文章编号: 1000-6788(2013)03-0742-06 中图分类号: O211.2 文献标志码: A

基于单纯形剖分确定非线性比式和问题全局解的新方法

汪春峰 [1], 刘三阳 [2]

(1. 河南师范大学 数学与信息科学学院, 新乡 453007; 2. 西安电子科技大学 理学院, 西安 710071)

摘 要 针对经济与金融中出现的一类特殊非线性比式和问题, 给出一种基于单纯形剖分的全局优化算法. 在算法中, 通过构造初始单纯形, 以及使用凸包络理论, 提出了一个确定原问题最优值下界的新方法. 在确定下界的同时, 将会得到原问题的 $n+1$ 可行解, 这些可行解可以用于上界的改善. 理论上证明了算法的收敛性, 数值算例表明算法是有效可行的.

关键词 全局优化; 分支定界; 单纯形对分; 比式和

New method for solving nonlinear sum of ratios problem based on simplicial bisection

WANG Chun-feng[1], LIU San-yang[2]

(1. College of Mathematics and Information Science, Henan Normal University, Xinxiang 453007, China; 2. School of Science, Xidian University, Xi'an 710071, China)

Abstract For solving a special class of nonlinear sum of ratios problem arised in economy and finance, a global optimization algorithm is presented based on simplicial bisection. In this algorithm, by constructing an initial simplex and using convex envelope theory, a new method to determine lower bound of the optimal value for the original problem is proposed. With the determination of the lower bound, $n+1$ feasible points of the original problem will be found, which can be used to improve upper bound. Convergence of the algorithm is shown and some numerical examples are given to illustrate the feasibility and effectiveness of the presented algorithm.

Keywords global optimization; branch and bound; simplicial bisection; sum of ratios

1 引言

本文考虑如下形式的非线性比式和问题

$$
(\mathrm{P})\begin{cases} \min & f(x)=\sum\limits_{i=1}^{p}\dfrac{c_i^{\mathrm{T}}x+\alpha_i}{n_i(x)} \\ \text{s.t.} & Ax\le b \\ & x\ge 0 \end{cases}
$$

其中 $p\ge 2,\ c_i\in R^n,\ \alpha_i\in R,\ i=1,2,\cdots,p,\ A\in R^{m\times n},\ b\in R^m$. 假设可行域 $D=\{x\mid Ax\le b,\ x\ge 0\}$ 是非空有界的, $n_i(x)$ 为凹函数, 且对所有 $x\in D$, 有 $c_i^{\mathrm{T}}x+\alpha_i\ge 0$, $n_i(x)>0$.

比式和问题在经济与金融、交通等诸多领域有广泛应用. 自 1990 年以来, 此类问题受到了众多学者的关注. 但是由于比式和问题为非凸规划问题, 存在多个非全局的局部最优解, 所以求解起来比较困难. 在问题 (P) 中, 若 $c_i^{\mathrm{T}}x+\alpha_i\ge 0$, 且 $n_i(x)$ 为大于 0 的仿射函数时, 文献 [1–4] 给出了几个有效的方法. 当 $n_i(x)$ 为不等于 0 的仿射函数时, 文献 [5–8] 提出了几个分支定界算法. 本文考虑分母为大于 0 的凹函数, 分子有非负限制的情况. 显然此类问题较之以上所考虑问题有所推广, 当然求解起来也更困难. 针对此类问题, 本文提出一个新的基于单纯形对分的全局优化方法, 并从理论上分析了算法的收敛性. 最后通过数值算例验证了算法的可行性及有效性.

收稿日期: 2010-10-08
资助项目: 国家自然科学基金 (11171094); 河南师范大学博士科研启动课题 (qd12103)
作者简介: 汪春峰, 男, 汉, 讲师, 研究方向: 最优化理论方法及应用.

