应 用 数 学
MATHEMATICA APPLICATA
2019, 32(4): 767-777

求二次比式和问题全局解的一个新的确定性算法

张博[1,2], 高岳林[1,2]
(1. 北方民族大学数学与信息科学学院, 宁夏 银川 750021; 2. 宁夏科学计算与智能信息处理协同创新中心, 宁夏 银川 750021)

摘要: 本文研究一类二次比式和规划问题. 首先, 利用等价转换的方法把原问题转化为一个非线性规划问题, 并且这个非线性规划问题的目标函数通项的分子和分母都分别是两项线性函数乘积和再加上一个线性函数的形式, 再根据两项线性函数乘积和的特性, 对目标函数进行线性松弛, 以确定原问题最优值的下界, 从而提出一个求解线性规划问题的分支定界算法, 并证明该算法的收敛性. 最后, 数值结果表明所提出的算法是可行有效的.

关键词: 全局最优化; 分式规划; 二次函数; 线性乘积和规划; 分支定界; 线性规划
中图分类号: O221.2 **AMS(2000) 主题分类:** 90C30; 90C32
文献标识码: A **文章编号:** 1001-9847(2019)04-0767-11

1. 引言

分式规划问题是非线性全局优化的一个重要分支, 而二次比式和规划问题是又一类特殊的分式规划问题. 在现实生活中, 许多实际问题均可抽象为二次比式和规划模型, 且在投资问题、运输方案、经济效益、生产管理等领域有着广泛的应用背景. 数十年来它吸引了许多专家和学者的高度关注; 其次, 从研究的观点来看, 二次比式和规划问题对理论分析和计算求解的方式提出了挑战. 本文主要考虑以下形式的二次比式和规划问题(QFP):

$$
(\mathrm{QFP}):\begin{cases}\min\ G(x)=\displaystyle\sum_{i=1}^{p}\frac{H_{1i}(x)}{H_{2i}(x)}\\ \text{s.t.}\quad x\in X=\{x\in\mathbb{R}^n|Ax\le b\},\end{cases}
$$

这里, $p\ge 2, H_{1i}(x), H_{2i}(x)$均为二次函数其表达式如下:

$$
H_{1i}(x)=x^{\mathrm{T}}Q_{1i}x+q_{1i}^{\mathrm{T}}x+p_{1i},
$$
$$
H_{2i}(x)=x^{\mathrm{T}}Q_{2i}x+q_{2i}^{\mathrm{T}}x+p_{2i},
$$

其中, $b\in\mathbb{R}^n$, $A\in\mathbb{R}^{m\times n}$, $Q_{si}\in\mathbb{R}^{n\times n}$, $q_{si}\in\mathbb{R}^n, p_{si}\in\mathbb{R}$, $s=1,2,X$是非空有界闭集. 根据$H_{2i}(x)$的连续性以及介值定理可知, 对任意的$x\in X$有$H_{2i}(x)>0$或者$H_{2i}(x)<0$. 如果存在某一个$i\in\{1,2,\cdots,p\}$使得$H_{2i}(x)<0$, 用$\frac{-H_{1i}(x)}{-H_{2i}(x)}$ 替代$\frac{H_{1i}(x)}{H_{2i}(x)}$, 使得所有分式项的分母项均大于零; 为了方便起见, 假设此时, $H_{1i}(x)<0, H_{2i}(x)>0$, 用 $\frac{H_{1i}(x)+MH_{2i}(x)}{H_{2i}(x)}$替代$\frac{H_{1i}(x)}{H_{2i}(x)}$, 其

* **收稿日期:** 2018-07-27
基金项目: 国家自然科学基金项目(61561001, 11161001), 宁夏高等教育一流学科建设基金(NXYLXK 2017B09), 北方民族大学研究生创新项目(YCX18084)
作者简介: 张博, 男, 汉族, 陕西人, 研究方向: 最优化理论与方法.
通迅作者: 高岳林.

中M是一个充分大的正数, 满足对所有的$x\in X, H_{1i}(x)+MH_{2i}(x)\geq 0$, 问题(QFP)的本质不变. 因此, 这里不失一般性, 假定对所有的$i=1,2,\cdots,p$, 均有$H_{1i}(x)\geq 0, H_{2i}(x)>0$.

此外, 对于问题(QFP)可能拥有多个局部最优解,这样会干扰寻找全局最优解, 使问题的难度增加, 因此研究此类问题是必要的. 本文为上述比式和规划问题建立了一个分支定界算法. 首先, 通过线性代数矩阵满秩分解的知识将原问题中通项的分子和分母的二次函数分别转化为相应的两项乘积和的形式,然后利用求解两项线性乘积和规划的知识[1]并结合一些线性化技巧分别构造问题目标函数中通项的下界, 并基于此技巧构造出能够为原问题提供可靠下界的线性松弛规划问题. 最后, 利用分支定界的思想以及文[2]的可行域缩减技术设计出关于本文问题的一个新的分支定界算法.

到目前为止, 已经有许多全局优化算法可以用来求解比式和规划问题. 当通项中的分子和分母都为线性函数时, 那么文[3-9]给出了不同的求解方法, 当问题为非线性比式和规划问题时, ZHANG, CHEN和JIA[10]提出了一种基于单调优化理论的算法用于求解此类问题. JIAO和LIU[11]构造了一种区间划分压缩算法并用求解二次比式和规划问题. 最近, JIAO和LIU[13]他们又提出了一种基于参数线性松弛规划问题来确定下界的方法, 并构造了相应的分支定界算法来求解二次比式和规划问题. 文[14]为广义多项式比式和问题, GAO, Mishra和SHI讨论了函数比之和的最大化问题, 推广了一种求解线性比和问题的算法,并给出了问题的复杂性. 文[15]给出了一种基于单纯形剖分的确定性非线性比式和问题全局优化算法. 关于其他类型非线性比式和规划问题还可以参考文[16-17].

