加速梯度算法在池化问题上的应用

献给越民义教授 100 华诞

程秋韵¹, 白廷琴^{1*}, 李倩², 余长君¹

1. 上海大学理学院数学系, 上海 200444;

2. 上海工程技术大学数理与统计学院, 上海 201620

E-mail: autumn05_cqy@163.com, yqbai@t.shu.edu.cn, liqian15123329166@163.com, yuchangjun@126.com

收稿日期: 2020-01-06; 接受日期: 2020-06-11; 网络出版日期: 2020-09-09; * 通信作者

国家自然科学基金 (批准号: 11771275) 资助项目

摘要 池化问题是石油生产计划中的重要问题之一. 它是一类非凸的网络流优化问题, 且模型中对物流性质的约束使得该问题是强 NP- 难的, 因此, 如何解决这类非凸优化问题成了目前研究的重点. 另一方面, 随着工业发展, 问题规模的不断增大也给池化问题的求解带来新的挑战. 本文利用向量化的技巧对原有池化问题模型进行了等价转换和改进, 使其表示为一个改进的 P- 形式. 通过分析 P- 形式模型的特殊结构, 本文设计了加速梯度算法, 并证明了算法的收敛性. 数值实验的结果验证了模型的优越性和算法的有效性.

关键词 池化问题 改进的 P- 形式 改进的加速梯度算法

MSC (2010) 主题分类 90C26

1 引言

池化问题 (pooling problem), 是指在炼油和石油化工生产中所遇到的物流混合问题, 通常也称为混流问题 (blending problem). 这里的“池化 (pooling)”, 或者说“混流 (blending)”, 通常发生在存储罐中混合的过程中, 即将不同属性的原油, 或者不同性质的成分, 加入到同一罐中进行处理再调和, 以达到生产所需产品的目的. 因此, 如何在极小化成本的目标下正确地计算混合物流的性质并用于生产和经营是十分重要的问题.

池化问题由 Haverly^[1] 第一次提出. 此后, Haverly^[2] 证明了池化问题的计算难度高是由于问题存在多个局部最优解. 该模型主要的求解方法是递归算法, 但其缺点是算法本身的不收敛性. 因此在此基础上, 关于池化问题的研究逐渐集中在两个方面. 一方面是随着物流性质和问题规模的变化, 如何构造池化问题的数学模型. 其主要包括 3 种主流形式: 最早由 Haverly 提出的 P- 形式, Ben-Tal^[3] 基于混流的成分所占百分比提出的 Q- 形式, 以及 Quesada 和 Grossmann^[4] 提出的用额外的约束条件

英文引用格式: Cheng Q Y, Bai Y Q, Li Q, et al. Applications of the accelerated gradient algorithm for pooling problem (in Chinese). Sci Sin Math, 2020, 50: 1133–1148, doi: 10.1360/SSM-2020-0005

将 P- 形式和 Q- 形式结合起来的 PQ- 形式. 另一方面则是如何设计更有效的算法求解池化问题. 一些学者如 Frimannslund 和 Haugland^[5] 开始使用逐次线性规划 (successive linear programming, SLP) 算法求解池化问题. 这一算法的优势在于计算速度非常快, 且适用于大规模问题的解决. 但它的缺点在于不能保证得到 KKT (Karush-Kuhn-Tucker) 点甚至是可行点. 近年来, Dai 等^[6] 利用非单调技巧改进了已有的逐次线性规划算法对问题进行求解. 此外, 求解池化问题的全局优化算法主要有 Foulds 等^[7-9]、Audet 等^[8]、Alfaki 和 Haugland^[9] 的分支定界法, Adhya 等^[10] 的 Lagrange 松弛法, 以及 Dey 和 Gupte 的混合整数规划 (mixed integer linear programming, MILP) 技术 (参见文献 [11]) 等. 这类算法的优点是可以求得池化问题的全局最优解, 但缺点是计算速度较慢.

本文首先通过引入辅助变量, 利用向量化的技巧将原池化问题模型等价地转换为一个向量化模型. 并通过构造该向量化模型的罚形式, 将池化问题的标准形式转换成一个非凸无约束优化问题, 即它的目标函数是一个非凸光滑项和一个非光滑凸项的和. 接着, 引入改进的加速梯度算法对问题进行求解, 通过挖掘模型的特殊结构, 对算法进行了收敛性分析. 最后, 数值实验结果验证了模型的优越性和算法的有效性.