2 预备知识

首先通过一个简单变形, 问题 (P) 可以改写为如下等价问题:

$$\begin{aligned}
\min\ & \sum_{i=1}^{p}\frac{y_i}{-n_i(x)}\\
\text{s.t.}\ & y_i+c_i^{\mathrm{T}}x=-\alpha_i,\ i=1,2,\cdots,p\\
& Ax\le b\\
& x\ge 0
\end{aligned}$$

在后面, 我们会看到, 这个问题在寻找问题 (P) 最优值的上、下界的过程中起着重要作用.

为求解问题 (P), 本文给出一个新的单纯形分支定界算法. 在算法中, 单纯形对分及上、下界估计是两个基本环节.

2.1 初始单纯形及单纯形对分

本文采用文献 [9] 中的方法构造包含 D 的初始单纯形 S^0. 具体过程如下: 首先计算

$$\gamma=\max\left\{\sum_{r=1}^{n}x_r\middle|x\in D\right\},\ \gamma_r=\min\{x_r\mid x\in D\},\ r=1,2,\cdots,n.$$

定义 S^0 为如下集合

$$S^0=\left\{x\in R^n\middle|x_r\ge\gamma_r,\ r=1,2,\cdots,n,\sum_{r=1}^{n}x_r\le\gamma\right\},$$

则 S^0 为包含的 D 单纯形, 且其顶点为 $\{V^1,V^2,\cdots,V^{n+1}\}$, 其中 $V^1=(\gamma_1,\gamma_2,\cdots,\gamma_n)$,

$$V^{j+1}=(\gamma_1,\gamma_2,\cdots,\gamma_{j-1},\tau_j,\gamma_{j+1},\cdots,\gamma_n),\ \tau_j=\gamma-\sum_{r\ne j}\gamma_r,\ j=1,2,\cdots,n.$$

单纯形对分过程如下: 令 $S=[V^1,V^2,\cdots,V^{n+1}]$ 表示将被剖分的 S^0 的子单纯形, c 为 S 任意最长边 $[V^s,V^{\tilde{s}}]$ 的中点, i.e.

$$\|V^s-V^{\tilde{s}}\|=\max_{\tilde{j},j=1,2,\cdots,n+1}\{\|V^{\tilde{j}}-V^j\|\},$$

其中 $\|\cdot\|$ 为 R^n 中任一范数. 则 $\{S^1,S^2\}$ 称为 S 的单纯形对分, 其中 S^1, S^2 的顶点分别为

$$\{V^1,V^2,\cdots,V^{s-1},c,V^{s+1},\cdots,V^{n+1}\},\ \{V^1,V^2,\cdots,V^{\tilde{s}-1},c,V^{\tilde{s}+1},\cdots,V^{n+1}\}.$$

根据文献 [9] 知, 单纯形对分过程是穷举的, 即如果 $\{S^{\tilde{r}}\}$ 表示由上述分支过程所形成的一嵌套序列 (i.e. 对于所有 $\tilde{r}$, $S^{\tilde{r}+1}\subseteq S^{\tilde{r}}$), 则存在 $x\in R^n$ 使得 $\bigcap_{\tilde{r}}S^{\tilde{r}}=\{x\}$.

2.2 下界

在这一部分, 我们将介绍如何计算 $f(x)$ 在 $S\bigcap D$ 上的下界 $LB(S)$, 其中 $S=[V^1,V^2,\cdots,V^{n+1}]$ 表示初始单纯形 S^0 或者它的子单纯形. 这一过程是本文算法的实质性内容, 其过程依赖于下面定理.

定理 1 令 U 表示以顶点 $V^1,V^2,\cdots,V^{n+1}$ 为列构成的矩阵, $e=(1,1,\cdots,1)\in R^{n+1}$, 且对每个 $j\in\{1,2,\cdots,n+1\}$, θ_j 为下面线性规划问题的最优值:

$$(\mathrm{LP})_j\left\{\begin{aligned}
\min\ & \sum_{i=1}^{p}\frac{y_i}{-n_i(V^j)}\\
\text{s.t.}\ & AU\lambda\le b\\
& U\lambda\ge 0\\
& y_i+c_i^{\mathrm{T}}U\lambda=-\alpha_i,\ i=1,2,\cdots,p\\
& e\lambda=1\\
& \lambda\ge 0
\end{aligned}\right.$$