本文后续章节布局如下: 第二节给出问题预处理时必要的两个引理; 第三节利用已知的两个引理得出原问题的一个等价问题并给出等价问题的线性松弛过程; 第四节给出了新的分支定界算法以及相应的超矩形剖分方法和缩减技术, 并证明了算法的收敛性; 第五节通过数值实验说明了我们提出的算法是有效可行的; 第六节为结论.

2. 预备知识

为了求解问题(QFP), 我们首先将其转化为一个具有特殊结构的等价问题以便建立线性松弛问题, 为了验证这种转化的可能性, 我们首先引入以下引理:

引理2.1[12] 对任意矩阵Q满足秩$(Q)=1$, 存在两个向量α和β, 使得$Q=\alpha\beta^{\mathrm{T}}$.

引理2.2[12] 对任意矩阵$Q_{si}\in\mathbb{R}^{n\times n}$, 假定秩$(Q_{si})=r_{si}$, 则一定存在两个向量$\alpha_{sij}$和$\beta_{sij}$, 使得$Q_{si}=\sum_{j=1}^{r_{si}}\alpha_{sij}\beta_{sij}^{\mathrm{T}}$. 其中$\alpha_{sij}\in\mathbb{R}^n, \beta_{sij}\in\mathbb{R}^n, i=1,2,\cdots,p, s=1,2$.

3. 等价问题以及线性松弛规划

根据以上两个引理 2.1 和 2.2 我们知道, 对每一个$i=1,2,\cdots,m, s=1,2$, 二次函数$H_{si}(x)$都可转化为以下形式:

$$\mathrm{H}_{si}(x)=x^{\mathrm{T}}Q_{si}x+q_{si}^{\mathrm{T}}x+p_{si}=\sum_{j=1}^{r_{si}}(\alpha_{sij}^{\mathrm{T}}x)(\beta_{sij}^{\mathrm{T}}x)+q_{si}^{\mathrm{T}}x+p_{si}. \tag{3.1}$$

那么问题(QFP)可以进一步转化为以下等价问题:

$$(\mathrm{EQFP}):\begin{cases}\min\ G(x)=\displaystyle\sum_{i=1}^{p}\frac{\sum_{j=1}^{r_{1i}}(\alpha_{1ij}^{\mathrm{T}}x)(\beta_{1ij}^{\mathrm{T}}x)+q_{1i}^{\mathrm{T}}x+p_{1i}}{\sum_{j=1}^{r_{2i}}(\alpha_{2ij}^{\mathrm{T}}x)(\beta_{2ij}^{\mathrm{T}}x)+q_{2i}^{\mathrm{T}}x+p_{2i}}\\ \text{s.t. } x\in X=\{x\in\mathbb{R}^n|Ax\leq b\},\end{cases}$$

其中,$\alpha_{sij}\in\mathbb{R}^n$, $\beta_{sij}\in\mathbb{R}^n$, $q_{si}\in\mathbb{R}^n, p_{si}\in\mathbb{R}, i=1,2,\cdots,p, s=1,2, X$是非空有界闭集.

为了获得问题(QFP)的分支定界算法, 我们需要确立线性松弛规划子问题, 用于估计问题(QFP)最优值的下界, 具体方法如下:

首先, 求解以下$2n$个线性规划问题:

$$\underline{x}_k = \min_{x\in X} x_k, \overline{x}_k = \max_{x\in X} x_k, k = 1, 2, \cdots, n.$$

从而得到包含可行域X的初始超矩形$H = [\underline{x}, \overline{x}] = \{x|\underline{x}_k \leq x \leq \overline{x}_k, k = 1, 2, \cdots, n\}$.

然后, 对每一个$i = 1, 2, \cdots, p, s = 1, 2, j = 1, 2, \cdots, r_{si}$, 继续求解以下线性规划问题:

$$l_{sij} = \min_{x\in X\cap H} \alpha_{sij}^{\mathrm{T}} x, u_{sij} = \max_{x\in X\cap H} \alpha_{sij}^{\mathrm{T}} x, \tag{3.2}$$

$$w_{sij} = \min_{x\in X\cap H} \beta_{sij}^{\mathrm{T}} x, v_{sij} = \max_{x\in X\cap H} \beta_{sij}^{\mathrm{T}} x. \tag{3.3}$$

根据式(3.2)和(3.3)很容易得到

$$\alpha_{sij}^{\mathrm{T}} x - l_{sij} \geq 0, \tag{3.4}$$

$$\alpha_{sij}^{\mathrm{T}} x - u_{sij} \leq 0, \tag{3.5}$$

$$\beta_{sij}^{\mathrm{T}} x - w_{sij} \geq 0, \tag{3.6}$$

$$\beta_{sij}^{\mathrm{T}} x - v_{sij} \leq 0. \tag{3.7}$$

再由(3.4)-(3.7), 不难得出

$$(\alpha_{sij}^{\mathrm{T}} x)(\beta_{sij}^{\mathrm{T}} x) \geq \max\{\underline{\phi}_{sij}(x), \underline{\varphi}_{sij}(x)\}, \tag{3.8}$$

$$(\alpha_{sij}^{\mathrm{T}} x)(\beta_{sij}^{\mathrm{T}} x) \leq \min\{\overline{\phi}_{sij}(x), \overline{\varphi}_{sij}(x)\}, \tag{3.9}$$

这里,

$$\begin{gathered}
\underline{\phi}_{sij}(x) = l_{sij}\beta_{sij}^{\mathrm{T}} x + w_{sij}\alpha_{sij}^{\mathrm{T}} x - l_{sij}w_{sij}, \\
\underline{\phi}_{sij}(x) = u_{sij}\beta_{sij}^{\mathrm{T}} x + v_{sij}\alpha_{sij}^{\mathrm{T}} x - u_{sij}v_{sij}, \\
\underline{\phi}_{sij}(x) = l_{sij}\beta_{sij}^{\mathrm{T}} x + v_{sij}\alpha_{sij}^{\mathrm{T}} x - l_{sij}v_{sij}, \\
\underline{\phi}_{sij}(x) = u_{sij}\beta_{sij}^{\mathrm{T}} x + w_{sij}\alpha_{sij}^{\mathrm{T}} x - u_{sij}w_{sij}.
\end{gathered}$$

