we solve the relaxed problem $\overline{\mathrm{P}}(\mathbf{s}^k, \mathbf{t}^k)$ recursively for $k = 1, 2, \ldots$. The rectangle $[\mathbf{s}^k, \mathbf{t}^k]$ is discarded unless the value of $\overline{\mathrm{P}}(\mathbf{s}^k, \mathbf{t}^k)$ is less than the value of the incumbent $\mathbf{x}^k$, the best feasible solution of (1) obtained in the course of the algorithm. Note that $\boldsymbol{\omega}^k \in [\mathbf{s}^k, \mathbf{t}^k]$ determined by (9) from $(\overline{\mathbf{y}}^k, \overline{\boldsymbol{\eta}}^k)$ is feasible for (1). This feasible solution $\boldsymbol{\omega}^k$ is also used to divide $[\mathbf{s}^k, \mathbf{t}^k]$ into two subrectangles, according to (10) and (11). If we select $[\mathbf{s}^k, \mathbf{t}^k]$ with the least value of $\overline{\mathrm{P}}(\mathbf{s}^k, \mathbf{t}^k)$ and divide it at each iteration, every accumulation point of the sequence $\{\mathbf{x}^k\}$ is an optimal solution of (1). This can be proven using Lemmas 4.3 and 4.4, but the proof is omitted here and will be presented elsewhere, together with the more detailed description of the algorithm.

5 Numerical results

We coded the algorithm sketched in the previous section using GNU Octave (version 3.0.5) [6], a MATLAB-like computational tool, and tested it on an AMD Opteron 256 (3.0GHz) single core processor. The problem used as a benchmark is of the form:

$$\left|\begin{array}{lll} \text{minimize} & \displaystyle\sum_{i=1}^{q} \left| \frac{\mathbf{d}^i\mathbf{x} + \delta^i}{\mathbf{c}^i\mathbf{x} + \gamma^i} \right|^2 & \\ \text{subject to} & \mathbf{c}^i\mathbf{x} + \gamma^i \geq 0, & i = 1, \ldots, q \\ & 0 \leq x_j \leq 10.0, & j = 1, 2, 3, \end{array}\right. \tag{14}$$

which simulates a triangulation problem with $q/2$ images. We generated all c^i_j, γ^i, d^i_j and δ^i randomly in the interval $[-0.5, 0.5]$, and solved ten instances for each q, ranged from 50 to $1,000$. As for the subdivision rule, we tested the usual bisection rule as well as our proposed one, referred to as *ω-subdivision.*

Figure 2 depicts the variation in the average CPU seconds required by the algorithm when p was changed from 50 to 300 in 50 increments. Figure 3 shows the variation in the average number of iterations for the same set of instances. The solid lines represent the results by ω-subdivision, and the dashed lines are those by bisection. It is quite obvious that ω-subdivision is much superior to bisection. The CPU seconds under both rules increase rather moderately as a function in q. It should be noted, however, that the numbers of iterations show no sign of increase. The results for instances with larger q by ω-subdivision is listed in Table 1. We can conclude from these results that the algorithm proposed for solving (1) in this paper has performance more than enough, at least for triangulation problems in computer vision.

q	400	600	800	1,000
CPU seconds	83.17	134.2	212.3	299.3
# iterations	138.8	132.2	137.8	137.4