则 $f(x)$ 在 $S\bigcap D$ 上的下界 $LB(S)$ 可通过下面的线性规划求解确定

$$(\mathrm{LP})\left\{\begin{aligned}
LB(S)=\ \min\ & \sum_{j=1}^{n+1}\theta_j\lambda_j\\
\text{s.t.}\ & AU\lambda\le b\\
& U\lambda\ge 0\\
& e\lambda=1\\
& \lambda\ge 0
\end{aligned}\right.$$

如果 (LP) 可行域为空集, 则令 $LB(S)=+\infty$.

证明 定义函数 $g: R^n \to R$ 为

$$g(x)=\min_{\xi,y}\left\{\sum_{i=1}^{p}\frac{y_i}{-n_i(x)}\,\middle|\; y_i+c_i^{\mathrm{T}}\xi=-\alpha_i,\ i=1,2,\cdots,p,\ \xi\in S\bigcap D\right\} \tag{1}$$

则, 当 $S\bigcap D\neq\emptyset$ 时, $g(x)$ 是一关于 x 的凹函数. 从而可构造 $g(x)$ 在 S 上的凸包络 $\delta(x)$ 如下:

$$\delta(x)=\sum_{j=1}^{n+1}g(V^j)\lambda_j,$$

其中 $\lambda=(\lambda_1,\lambda_2,\cdots,\lambda_{n+1})$ 满足 $U\lambda=x,\ e\lambda=1,\ \lambda\geq 0$.

根据凸包络的定义, 对所有 $x\in S$, $\delta(x)\leq g(x)$, 因此有

$$\begin{aligned}&\min\left\{\sum_{i=1}^{p}\frac{c_i^{\mathrm{T}}x+\alpha_i}{n_i(x)}\,\middle|\,x\in S\bigcap D\right\}=\min\left\{\sum_{i=1}^{p}\frac{y_i}{-n_i(x)}\,\middle|\,y_i+c_i^{\mathrm{T}}x=-\alpha_i,\ i=1,2,\cdots,p,\ x\in S\bigcap D\right\}\\ &\geq\min_{x\in S\bigcap D}\left\{\min_{\xi,y}\left\{\sum_{i=1}^{p}\frac{y_i}{-n_i(x)}\,\middle|\,y_i+c_i^{\mathrm{T}}\xi=-\alpha_i,\ i=1,2,\cdots,p,\ \xi\in S\bigcap D\right\}\right\}=\min_{x\in S\bigcap D}g(x)\geq\min_{x\in S\bigcap D}\delta(x)\end{aligned} \tag{2}$$

另外, 我们有下列关系成立

$$\begin{aligned}x\in S\bigcap D\quad&\Leftrightarrow\quad\lambda\in\{\lambda\mid AU\lambda\leq b,\ U\lambda\geq 0,\ e\lambda=1,\ \lambda\geq 0\},\\ y_i+c_i^{\mathrm{T}}\xi=-\alpha_i\quad&\Leftrightarrow\quad y_i+c_i^{\mathrm{T}}U\lambda=-\alpha_i,\ i=1,2,\cdots,p.\end{aligned}$$

结合 (2) 式, 可计算 $f(x)$ 在 $S\bigcap D$ 上的下界 $LB(S)$:

$$LB(S)=\min_{x\in S\bigcap D}\delta(x)=\min\left\{\sum_{j=1}^{n+1}g(V^j)\lambda_j\,\middle|\,AU\lambda\leq b,\ U\lambda\geq 0,\ e\lambda=1,\ \lambda\geq 0\right\},$$

其中对于每个 $j\in\{1,2,\cdots,n+1\}$,

$$g(V^j)=\min\left\{\sum_{i=1}^{p}\frac{y_i}{-n_i(V^j)}\,\middle|\,AU\lambda\leq b,\ y_i+c_i^{\mathrm{T}}U\lambda=-\alpha_i,\ i=1,2,\cdots,p,\ U\lambda\geq 0,\ e\lambda=1,\ \lambda\geq 0\right\}=\theta_j.$$

下面定理说明了由算法确定的下界序列 $\{LB_k\}$ 是单调增加的.