为了方便起见, 这里记

$$\begin{gathered}
h_{sij}(x) = x^{\mathrm{T}} Q_{si} x = (\alpha_{sij}^{\mathrm{T}} x)(\beta_{sij}^{\mathrm{T}} x), \\
\underline{h}_{sij} = \max\{\underline{\phi}_{sij}(x), \underline{\varphi}_{sij}(x)\}, \ \overline{h}_{sij} = \min\{\overline{\phi}_{sij}(x), \overline{\varphi}_{sij}(x)\}.
\end{gathered}$$

定理3.1　令$\varepsilon_k = \overline{x}_k - \underline{x}_k (k = 1, 2, \cdots, n)$, 则对所有的$x \in X \cap H$, 当$\varepsilon_k \to 0$时,$h_{sij}(x) - \underline{h}_{sij}(x) \to 0, \overline{h}_{sij}(x) - h_{sij}(x) \to 0$, $i = 1, 2, \cdots, p, s = 1, 2, j = 1, 2, \cdots, r_{si}$.

证　对所有的$x \in X \cap H$, 由式(3.8)得:

$$\begin{aligned}
0 &\leq h_{sij}(x) - \underline{h}_{sij}(x) \\
&= (\alpha_{sij}^{\mathrm{T}} x)(\beta_{sij}^{\mathrm{T}} x) - \max\{\underline{\phi}_{sij}(x), \underline{\varphi}_{sij}(x)\} \\
&= \min\{(\alpha_{sij}^{\mathrm{T}} x - l_{sij})(\beta_{sij}^{\mathrm{T}} x - w_{sij}), (\alpha_{sij}^{\mathrm{T}} x - u_{sij})(\beta_{sij}^{\mathrm{T}} x - v_{sij})\} \\
&\leq \min\{(u_{sij} - l_{sij})(v_{sij} - w_{sij}), (l_{sij} - u_{sij})(w_{sij} - v_{sij})\} \\
&= \min\{|u_{sij} - l_{sij}| \cdot |v_{sij} - w_{sij}|, |l_{sij} - u_{sij}| \cdot |w_{sij} - v_{sij}|\} \\
&= |u_{sij} - l_{sij}| \cdot |v_{sij} - w_{sij}|.
\end{aligned} \tag{3.10}$$

另外, 假设$l_{sij} = \alpha_{sij}^{\mathrm{T}} x_1, u_{sij} = \alpha_{sij}^{\mathrm{T}} x_2$, 那么$|u_{sij} - l_{sij}| = |\alpha_{sij}^{\mathrm{T}}(x_2 - x_1)|$, 当$\varepsilon_k \to 0, k = 1, 2, \cdots, n$时, $\|\overline{x} - \underline{x}\| \to 0 \Rightarrow \|x_2 - x_1\| \to 0$, 所以$|u_{sij} - l_{sij}| \to 0$. 同理可证$|v_{sij} - w_{sij}| \to 0$. 所以$\varepsilon_k \to 0$时, $h_{sij}(x) - \underline{h}_{sij}(x) \to 0$. 类似的由式(3.9)也可证明$\varepsilon_k \to 0$时,

$$\begin{aligned}
0 &\leq \overline{h}_{sij}(x) - h_{sij}(x) \\
&\leq \min\{|u_{sij} - l_{sij}| \cdot |v_{sij} - w_{sij}|, |l_{sij} - u_{sij}| \cdot |w_{sij} - v_{sij}|\}
\end{aligned}$$

$$= |u_{sij} - l_{sij}| \cdot |v_{sij} - w_{sij}|. \tag{3.11}$$

同样有$\overline{h}_{sij}(x) - h_{sij}(x) \to 0$. 证毕.

不失一般性, 对任意的$x \in H^k \subseteq H, i = 1, 2, \cdots, m$, 定义

$$\underline{H}_{1i}(x) = \sum_{j=1}^{r_{1i}} \underline{h}_{1ij}(x) + q_{1i}^{\mathrm{T}} x + p_{1i}, \overline{H}_{2i}(x) = \sum_{j=1}^{r_{2i}} \overline{h}_{2ij}(x) + q_{2i}^{\mathrm{T}} x + p_{2i},$$

$$\overline{\overline{H}}_{2i} = \max_{x \in X \cap H} \overline{H}_{2i}(x), \underline{G}(x) = \sum_{i=1}^{m} \frac{\underline{H}_{1i}(x)}{\overline{\overline{H}}_{2i}}.$$

定理3.2 对任意的$x \in H^k = [\underline{x}^k, \overline{x}^k] \subseteq H, i = 1, 2, \cdots, p$, 以下结论成立:

(i) $H_{1i}(x) \geq \underline{H}_{1i}(x), \overline{H}_{2i}(x) \geq H_{2i}(x), G(x) \geq \underline{G}(x)$;

(ii) 当$\|\overline{x}^k - \underline{x}^k\| \to 0$时,$H_{1i}(x) - \underline{H}_{1i}(x) \to 0, \overline{H}_{2i}(x) - H_{2i}(x) \to 0, G(x) - \underline{G}(x) \to 0$.