下面介绍本文余下内容的结构. 第 2 节回顾池化问题的基本概念, 主要包含了相关的记号、定义和定理, 同时给出池化问题的标准模型. 第 3 节引入新的参数, 经过转换后得到了原问题的改进的 P- 形式. 第 4 节采用改进的加速梯度算法对改进的 P- 形式进行求解, 并证明算法的收敛性. 数值实验的结果展示在第 5 节中, 验证了算法和模型的优越性. 最后第 6 节对全文进行总结.

2 基本概念

本节介绍关于池化问题的一些基本记号、定义和定理.

2.1 基本记号和定理

$\|\cdot\|$ 表示 Euclid 范数, $\|\cdot\|_1$ 表示 L_1 范数, $\|\cdot\|_\infty$ 表示 L_∞ 范数, A^T 表示矩阵 $A \in \mathbb{R}^{m \times n}$ 的转置.

定义 2.1^[12] 一个广义的实值函数 $h: \mathbb{R}^n \rightarrow [-\infty, +\infty]$ 的域定义为 $\text{dom } h = \{x \in \mathbb{R}^n : h(x) < +\infty\}$, 若满足 $h \neq -\infty$, 且 $\text{dom } h \neq \emptyset$, 则称 h 是真值的. 若它还是下半连续的, 则认为这样一个函数是闭的.

当一个真闭函数 h 的下半水平集是有界的, 称 h 是水平有界的, 即对任意 $r \in \mathbb{R}$, 集合 $\{x \in \mathbb{R}^n : h(x) \leq r\}$ 是有界的.

定义 2.2^[12] 对真闭凸函数 $h: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$, 在 $x \in \text{dom } h$ 处 h 的次梯度为

$$\partial h(x) = \{\xi \in \mathbb{R}^n : h(u) - h(x) - \langle \xi, u - x \rangle \geq 0, \forall u \in \mathbb{R}^n\}.$$

定义 2.3^[13] 定义一个函数 $f \in \mathcal{C}_{L_f}^{1,1}$, 即 f 有 Lipschitz 连续梯度的连续可微函数:

$$\|\nabla f(x) - \nabla f(y)\| \leq L_f \|x - y\|, \quad \forall x, y \in \mathbb{R}^n,$$

这里 L_f 是 ∇f 的 Lipschitz 常数.

定义 2.4^[14] 按如下形式将空间 $\mathbb{R}^N$ 分解成 n 个子空间: 令 $U \in \mathbb{R}^{N \times N}$ 是一个关于 $N \times N$ 单位阵的列置换, 且令 $U = [U_1, U_2, \dots, U_n]$, 即矩阵 U 可以被分解成 n 个子矩阵的形式, 于是, U_i 的大小为 $N \times N_i$, 且 $\sum_i N_i = N$.

显然, 任意向量 $x \in \mathbb{R}^N$ 都可以被唯一写成 $x = \sum_i U_i x^{(i)}$, 这里 $x^{(i)} = U_i^T x \in \mathbb{R}_i \equiv \mathbb{R}^{N_i}$. 因此,

$$U_i^T U_j = \begin{cases} N_i \times N_i \text{ 单位阵,} & \text{若 } i = j, \\ N_i \times N_j \text{ 零矩阵,} & \text{其他.} \end{cases}$$

为了简化, 在后文中记 $x = (x^{(1)}, \dots, x^{(n)})^T$.

2.2 池化问题的模型

本小节将详细介绍池化问题, 并给出具体模型. 首先, 给出池化问题的定义.

定义 2.5 (池化问题^[15]) 若一有向无圈图, 其成分和性质已知, 则池化问题就是一类在满足需求与供给的基础上分配每一条边上的流量, 同时满足中间点和输出点的成分界限约束以达到总成本最小的目标的非凸优化问题, 它的其他约束同最小费用流问题一致.

池化问题的一个简单例子如图 1 所示.

其次, 给出池化问题的具体模型. 考虑如下类有向无圈图 $\mathcal{G} = (N, A)$, 其中 N 是图中所有点的集合, A 是图中所有有向边的集合. 为了表述方便, 这里 N 可以根据点的类型被分成 3 个子集 $S, I, T \subset N$, 并且有 $N = S \cup I \cup T$, 其中 S 表示的是输入集, T 表示的是输出集, 而 I 表示的是中间点集或者说池的集合. 一般地, 如果 $A \cap (I \times I) = \emptyset$, 即 $A \subseteq (S \times I) \cup (I \times T) \cup (S \times T)$, 也就是说 $\mathcal{G}$ 是一个三部图, 则这个问题被称为标准池化问题; 若不满足上述的条件, 则剩下的问题被称为一般池化问题. 本文的主要目标是提出一个针对各类问题的通用解决方法.