定理 2 令 S, $\bar{S}$ 是两个 n 维单纯形, 且 $\bar{S}\subseteq S$. 则 $LB(S)\leq LB(\bar{S})$.

证明 记 $S=[V^1,V^2,\cdots,V^{n+1}]$, $\bar{S}=[\bar{V}^1,\bar{V}^2,\cdots,\bar{V}^{n+1}]$.

(i) 若 $\bar{S}\bigcap D=\emptyset$, 则 $LB(\bar{S})=+\infty$, 结论显然成立.

(ii) 若 $\bar{S}\bigcap D\neq\emptyset$. 令 $g(x),\bar{g}(x)$ 分别为由 (1) 式确定的在 S, $\bar{S}$ 上的凹函数, $\delta(x)$, $\bar{\delta}(x)$ 分别为 $g(x),\bar{g}(x)$ 的凸包络. 因为 $\bar{S}\subseteq S$, 所以, 对于所有 $x\in\bar{S}$, 有 $\bar{g}(x)\geq g(x)$. 进而有 $\bar{\delta}(x)\geq\delta(x)$. 综上可知

$$LB(\bar{S})=\min\{\bar{\delta}(x)\mid x\in\bar{S}\bigcap D\}\geq\min\{\delta(x)\mid x\in\bar{S}\bigcap D\}\geq\min\{\delta(x)\mid x\in S\bigcap D\}=LB(S).$$

2.3 上界

对于每个由算法产生的 n 维单纯形 S, 若 $LB(S)$ 有限, 则算法将产生问题 (P) 的一些可行解. 随着算法的进行, 会得到越来越多的可行解, 这些可行解可用于上界的更新. 具体过程如下: 假设 UB 为当前最好上界, 令 (λ^j,y^j) 为线性规划 $(\mathrm{LP})_j(j=1,2,\cdots,n+1)$ 在单纯形 S 上的最优解, 则 $x^j=U\lambda^j(j=1,2,\cdots,n+1)$ 为问题 (P) 的可行解. 此外, 令 λ^* 为线性规划问题 (LP) 在单纯形 S 上的最优解, 则 $x^*=U\lambda^*$ 是问题 (P) 的可行解. 由此可知在计算下界 $LB(S)$ 的同时, 我们可以得到一个可行点集 $F(S)=\{x^1,\ x^2,\cdots,x^{n+1},x^*\}$, 从而可用下式更新上界

$$UB=\min\{f(x^1),f(x^2),\cdots,f(x^{n+1}),f(x^*),UB\}.$$

3 算法及其收敛性

在前文基础上, 下面给出算法过程的具体描述.

分支定界算法

步 0 选取 $\epsilon\geq 0$. 构造包含问题 (P) 可行域 D 的 n 维单纯形 $S^0\subseteq R^n$; 计算 $f(x)$ 在 $S^0\bigcap D$ 上的下界 $LB(S^0)$; 确定有限可行集 $F(S^0)\subseteq S^0\bigcap D$; 置 $F=F(S^0)$, $LB_0=LB(S^0)$, $UB_0=\min\{f(x)\mid x\in F\}$; 选

取点 $x^0 \in F$ 使得 $f(x^0) = UB_0$. 若 $UB_0 - LB_0 \leq \epsilon$, 停: x^0 是问题 (P) 的 ϵ- 最优解, 且 UB_0 是 ϵ- 最优值. 否则, 置 $P_0 = \{S^0\}$, $k = 1$, 并转步 1.

步 1 使用单纯形对分规则将 S^{k-1} 剖分为两个子单纯形 $S^{k,1}, S^{k,2}$.

步 2 对于 $i = 1, 2$, 计算 $f(x)$ 在 $S^{k,i} \bigcap D$ 的下界 $LB(S^{k,i})$, 并确定有限集 $F(S^{k,i}) \subseteq S^{k,i} \bigcap D$.