证 (i)对任意的$x \in H^k$, 以及$i = 1, 2, \cdots, p$, 根据(3.10)可知, 当$s = 1$ 时,

$$\begin{aligned} H_{1i}(x) - \underline{H}_{1i}(x) &= \sum_{j=1}^{r_{1i}} (\alpha_{1ij}^{\mathrm{T}} x)(\beta_{1ij}^{\mathrm{T}} x) - \sum_{j=1}^{r_{1i}} \underline{h}_{1ij}(x) \\ &= \sum_{j=1}^{r_{1i}} h_{1ij}(x) - \sum_{j=1}^{r_{1i}} \underline{h}_{1ij}(x) = \sum_{j=1}^{r_{1i}} (h_{1ij}(x) - \underline{h}_{1ij}(x)) \\ &\geq 0. \end{aligned}$$

同理, 根据式(3.11)可知, 当$s = 2$时, $\overline{H}_{2i}(x) - H_{2i}(x) = \sum_{j=1}^{r_{1i}} (\overline{h}_{2ij}(x) - h_{2ij}(x)) \geq 0$. 因为$H_{2i}(x) > 0$, 所以$\overline{\overline{H}}_{2i} \geq \overline{H}_{2i}(x) \geq H_{2i}(x) > 0$. 虽然$H_{1i}(x) \geq 0$, 但是存在$x \in H^k$使得$\underline{H}_{2i}(x) < 0$, 以及$\underline{G}(x) < 0$, 此种情况下, 显然 $G(x) > \underline{G}(x)$, 满足结论(i); 当$\underline{H}_{1i}(x) \geq 0$即$\underline{G}(x) \geq 0$ 时, $G(x) = \sum_{i=1}^{m} \frac{H_{1i}(x)}{H_{2i}(x)} \geq \sum_{i=1}^{m} \frac{\underline{H}_{1i}(x)}{H_{2i}(x)} \geq \sum_{i=1}^{m} \frac{\underline{H}_{1i}(x)}{\overline{H}_{2i}(x)} \geq \sum_{i=1}^{m} \frac{\underline{H}_{1i}(x)}{\overline{\overline{H}}_{2i}} = \underline{G}(x)$ 同样满足结论(i).

(ii) 当$\|\overline{x}^k - \underline{x}^k\| \to 0$时, $H_{1i}(x) - \underline{H}_{1i}(x) = \sum_{j=1}^{r_{1i}} (h_{1ij}(x) - \underline{h}_{1ij}(x))$, 结合定理3.1 可知, $H_{1i}(x) - \underline{H}_{1i}(x) \to 0$, 同理, 结合定理3.1我们也可以得到$\overline{H}_{2i}(x) - H_{2i}(x) \to 0$; 由于

$$\begin{aligned} G(x) - \underline{G}(x) &= \sum_{i=1}^{p} [(\frac{H_{1i}(x)}{H_{2i}(x)} - \frac{H_{1i}(x)}{\overline{H}_{2i}(x)}) + (\frac{H_{1i}(x)}{\overline{H}_{2i}(x)} - \frac{H_{1i}(x)}{\overline{\overline{H}}_{2i}}) + (\frac{H_{1i}(x)}{\overline{\overline{H}}_{2i}} - \frac{\underline{H}_{1i}(x)}{\overline{\overline{H}}_{2i}})] \\ &= \sum_{i=1}^{p} [\frac{H_{1i}(x)}{H_{2i}(x)\overline{H}_{2i}(x)} (\overline{H}_{2i}(x) - H_{2i}(x)) + \frac{H_{1i}(x)}{\overline{H}_{2i}(x)\overline{\overline{H}}_{2i}} (\overline{\overline{H}}_{2i} - \overline{H}_{2i}(x)) \\ &\quad + \frac{1}{\overline{\overline{H}}_{2i}} (H_{1i}(x) - \underline{H}_{1i}(x))]. \end{aligned} \tag{3.12}$$

基于上述讨论, 可以知道, 对每一个$i = 1, 2, \cdots, p, H_{1i}(x), H_{2i}(x), \overline{H}_{2i}(x)$都是大于零的并且都是连续的, 所以它们在有界闭集$X$, H以及剖分后的子区域X^k, H^k上都是有界并且是非负的, 那么分别存在相当大的正数N_{1i}, N_{2i}使得$\frac{H_{1i}(x)}{H_{2i}(x)\overline{H}_{2i}(x)} \leq N_{1i}$, $\frac{H_{1i}(x)}{H_{2i}(x)\overline{\overline{H}}_{2i}} \leq N_{2i}$. 记$M_i = \max\{N_{1i}, N_{2i}, \frac{1}{\overline{\overline{H}}_{2i}}\}$, 根据式(3.12)可知

$$G(x) - \underline{G}(x) \leq \sum_{i=1}^{p} M_i [(\overline{H}_{2i}(x) - H_{2i}(x)) + (\overline{\overline{H}}_{2i} - \overline{H}_{2i}(x)) + (H_{1i}(x) - \underline{H}_{1i}(x))].$$

因为当$\|\overline{x}^k - \underline{x}^k\| \to 0$时, $\|\overline{\overline{H}}_{2i} - \overline{H}_{2i}(x)\| \leq [\max_{x \in X \cap H} \overline{H}_{2i}(x) - \min_{x \in X \cap H} \overline{H}_{2i}(x)] \to 0$. 所以, $G(x) - \underline{G}(x) \to 0$. 证毕.

那么, 根据上述讨论, 我们就得到了问题(QFP)的非线性松弛规划子问题(NLRP(H^k)):

$$(\mathrm{NLRP}(H^k)):\begin{cases}\min\ \underline{G}(x)=\sum\limits_{i=1}^{p}\dfrac{\underline{H}_{1i}(x)}{\overline{\overline{H}}_{2i}}\\ \text{s.t. }\ x\in X\cap H^k.\end{cases}$$

显然, (NLRP(H^k))可以转化为(QFP)的一个线性松弛规划子问题(LRP(H^k)):

$$(\mathrm{LRP}(H^k)):\begin{cases}\min\ \underline{G}(x)=\sum\limits_{i=1}^{p}\dfrac{\sum_{j=1}^{r_{1i}}y_{ij}+q_{1i}^{\mathrm{T}}x+p_{1i}}{\overline{\overline{H}}_{2i}}\\ \text{s.t. }\underline{\phi}_{1ij}\le y_{ij},\\ \quad\underline{\varphi}_{1ij}\le y_{ij}, i=1,2,\cdots,p, j=1,2,\cdots,r_{1i},\\ \quad x\in X\cap H^k.\end{cases}$$

综上所述, 我们知道在算法迭代到第k步时, 只要在此步所对应的超矩形H^k上求解问题(LRP(H^k)) 的最优值v(LRP^k), 那么这个最优值同时也是原问题(QFP) 在H^k上全局最优值v(QFP)的一个下界, 即$v(\mathrm{LRP}^k)\le v(\mathrm{QFP})$.