Table 1: Performance of the algorithm under the proposed subdivision rule

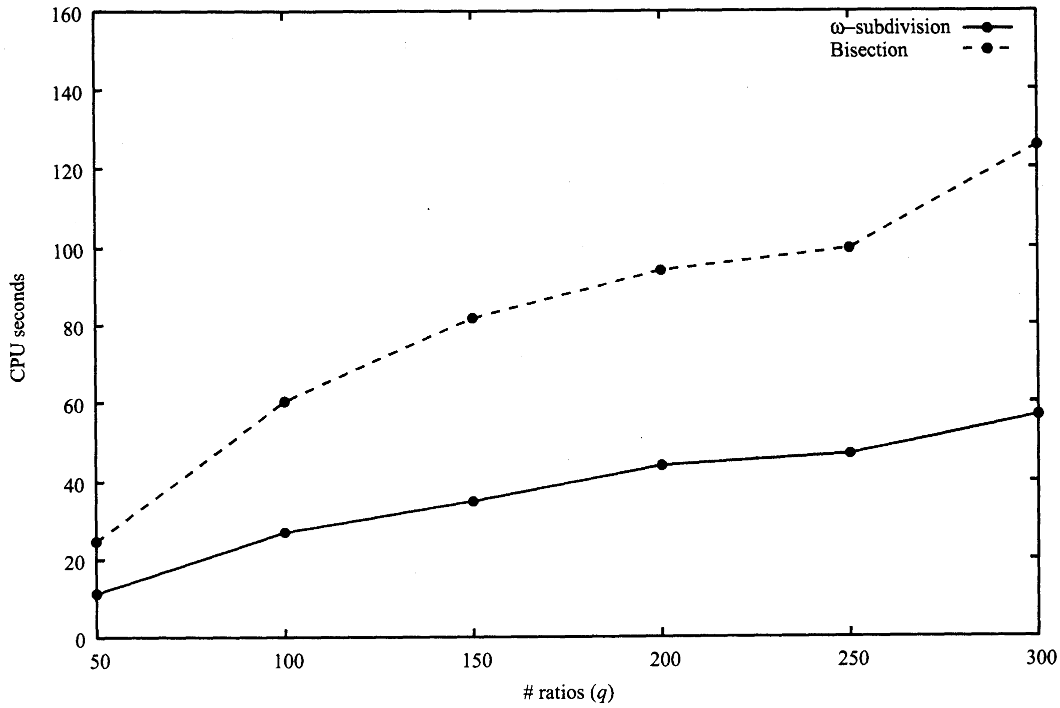

Figure 2: Average CPU seconds.

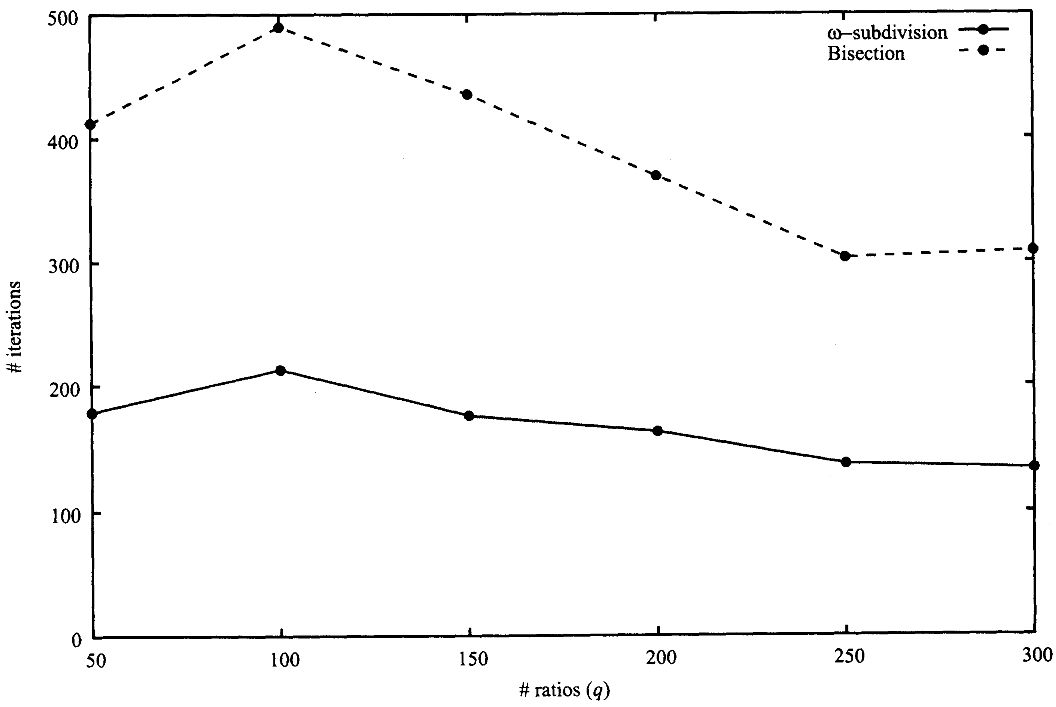

Figure 3: Average numbers of iterations.

References

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

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

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

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

[5] 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.

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

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

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

[9] 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.

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

[11] 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.

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

[13] 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.

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

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

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