步 3 置 $F = F \bigcup \{F(S^{k,i}) \mid i = 1, 2\}$; $UB_k = \min\{f(x) \mid x \in F\}$; 选取 $x^k \in F$ 使得 $f(x^k) = UB_k$.

步 4 置 $P_k = P_{k-1} \setminus \{S^{k-1}\} \bigcup \{S^{k,i} \mid i = 1, 2, LB(S^{k,i}) < UB_k\}$.

步 5 置 $LB_k = \min\{LB(S) \mid S \in P_k\}$, 并令 $S^k \in P_k$ 为满足 $LB_k = LB(S^k)$ 的单纯形. 若 $UB_k - LB_k \leq \epsilon$, 停: x^k 是问题 (P) 的 ϵ- 最优解, 且 UB_k 是 ϵ- 最优值. 否则, 置 $k = k + 1$, 转步 1.

下面定理给出算法的收敛性.

定理 3 (a) 若算法有限步终止, 则算法终止时可求得问题 (P) 的一个 ϵ- 全局最优解.

(b) 若算法无限步终止, 假定 $\{S^q\}$ 为由算法产生的一无限递减序列, 且满足 $\bigcap_{q=1}^{\infty} S^q = \{\bar{x}\}$, 其中 $\bar{x} \in D$. 则可行解序列 $\{x^q\}$ 的任一聚点是问题 (P) 的全局最优解.

证明 (a) 当算法有限步终止时, 结论显然.

(b) 当算法无限步终止时, 对于每个 q, 令 $x^{qj} (j = 1, 2, \cdots, n+1)$ 和 x^{*q} 分别为求解 $(\mathrm{LP})_j (j = 1, 2, \cdots, n+1)$ 和 (LP) 在单纯形 S^q 上最优值时所得到的可行解. 因为随着 $q \to \infty$, S^q 收敛到 $\bar{x}$, 所以随着 $q \to \infty$, 有 $x^{qj} \to \bar{x} (j = 1, 2, \cdots, n+1)$ 及 $x^{*q} \to \bar{x}$.

令 $V^{qj} (j = 1, 2, \cdots, n+1)$ 表示 S^q 的顶点. 由以上讨论可知, $V^{qj} \to V^{*j} = \bar{x} (j = 1, 2, \cdots, n+1)$, 从而有 $\lim\limits_{q \to \infty} g(V^{qj}) = g(V^{*j}) = f(\bar{x}) (j = 1, 2, \cdots, n+1)$. 因此, $\lim\limits_{q \to \infty} LB_q = \lim\limits_{q \to \infty} LB(S^q) = \sum_{j=1}^{n+1} g(V^{*j}) \lambda_j = f(\bar{x}) \sum_{j=1}^{n+1} \lambda_j = f(\bar{x})$, 进而有 $\lim\limits_{q \to \infty} (UB_q - LB_q) = \lim\limits_{q \to \infty} UB_q - f(\bar{x}) = \lim\limits_{q \to \infty} f(x^q) - f(\bar{x}) = 0$. 根据文献 [10], 知结论 (b) 成立.

4 数值实验

为验证算法的可行性及有效性, 我们做了一些数值实验. 程序编写采用 Matlab 7.1, 数值实验在 Pentium IV (3.06 GHZ) 微机上进行. 算法中的线性规划问题使用单纯形方法求解. 在这些数值实验中, 收敛性误差取为 $\epsilon = 1.0\mathrm{E} - 3$.