定上界的操作是在分支的过程中不断产生问题(QFP)的可行点来完成的. 通过求解原问题在H^k 上的一个有效下界v(LRP^k), 它的最优解是原问题在H^k上的一个可行点, 而我们通过在分支定界的过程中不断增加满足可行域X 的可行点, 以达到更新上界的目的.

4. 超矩阵的剖分和缩减

在这一节, 我们给出超矩形的分支、缩减步骤对应的算法(RPF)和算法(RSA).

首先, 为了便于本文算法的分支, 这里采用二分法的思想. 令$H^k=[l^k,u^k]\in H$ 表示当前所要剖分的超矩形, 用x^k表示问题(QFP)当前最优解, 显然$x^k\in X\cap H^k$. 假设$x^k\in H^k$, 对超矩形$H^k=[l^k,u^k]$, 我们给出算法(RPF)用来对其剖分:

算法(RPF):

步 1 计算$\omega=\max\{(x_j^k-l_j^k)(u_j^k-x_j^k):j=1,2,\cdots,n\}$,

若 $\omega=0$,

令 $u_\mu^k-l_\mu^k=\max\{u_j^k-l_j^k:j=1,2,\cdots,n\}$,

则 $x_\mu^k=\frac{u_\mu^k+l_\mu^k}{2}$;

否则, 找出第一个$x_j^k\in\arg\max\omega$, 记作$x_\mu^k=x_j^k$.

记$x'=(l_1^k,l_2^k,\cdots,l_{j-1}^k,x_\mu^k,l_{j+1}^k,\cdots,l_n^k)^{\mathrm{T}},x''=(u_1^k,u_2^k,\cdots,u_{j-1}^k,x_\mu^k,u_{j+1}^k,\cdots,u_n^k)^{\mathrm{T}}$.

步 2 通过x'与x''的连线或它们连线所在的平面将超矩形H^k剖分为两个子超矩形$H^{k1}=[l^{k1},u^{k1}]$和$H^{k2}=[l^{k2},u^{k2}]$, 则这两个子超矩形分别为:

$$H^{k1}=\prod_{j=1}^{\mu-1}[l_j^k,u_j^k]\times[l_\mu^k,x_\mu^k]\times\prod_{j=\mu+1}^{n}[l_j^k,u_j^k],$$

$$H^{k2}=\prod_{j=1}^{\mu-1}[l_j^k,u_j^k]\times[x_\mu^k,u_\mu^k]\times\prod_{j=\mu+1}^{n}[l_j^k,u_j^k].$$

其次, 为了提高算法的收敛速度, 这里使用文 [10] 中的超矩形缩减技术. 假设问题(QFP)中的所有线性不等式约束的展开形式为: $\sum_{j=1}^{n}a_{ij}x_j\le b_i, i=1,2,\cdots,m$, 文 [10] 中的超矩形缩减技术可以表述为算法(RSA):

算法(RSA):

令 $I^k:=\{1,2,\cdots,m\}$,

步 1 对每一个 $i = 1, 2, \cdots, m$,

分别计算$rU_i = \sum_{j=1}^{n} \max\{a_{ij}l_j^k, a_{ij}u_j^k\}$和$rL_i = \sum_{j=1}^{n} \min\{a_{ij}l_j^k, a_{ij}u_j^k\}$.

步 2 若 $rL_i > b_i$, 停止. 问题(QFP)在H^k上没有可行解(H^k被删除);

若 $rU_i \leq b_i$, 令 $I^k := I^k - \{i\}$, 问题(QFP)的第i个线性不等式约束被删除;

否则, 对每一个 $j = 1, 2, \cdots, n$

若 $a_{ij} > 0$, 令 $u_j^k = \min\{u_j^k, \frac{b_i - rL_i + \min\{a_{ij}l_j^k, a_{ij}u_j^k\}}{a_{ij}}\}$;

否则 $a_{ij} < 0$, 令 $l_j^k = \max\{l_j^k, \frac{b_i - rU_i + \max\{a_{ij}l_j^k, a_{ij}u_j^k\}}{a_{ij}}\}$.

为了方便起见, 我们把该算法产生的新的超矩形仍然记为H^k, 它是原来超矩形的子集.

5. 新的分支定界算法及其收敛性

最后, 基于上述讨论, 本文求解问题(QFP)全局最优解的分支定界算法总结如下:

步 0 (初始化)

步 0.1 构造包含可行域X的n维超矩形$H^0 = H = [l, u]$; 令分支的所有活跃节点的集合记为$Q_0 = \{H^0\}$, 上界为$U_0 = +\infty$, 将本文上述所有可行点都放入集合W 中, 所需容忍度$\epsilon > 0$.

步 0.2 解线性规划问题(LRP(H^0)), 如果不可行, 那么原问题无解; 否则, 记得到的最优值和最优解分别为$L(H^0), (x^0, y^0)$, 置$L_0 = L(H^0)$; 如果$x^0 \in X$, 令$W = W \cup \{x^0\}, U_0 = G(x^0)$, 若$U_0 - L_0 \leq \epsilon$, 那么称$x^0$为问题(QFP)的$\epsilon$-全局最优解, 迭代终止; 否则, 置$k = 1$, 转入步1.

步 1 (超矩形的分支和缩减)

在Q_k中挑选出当前最小下界所对应的超矩形H^k, 超矩形H^k直接经过剖分算法(RPF)的分解可产生形如第四节中的两个新的子超矩形H^{k1}和H^{k2}; 然后运用算法(RSA)分别对H^{k1}和H^{k2}按以下步骤缩减, 为了方便起见, 我们把缩减后的超矩形记为$H^{ki}, i \in \Gamma$, 其中Γ是缩减后超矩形的指标集合.

步 1.1 运用算法(RSA)对H^{k1}进行缩减.