进一步地, 针对流的方向和成分性质, 引入如下记号: 对于任意点 $i \in N$, $N_i^+ = \{j \in N : (i, j) \in A\}$ 表示点 i 指向的所有点的集合; 同样地, $N_i^- = \{j \in N : (j, i) \in A\}$ 表示指向点 i 的所有点的集合. 类似地, 关于 S 、 I 和 T , 同样有 $N_s^- = \emptyset, \forall s \in S$; $N_t^+ = \emptyset, \forall t \in T$; 且有 $I = N \setminus (S \cup T)$. 有关流的成分, 用 K 来表示流的所有成分的集合. 因此定义两种类型的常数 q_s^k 和 $\bar{q}_t^k$, 分别表示在输入集点 $s \in S$ 处成分 $k \in K$ 的值和在输出集点 $t \in T$ 处成分 $k \in K$ 的上界值. 此外, 我们认为在中间点处成分的值是未知的, 因此定义变量 w_i^k , 表示中间集点 $i \in I$ 处成分 $k \in K$ 的值.

下面讨论池化问题模型的决策变量、目标函数和约束条件.

- 决策变量. 混流的大小及混流的性质 r 和 w .
- 目标函数. 总成本最小, 即单位成本与流量的乘积的和 $\sum_{(i,j) \in A} c_{ij} r_{ij}$, 这里 c_{ij} 和 r_{ij} 分别表示每条边 (i, j) 的成本和流量.

图 1 池化问题的一个简单例子

- 容量约束. 经过每一个节点的流的流量应不超过该点的容量大小, 即

$$\sum_{j \in N_i^+} r_{ij} \leq b_i, \quad \forall i \in N \setminus T, \quad \sum_{j \in N_t^-} r_{jt} \leq b_t, \quad \forall t \in T.$$

- 质量平衡约束. 流向该节点的流量应等于从该节点流出的流量, 即

$$\sum_{j \in N_i^-} r_{ji} - \sum_{j \in N_i^+} r_{ij} = 0, \quad \forall i \in I.$$

• 物流性质约束. 一方面, 每个中间节点的功能是将所有进入的正向流进行接收并混合, 最后将生成的混合物运送进其他的中间节点, 或者直接运送至输出点; 另一方面, 输出点将所有进入的正向流进行接收并混合并由此生成最终的产品. 注意到混合过程一定是遵循线性混合的, 这就意味着一个点的流量总和一定是所有集中到这个点的产品和每个输入到这个点的流量的总和. 所以, 针对中间点集, 节点需满足物流性质平衡的约束, 即

$$\sum_{s \in N_i^- \cap S} q_s^k r_{si} + \sum_{j \in N_i^- \cap I} w_j^k r_{ji} - w_i^k \sum_{j \in N_i^-} r_{ji} = 0, \quad \forall i \in I, \quad \forall k \in K;$$

针对输出集, 节点需满足物流性质界限的约束, 即

$$\sum_{s \in N_i^- \cap S} q_s^k r_{st} + \sum_{j \in N_i^- \cap I} w_j^k r_{jt} - \bar{q}_t^k \sum_{j \in N_t^-} r_{jt} \leq 0, \quad \forall t \in T, \quad \forall k \in K.$$

- 流的界限约束. 每条边 $(i, j) \in A$ 上的流 r_{ij} 一定是非负的, 即 $r_{ij} \geq 0, \forall (i, j) \in A$.

综上所述, 池化问题的 P- 形式如下所示:

$$\begin{aligned} \text{(P)} \quad & \min_{f, w} \sum_{(i, j) \in A} c_{ij} r_{ij} \\ \text{s.t.} \quad & \sum_{j \in N_i^+} r_{ij} \leq b_i, \quad \forall i \in N \setminus T, \\ & \sum_{j \in N_t^-} r_{jt} \leq b_t, \quad \forall t \in T, \\ & \sum_{j \in N_i^-} r_{ji} - \sum_{j \in N_i^+} r_{ij} = 0, \quad \forall i \in I, \\ & \sum_{s \in N_i^- \cap S} q_s^k r_{si} + \sum_{j \in N_i^- \cap I} w_j^k r_{ji} - w_i^k \sum_{j \in N_i^-} r_{ji} = 0, \quad \forall i \in I, \quad \forall k \in K, \\ & \sum_{s \in N_i^- \cap S} q_s^k r_{st} + \sum_{j \in N_i^- \cap I} w_j^k r_{jt} - \bar{q}_t^k \sum_{j \in N_t^-} r_{jt} \leq 0, \quad \forall t \in T, \quad \forall k \in K, \\ & r_{ij} \geq 0, \quad \forall (i, j) \in A. \end{aligned}$$