例 1[5]

$$\begin{array}{ll} \min & \dfrac{x_1 + 3x_2 + 2}{4x_1 + x_2 + 3} + \dfrac{4x_1 + 3x_2 + 1}{x_1 + x_2 + 4} \\ \text{s.t.} & -x_1 - x_2 \leq -1 \\ & x_1 \geq 0, x_2 \geq 0 \end{array}$$

例 2[11]

$$\begin{array}{ll} \min & \dfrac{-x_1 + 2x_2 + 2}{3x_1 - 4x_2 + 5} + \dfrac{4x_1 - 3x_2 + 4}{-2x_1 + x_2 + 3} \\ \text{s.t.} & x_1 + x_2 \leq 1.5 \\ & x_1 - x_2 \leq 0 \\ & 0 \leq x_1 \leq 1,\ 0 \leq x_2 \leq 1 \end{array}$$

例 3[4,5]

$$\begin{array}{ll} \min & \dfrac{-3x_1 - 5x_2 - 3x_3 - 50}{3x_1 + 4x_2 + 5x_3 + 50} + \dfrac{-3x_1 - 4x_3 - 50}{4x_1 + 3x_2 + 2x_3 + 50} + \dfrac{-4x_1 - 2x_2 - 4x_3 - 50}{5x_1 + 4x_2 + 3x_3 + 50} \\ \text{s.t.} & 6x_1 + 3x_2 + 3x_3 \leq 10 \\ & 10x_1 + 3x_2 + 8x_3 \leq 10 \\ & x_1 \geq 0, x_2 \geq 0, x_3 \geq 0 \end{array}$$

例 4[12]

$$\begin{array}{ll} \min & \sum\limits_{i=1}^{5} \dfrac{c_i^{\mathrm{T}} x + d_i}{e_i^{\mathrm{T}} x + f_i} \\ \text{s.t.} & Ax \leq b \\ & x \geq 0 \end{array}$$

其中

$$c_1=(0,\ 0.1,\ 0.3,\ -0.3,\ -0.5,\ -0.5,\ 0.8,\ -0.4,\ 0.4,\ -0.2,\ -0.2,\ 0.1)^{\mathrm{T}},\qquad d_1=-14.6,$$
$$e_1=(-0.3,\ -0.1,\ -0.1,\ -0.1,\ 0.1,\ 0.4,\ 0.2,\ -0.2,\ 0.4,\ 0.2,\ -0.4,\ 0.3)^{\mathrm{T}},\qquad f_1=14.2,$$
$$c_2=(-0.2,\ -0.5,\ 0,\ -0.4,\ -0.1,\ 0.6,\ 0.1,\ 0.2,\ 0.2,\ -0.1,\ -0.2,\ -0.3)^{\mathrm{T}},\qquad d_2=-7.1,$$
$$e_2=(0,\ 0.1,\ -0.1,\ 0.3,\ 0.3,\ -0.2,\ 0.3,\ 0,\ -0.4,\ 0.5,\ -0.3,\ 0.1)^{\mathrm{T}},\qquad f_2=1.7,$$
$$c_3=(0.1,\ -0.3,\ 0,\ -0.1,\ 0.1,\ 0,\ -0.3,\ 0.2,\ 0,\ -0.3,\ -0.5,\ -0.3)^{\mathrm{T}},\qquad d_3=-1.7,$$
$$e_3=(0.8,\ -0.4,\ 0.7,\ -0.4,\ -0.4,\ 0.5,\ -0.2,\ -0.8,\ 0.5,\ 0.6,\ -0.2,\ 0.6)^{\mathrm{T}},\qquad f_3=8.1,$$
$$c_4=(0.1,\ -0.5,\ -0.1,\ -0.1,\ 0.2,\ 0.5,\ -0.6,\ -0.7,\ -0.5,\ -0.7,\ 0.1,\ -0.1)^{\mathrm{T}},\qquad d_4=-4,$$
$$e_4=(0,\ 0.6,\ -0.3,\ 0.3,\ 0,\ 0.2,\ 0.3,\ -0.6,\ -0.2,\ -0.5,\ 0.8,\ -0.5)^{\mathrm{T}},\qquad f_4=26.9,$$
$$c_5=(-0.7,\ 0.5,\ -0.1,\ -0.2,\ 0.1,\ 0.3,\ 0,\ 0.1,\ 0.2,\ -0.6,\ -0.5,\ 0.2)^{\mathrm{T}},\qquad d_5=-6.8,$$
$$e_5=(0.4,\ 0.2,\ -0.2,\ 0.9,\ 0.5,\ -0.1,\ 0.3,\ -0.8,\ -0.2,\ 0.6,\ -0.2,\ -0.4)^{\mathrm{T}},\qquad f_5=3.7,$$