若 H^{k1}被删除, 则 $Q_k = Q_k \backslash \{H^k\}$, 转到步1.2;

否则, 令 $Q_k = Q_k \backslash \{H^k\} \cup H^{k1}$;

求解线性松弛规划问题(LRP(H^{k1})), 得到最优值记为L_{k1}, 最优解记为(x^{k1}, y^{k1});

若 $x^{k1} \in X$, 让 $W = W \cup x^{k1}$, 转到步1.2;

步 1.2 运用算法(RSA)对H^{k2}进行缩减,

若 H^{k2}被删除,

若 H^{k1}已经被删除, 转到步3;

若 H^{k1}没有被删除, 转到步2;

否则 令 $Q_k = Q_k \cup H^{k2}$;

求解线性松弛规划问题(LRP(H^{k2})), 得到最优值记为L_{k2}, 最优解记为(x^{k2}, y^{k2});

若 $x^{k2} \in X$, 让$W = W \cup x^{k2}$, 转到步2;

步 2 (定界)

如果$W = \varnothing$, 那么$U_k = +\infty$; 如果$W \neq \varnothing$, 那么$U_k = \min\{G(x) : x \in W\}$, 当前的最优解为$x^k \in \arg\min G$;

如果$Q_k = \varnothing$, 那么L_k保持不变; 如果$Q_k \neq \varnothing$, 那么$L_k = \min\{L(H) : H \in Q_k\}$;

步 3 (剪枝规则)

让$Q_{k+1} = Q_k \backslash \{H : U_k - L(H) > \varepsilon, H \in Q_k\}$. 若$Q_k = \varnothing$, 迭代终止:$U_k$是(QFP)的$\epsilon$-全局最优值, x^k是ϵ-全局最优解, 否则, 置$k = k + 1$, 转到步1.

下面为了说明我们算法是收敛的, 给出定理5.1并给予相应的证明.

定理 5.1　设问题(QFP)的可行域X是n维的, 且超矩形的剖分是耗尽的, 那么以下两个结论成立:

(i) 如果算法在迭代过程中有限步终止, 则问题(QFP)的全局最优解将在算法终止时得到;

(ii) 如果算法在有限步不能终止, 则将产生可行解组成的序列$\{x^k\}$并且它的每一个聚点x^*必是问题(QFP)的全局最优解.

证 (i)若算法是有限步终止, 那么根据终止准则可知$U^k - L^k < \epsilon$. 步2和步3暗示了

$$G(x^k) - L^k \le \epsilon. \tag{5.1}$$

假设此时的全局最优解是x^*, 我们知道

$$U^k = G(x^k) \ge G(x^*) \ge L^k. \tag{5.2}$$

因此, 将式(5.1)和(5.2)联立可得

$$G(x^k) + \epsilon \ge G(x^*) + \epsilon \ge L^k + \epsilon \ge G(x^k).$$

故(i)得证.

(ii) 如果算法的迭代是无限的, 则通过求解问题(LRP^k)可产生原问题的可行解序列$\{x^k\}$, 以及对应的序列$\{(x^k, y^k)\}$. 根据步1, 随着可行域的不断加细我们有

$$L^k = \underline{G}(x^k, y^k) \le G(x^U) \le G(x^k) = U^k, k = 1, 2, \cdots. \tag{5.3}$$

因为下界序列$\{L^k = \underline{G}(x^k, y^k)\}$是单调不减且有界的, 且上界序列$\{U^k = G(x^k)\}$是单调不增且有界, 故原问题的上下界都是收敛的. 对式(5.3) 两边取极限可得

$$\lim_{k\to\infty} \underline{G}(x^k, y^k) \le G(x^*) \le \lim_{k\to\infty} G(x^k). \tag{5.4}$$

于是, 令$L = \lim_{k\to\infty} \underline{G}(x^k, y^k), U = \lim_{k\to\infty} G(x^k)$, 那么式(5.4)将转化为

$$L \le G(x^*) \le U. \tag{5.5}$$

不失一般性, 假设算法在迭代过程中形成的超矩形序列$\{H^k = [l^k, u^k]\}$满足$x^k \in H^k$以及$H^{k+1} \in H^k$. 由于算法在每步迭代中剖分超矩形都是沿着最长边来剖分的, 这保证了超矩形尽最大可能的加细, 那么$\lim_{k\to\infty} H^k = x^*$, 在这个过程中也会形成关于 x 的序列 $\{x^k\}$同时满足$\lim_{k\to\infty} x^k = x^*$, 根据函数$G(x)$的连续性, 就会有

$$L \le G(x^*) \le \lim_{k\to\infty} G(x^k) = U.$$

所以, 序列$\{x^k\}$的每一个聚点x^*都是问题(QFP)的全局最优解.

6. 数值实验

在这一节, 我们用几个其他文献中已知的算例来测试文中算法的有效性. 本文所有的线性规划问题均选用单纯形法求解, 算法所有测试过程均用 MATLAB9.0.0.341360(R2016a) 在 Inter(R)Core (TM)i5-3570K, CPU@3.2GHz, 4GB 内存, 64 位 Windows7 操作系统的计算机上运行.

算例 1[11,14,16]:

$$\begin{cases} \max\ \ \dfrac{-x_1^2 + 3x_1 - x_2^2 + 3x_2 + 3.5}{x_1 + 1.0} + \dfrac{x_2}{x_1^2 - 2x_1 + x2^2 - 8x_2 + 20.0} \\ \text{s.t.}\ \ 2x_1 + x_2 \le 6, \\ \quad\ \ 3x_1 + x_2 \le 8, \\ \quad\ \ x_1 - x_2 \le 1, \\ \quad\ \ x_1 \ge 1, x_2 \ge 1. \end{cases}$$

算例 2[11,13]:

$$
\begin{cases}
\max \ \dfrac{3x_1+5x_2+3x_3+50}{3x_1+4x_2+5x_3+50}+\dfrac{3x_1+5x_2+50}{3x_1+5x_2+3x_3+50} \\
\qquad +\dfrac{4x_1+2x_2+4x_3+50}{5x_1+4x_2+3x_3+50} \\
\text{s.t.} \ \ 6x_1+3x_2+3x_3\le 10, \\
\qquad 10x_1+3x_2+8x_3\le 10, \\
\qquad x_1,x_2,x_3\ge 0.
\end{cases}
$$

算例 3[13,14,16]:

$$
\begin{cases}
\max \ \dfrac{-x_1^2+4x_1-2x_2^2+8x_2-3x_3^2+12x_3+56}{x_1^2-2x_1+x_2^2-2x_2+x_3+20} \\
\qquad +\dfrac{-2x_1^2+16x_1-x_2^2+8x_2+2}{2x_1+4x_2+6x_3} \\
\text{s.t.} \ \ x_1+x_2+x_3\le 10, \\
\qquad -x_1-x_2+x_3\le 4, \\
\qquad x_1,x_2,x_3\ge 1.
\end{cases}
$$

算例 4[14,16]:

$$
\begin{cases}
\max \ \dfrac{-x_1^2+16x_1-x_2^2+16x_2-x_3^2+16x_3-x_4^2+16x_4-214}{2x_1-x_2-x_3+x_4+2} \\
\qquad +\dfrac{-x_1^2+16x_1-2x_2^2+20x_2-3x_3^2+60x_3-4x_4^2+56x_4-586}{-x_1+x_2+x_3-x_4+10} \\
\qquad +\dfrac{-x_1^2+20x_1-x_2^2+20x_2-x_3^2+20x_3-x_4^2+20x_4-324}{x_1^2-4x_4} \\
\text{s.t.} \ \ 6\le x_1\le 10, \\
\qquad 4\le x_2\le 6, \\
\qquad 8\le x_3\le 12, \\
\qquad 6\le x_4\le 8, \\
\qquad x_1+x_2+x_3+x_4\le 34.
\end{cases}
$$

算例 5[11,17]:

$$
\begin{cases}
\max \ \dfrac{-x_1^2+3x_1-x_2^2+3x_2+3.5}{x_1+1.0}+\dfrac{x_2}{x_1^2-2x_1+x2^2-8x_2+20.0} \\
\text{s.t.} \ \ 2x_1+x_2\le 6, \\
\qquad 3x_1+x_2\le 8, \\
\qquad x_1-x_2\le 1, \\
\qquad x_1\ge 1, x_2\ge 2.
\end{cases}
$$

算例1-5的计算结果我们已呈现在表6.1中, 算法运行过程中的容忍度分别为$10^{-3}, 10^{-5}$, 10^{-2},10^{-2},10^{-6}, 显然从表6.1中我们知道这些容忍度分别为其他各个文献中所用容忍度的最大值, 足以与其他算法对比. 通过表6.1可以看出, 本文算法总体来说比文[11,13,17]计算效果要好, 与文[14]相比, 我们算法仅在算例3表现的更好, 其他算例表现略差, 虽然总体计算能力低于文[16]中的算法, 但是我们所提出的算法是有效可行的.

为了进一步说明本文算法的性能, 对于算例6, 我们也进行了若干的伪随机试验并呈现在了表6.2中, 下文也做出了对应的说明.

表 6.1　算例 1-5 利用本文算法产生的数值结果

E	R	x^*	G(x^*)	I	ϵ
1	本文	(1.00000000014839,1.7999999928353)	4.06080426335836	113	10^{-3}
1	[11]	(1.000000000,1.768310547)	4.060807586	224	10^{-3}
1	[14]	(1.000,1.7436)	4.0608	60	-
1	[16]	(1.00,1.74)	4.06	17	10^{-3}
2	本文	(0.00000,0.00000,0.00000)	3.00000	1	10^{-5}
2	[11]	(0.00000,0.00000,0.00000)	3.00000	1	10^{-5}
2	[13]	(0.00000,0.00000,0.00000)	3.00000	1	10^{-5}
3	本文	(1.82031250,1.00000000,1.000000006537)	6.1198	49	10^{-2}
3	[13]	(1.857910156,1.044433594,1.0)	6.119830507	1001	10^{-2}
3	[14]	(1.8188,1.000,1.000)	6.1198	90	-
3	[16]	(1.81,1.00,1.00)	6.12	24	10^{-2}
4	本文	(6.000,6.000,9.991,8.000)	16.1658	148	10^{-2}
4	[14]	(6.000,6.000,10.0604,8.000)	16.1686	59	-
4	[16]	(6.00,6.00,10.06,8.00)	16.17	37	10^{-2}
5	本文	(1.000000000,2.000000000)	4.035714286	75	10^{-6}
5	[11]	(1.000000000,2.000038147)	4.035714101	137	10^{-6}
5	[17]	(1.000338508,2.001015610)	4.035714286	1205	10^{-3}

表 6.2　算例 6 利用本文算法产生的数值结果

p	m	n	r	I	t
5	5	5	5	48.55	206.8727
10	5	5	5	48.64	411.5947
20	5	5	5	46.79	792.4378
30	5	5	5	74.90	3191.6535
50	5	5	5	196.28	9627.3659
50	10	5	3	332.50	5148.2764
50	10	5	5	494.64	13028.6566
50	30	5	3	153.47	2871.4493
50	30	5	5	180.54	5426.6331
70	40	5	3	152.23	4575.7491
70	40	5	5	184.19	8903.5991
90	50	5	3	160.32	6514.2262
90	50	5	5	168.75	11239.4278
5	100	40	3	107.37	1615.5653
5	100	40	5	91.81	4039.0127
5	100	50	5	123.21	4849.6315
5	100	60	5	129.09	6116.9021
5	100	70	5	147.13	6140.2802
5	100	100	50	197.47	156790.1423
5	100	150	50	283.12	243970.7918
5	100	200	100	423.84	387389.3067

算例 6:

$$
\text{(EQFP)}:\begin{cases}\min\ G(x)=\displaystyle\sum_{i=1}^{p}\frac{\sum_{j=1}^{r}(\alpha_{1ij}^{\mathrm{T}}x)(\beta_{1ij}^{\mathrm{T}}x)+q_{1i}^{\mathrm{T}}x+p_{1i}}{\sum_{j=1}^{r}(\alpha_{2ij}^{\mathrm{T}}x)(\beta_{2ij}^{\mathrm{T}}x)+q_{2i}^{\mathrm{T}}x+p_{2i}}\\ \text{s.t.}\ \ x\in X=\{x\in\mathbb{R}^n|Ax\le b\},\end{cases}
$$

这里,$A, b, \alpha_{sij}, \beta_{sij}, q_{si}, p_{si}, s = 1, 2, i = 1, 2, \cdots, p, j = 1, 2, \cdots, r$,的每个元素都是在$[0, 1]$上伪随机产生的, r表示矩阵的秩.