$$A=\begin{pmatrix}
-1.8 & -2.2 & 0.8 & 4.1 & 3.8 & -2.3 & -0.8 & 2.5 & -1.6 & 0.2 & -4.5 & -1.8\\
4.6 & -2.0 & 1.4 & 3.2 & -4.2 & -3.3 & 1.9 & 0.7 & 0.8 & -4.4 & 4.4 & 2.0\\
3.7 & -2.8 & -3.2 & -2.0 & -3.7 & 3.3 & 3.5 & -0.7 & 1.5 & -3.1 & 4.5 & -1.1\\
-0.6 & -0.6 & -2.5 & 4.1 & 0.6 & 3.3 & 2.8 & -0.1 & 4.1 & -3.2 & -1.2 & -4.3\\
1.8 & -1.6 & -4.5 & -1.3 & 4.6 & 3.3 & 4.2 & -1.2 & 1.9 & 2.4 & 3.4 & -2.9\\
-0.5 & -4.1 & 1.7 & 3.9 & -0.1 & -3.9 & -1.5 & 1.6 & 2.3 & -2.3 & -3.2 & 3.9\\
0.3 & 1.7 & 1.3 & 4.7 & 0.9 & 3.9 & -0.5 & -1.2 & 3.8 & 0.6 & -0.2 & -1.5\\
0.5 & -4.2 & 3.6 & -0.6 & -4.8 & 1.5 & -0.3 & 0.6 & -3.6 & 0.2 & 3.8 & -2.8\\
-0.1 & 3.3 & -4.3 & 2.4 & 4.1 & 1.7 & 1.0 & -3.3 & 4.4 & -3.7 & -1.1 & -1.4\\
-0.6 & 2.2 & 2.5 & 1.3 & -4.3 & -2.9 & -4.1 & 2.7 & -0.8 & -2.9 & 3.5 & 1.2\\
4.3 & 1.9 & -4.0 & -2.6 & 1.8 & 2.5 & 0.6 & 1.3 & -4.3 & -2.3 & 4.1 & -1.1\\
0.0 & 0.4 & -4.5 & -4.4 & 1.2 & -3.8 & -1.9 & 1.2 & 3.0 & -1.1 & -0.2 & 2.5\\
-0.1 & -1.7 & 2.9 & 1.5 & 4.7 & -0.3 & 4.2 & -4.4 & -3.9 & 4.4 & 4.7 & -1.0\\
-3.8 & 1.4 & -4.7 & 1.9 & 3.8 & 3.5 & 1.5 & 2.3 & -3.7 & -4.2 & 2.7 & -0.1\\
0.2 & -0.1 & 4.9 & -0.9 & 0.1 & 4.3 & 1.6 & 2.6 & 1.5 & -1.0 & 0.8 & 1.6
\end{pmatrix}$$

$$b=(15.7,31.8,-36.4,38.5,40.3,10.0,89.8,5.8,2.7,-16.3,-14.6,-72.7,57.7,-34.5,69.1)^{\mathrm{T}}.$$

例 5

$$\begin{aligned}
\min\quad & \frac{3x_1+4x_2+10}{-x_1^2+2x_1-x_2^2+15}+\frac{3x_1+5x_2+6}{2x_1+4x_2+3}+\frac{2x_1+3x_2+1}{-x_1^2+18}\\
\text{s.t.}\quad & x_1+x_2\le 4\\
& -x_1+x_2\le 4\\
& x_1\ge 0, x_2\ge 0
\end{aligned}$$

例 6 考虑下面的随机问题

$$\begin{aligned}
\min\quad & \sum_{i=1}^{p}\frac{c_i^{\mathrm{T}}x+d_i}{e_i^{\mathrm{T}}x+f_i}\\
\text{s.t.}\quad & Ax\le b\\
& x\ge 0
\end{aligned}$$

其中 c_i, $e_i\in R^n$, d_i, $f_i\in R$ 是在区间 $[-0.5,0.5]$ 上随机产生, $A\in R^{m\times n}$, $b\in R^m$ 是在区间 $[0,0.5]$ 上随机产生的.