表6.1和表6.2中, 表头的符号分别是: E: 算例编号; R: 计算方法; x^*: 原问题的最优解; $G(x^*)$: 目标函数的最优值; I: 平均迭代次数; ϵ: 容忍度; t: 迭代终止时CPU平均运行时间; p: 目标函数中二次比式的个数; r: 目标函数分子和分母中矩阵的秩, 这里假设它们秩都相等; m: 线性约束的个数; n: 目标函数的维数.

通过表6.2的计算结果, 可以知道当m, r和n固定不变时, 随着p的增加, 算法的平均迭代次数I在缓慢增加, 但是迭代时间t增加的很快; 当p, r和n固定不变时, 约束条件增多时, I和t都是快速减少的; 当p和m保持不变时, 随着维数n的增加, I和t都是急剧增加的, 此时若将矩阵的秩变小, 那么相应的I和t也在减少; 当约束条件的个数m相对于决策变量个数n非常大时, 因为这样形成的可行域非常狭小, 所以能够在很少的迭代步骤找到最优解, 但是每一步的迭代时间都很长; 当决策变量个数n相对于约束条件的个数m很大时, 形成的可行域范围很大, 迭代时间和步骤都变大.

参考文献:

[1] WANG C F, BAI Y Q, SHEN P P. A practicable branch-and-bound algorithm for globally solving linear multiplicative programming[J]. Optimization, 2017, 66(3):1-9.

[2] GAO Y, WEI F. A new bound-and-reduce approach of nonconvex quadratic programming problem-s[J]. Applied Mathematics & Computation, 2015, 250(250):298-308.

[3] KONNO H. Parametric simplex algorithms for solving a special class of nonconvex minimization problems[J]. Journal of Global Optimization, 1991, 1(1):65-81.

[4] KONNO H, ABE N. Minimization of the sum of three linear fractional functions[J]. Journal of Global Optimization, 1999, 15(4):419-432.

[5] SHEN P P, WANG C F. Global optimization for sum of linear ratios problem with coefficients[J]. Applied Mathematics & Computation, 2006, 176(1):219-229.

[6] WANG Y, SHEN P, LIANG Z. A branch-and-bound algorithm to globally solve the sum of several linear ratios [J]. Applied Mathematics & Computation, 2005, 168(1):89-101.

[7] BENSON H P. Solving sum of ratios fractional programs via concave minimization[J]. Journal of Optimization Theory & Applications, 2007, 135(1):1-17.

[8] JIAO H W,FENG Q G. Global optimization for sum of linear ratios problem using new pruning technique[J]. Mathematical Problems in Engineering,2008,2008(4):267-290.

[9] WANG C F, SHEN P P. A global optimization algorithm for linear fractional programming[J]. Applied Mathematics & Computation, 2008, 204(1):281-287.

[10] ZHANG Shuguang, CHEN Y, JIA L.A global algorithm for minimizing generalized quadratic fractional programs with nonconvex quadratic constraints[J]. Henan Science, 2010(6): 638-641.

[11] JIAO H W, LIU S. Range division and compression algorithm for quadratically constrained sum of quadratic ratios[J]. Computational & Applied Mathematics, 2015, 36(1):225-247.

[12] ZHAO Y, LIU S. Global optimization algorithm for mixed integer quadratically constrained quadratic program[J]. Journal of Computational and Applied Mathematics, 2017, 319: 159-169.

[13] JIAO H, LIU S. An efficient algorithm for quadratic sum-of-ratios fractional programs problem[J]. Numerical Functional Analysis and Optimization, 2017, 38(11): 1426-1445.

[14] GAO L, MISHRA S K, SHI J. An extension of branch-and-bound algorithm for solving sum-of-nonlinear-ratios problem[J]. Optimization Letters, 2012, 6(2):221-230.

[15] 汪春峰, 刘三阳. 基于单纯形剖分确定非线性比式和问题全局解的新方法[J]. 系统工程理论与实践, 2013, 33(3):742-747.

[16] BENSON H P. Using concave envelopes to globally solve the nonlinear sum of ratios problem[J]. Journal of Global Optimization, 2002, 22(1-4):343-364.

[17] JIAO H W, WANG Z, CHEN Y. Global optimization algorithm for sum of generalized polynomial ratios problem[J]. Applied Mathematical Modelling, 2013, 37(1-2):187-197.

A New Deterministic Algorithm for Solving the Global Solution of the Generalized Quadratic Ratio Problem

ZHANG Bo[1,2], *GAO Yuelin*[1,2]

(*1. School of Mathematics and Information Science, North Minzu University, Yinchuan 750021, China; 2. Ningxia Scientific Computing and Intelligent Information Processing Co-innovation Center, Yinchuan 750021, China*)

Abstract: The purpose of this paper is to solve a class of sum-of-quadratic-ratios problem. Firstly, we transform the Original problem into a nonlinear programming problem, in which the form of the molecular and denominator of the objective function are composed of the sum of the product of the two linear functions and adding a linear function. Then, according to the characteristic of the programming problem of the product of two linear functions, the objective function is loosened linearly to determine the lower bound of the optimal value of the original problem. A branch and bound algorithm for solving linear programming problems is proposed and the convergence of the algorithm is proved. Finally, the numerical results show that the proposed algorithm is feasible and effective.

Key words: Global optimization; Fractional programming; Quadratic function; Linear multiplicative programming; Branch and bound; Linear programming