例 1 至例 5 的计算结果见表 1. 对于例 6, 每种情况, 我们做 10 次随机实验, 然后取计算结果的平均值. 关于例 6 的具体计算结果见表 2. 从这些数值实验结果的比较可以看出, 我们的方法是有效可行的.

表 1 例 1–5 的计算结果及比较

例	方法	最优解	最优值	迭代次数
1	[5]	(1.0, 0.0)	1.428571	10
	ours	(1.0, 0.0)	1.4286	1
2	[11]	(0.0, 0.283935547)	1.623183358	71
	ours	(0.0, 0.2813)	1.6232	42
3	[4]	(0.0, 3.33333, 0.0)	–3.002924	66
	[5]	(0.0, 3.33329, 0.0)	–3.000042	30
	ours	(0.0, 3.3333, 0.0)	–3.0029	26
4	[12]	(6.224297,20.059738,3.774868,5.947937,0.0,7.456478, 0.0,23.312241,0.000204,41.031278,0.0,3.171060)	–16.077978	11
	ours	(6.2237,20.0603,3.7747,5.9478,0.0,7.4567, 0.0,23.3126,0.0,41.0318,0.0,3.1711)	–16.0780	1
5	ours	(0.0, 4.0)	–23.9094	1

表 2 例 6 的计算结果

p	(m, n)	平均运行时间 (单位秒)	平均迭代次数
3	(5,10)	1.2318	2
	(10,20)	13.2568	4.1
6	(5,10)	2.4282	2.4
	(10,20)	14.4695	4.3
9	(5,10)	15.1346	7.1
	(10,20)	29.0426	9.8

参考文献

[1] Konno H, Yamashita H. Minimizing sums and products of linear fractional functions over a polytope[J]. Naval Research Logistics, 1999, 46: 583–596.

[2] Konno H, Abe N. Minimization of the sum of three linear fractional functions[J]. Journal of Global Optimization, 1999, 15: 419–432.

[3] Kuno T. A branch-and-bound algorithm for maximizing the sum of several linear ratios[J]. Journal of Global Optimization, 2002, 22: 155–174.

[4] Wang Y J, Shen P P, Liang Z A. A branch-and-bound algorithm to globally solve the sum of several linear ratios[J]. Applied Mathematics and Computation, 2002, 168: 89–101.

[5] Ji Y, Zhang K C, Qu S J. A deterministic global optimization algorithm[J]. Applied Mathematics and Computation, 2002, 185: 382–387.

[6] Wang C F, Shen P P. A global optimization algorithm for linear fractional programming[J]. Applied Mathematics and Computation, 2002, 204: 281–287.

[7] Benson H P. A simplicial branch and bound duality-bounds algorithm for the linear sum-of-ratios problem[J]. European Journal of Operational Research, 2007, 182: 597–611.

[8] 汪春峰, 李娟, 申培萍. 线性比式和问题的全局优化算法 [J]. 河南师范大学学报: 自然科学版, 2010, 38: 4–7.
Wang C F, Li J, Shen P P. A global optimization algorithm for sum of linear ratios problem[J]. Journal of Henan Normal University: Natural Science, 2010, 38: 4–7.

[9] Horst R, Pardalos P M, Thoai N V. Introduction to global optimization[M]. Kluwer, Dordrecht, Netherlands, 1995.

[10] Horst R, Tuy H. Global optimization: Deterministic approaches[M]. Springer, Berlin, 1996.

[11] Jiao H W. A branch and bound algorithm for globally solving a class of nonconvex programming problems[J]. Nonlinear Analysis: Theory, Methods and Applications, 2009, 70: 1113–1123.

[12] Phuong NTH, Tuy H. A unified monotonic approach to generalized linear fractional programming[J]. Journal of Global Optimization, 2003, 26: 229–259.