以上即为标准池化问题和一般池化问题 (即可以被视作类似网络流模型) 的一般模型. 该模型的记号同 Dai 等^[6] 的相同. 可以看到, 模型具有以下特点: 它的目标是线性的, 约束中容量约束、质量平衡约束以及流的界限约束是线性的, 而流的性质约束是非线性, 或者称为双线性的. 正是由于非线性约束, Alfaki 和 Haugland^[16] 已经证明了池化问题是强 NP- 难的, Dai 等^[6] 则证明了即使仅存在一个

物流性质, 池化问题依然是强 NP- 难的. 另一方面, 池化问题在实际生产中有着非常重要的应用, 而现实的问题规模很大, 这就导致对池化问题的求解速度要求较高. 因而, 我们需要寻找一种快速的算法求解池化问题.

在通常情况下, 针对非线性约束的一种常见处理方法就是把它的非线性约束罚到目标函数中从而构造原问题的罚形式, 再进行求解. 但在得到原问题的罚形式后, 我们发现它的目标函数也是非凸的, 这也就启发我们去寻找一种新的方式来处理目标函数. 我们从一种快速的算法—加速梯度算法中获得灵感, 即先将池化问题的模型向量化, 再利用罚形式得到一个改进的 P- 形式, 从而得到一个无约束非凸优化问题, 其中目标函数为两个特殊函数的和. 我们将在接下来的章节中具体讨论.

3 改进的 P- 形式

本节首先通过引入新的记号和定义, 从而得到关于池化问题的标准模型的一个向量化模型. 利用模型的罚形式和等价转换技巧得到最终的改进的 P- 形式. 改进的 P- 形式是一个无约束非凸优化问题, 它的目标函数由两部分组成, 即一个非凸光滑项函数和一个非光滑凸项函数. 进一步地, 我们具体分析了两个函数的特点, 设计了改进的加速梯度算法有效地求解模型, 具体的介绍在第 4 节中.

3.1 新的参数和记号

出于以上考虑, 在沿用上述记号的同时, 引入了一些新的定义和记号.

首先, 有向图 $\mathcal{G} = (N, A)$ 中输入集 S 、中间集 I 和输出集 T 是已知的, 记各集合的大小分别为 $|S| = n_1$, $|I| = n_2$, $|T| = n_3$. 另一方面, 记点集 N 大小为 $|N| = n$, 即 $n = n_1 + n_2 + n_3$. 同样地, 对边集 A 大小记为 $m = |A|$, 而流的性质集合 K , 大小为 $|K|$.

其次, 引入了如下新的记号.

x 为向量变量, $x = (x_i) \in \mathbb{R}^{n^2}$. 先定义矩阵 $r = (r_{ij}) \in \mathbb{R}^{n \times n}$, 其中每条边 r_{ij} ($i \in N, j \in N$) 定义同上, 且表示有向图中每个点到其他点的关系, 再按列拉长得得到一个向量. 同时可以得到边 r_{ij} 与向量 x 中各元素下标的关联. 该变量对应的是原池化问题中混流的大小.

注 3.1 存在这样一种对应关系: 若已知 r_{ij} , 那么一定有 $k = (i-1) \times n + j$, 即 $r_{ij} = x_k$; 反过来, 若已知 x_k , 那么一定有 $i = \lceil \frac{k}{n} \rceil$ 和 $j = k - (i-1) \times n$, 即 $x_k = r_{ij}$.

w^k 为向量变量, $w^k = (w_i^k)^T \in \mathbb{R}^n$, 且元素 w_i^k 表示对性质 $k \in K$ 点 i 上的值. 为了简化书写, 用 w^{kT} 来表示列向量 w^k 的转置. 该变量对应的是原池化问题中混流的性质变量.

q^k 为常数向量, $q^k = (q_i^k)^T \in \mathbb{R}^n$, 且对每个性质 $k \in K$, 有

$$q_i^k = \begin{cases} q_s^k, & 1 \leq i \leq n_1, \\ 0, & n_1 + 1 \leq i \leq n_1 + n_2, \\ \bar{q}_t^k, & n_1 + n_2 + 1 \leq i \leq n. \end{cases}$$

该常量对应的是原池化问题中混流的性质常量.

c 为常数向量, $c = c_1 - c_2 = (c_1(k) - c_2(k)) \in \mathbb{R}^{n^2}$, 其中

$$c_1(k) = \begin{cases} c_{ij}, & \left\lceil \frac{k}{n} \right\rceil = i, \quad k - \left\lfloor \frac{k}{n} \right\rfloor \times n = j, \quad i \in S \text{ 且 } j \in N, \\ 0, & \text{否则,} \end{cases}$$

$$c_2(k) = \begin{cases} c_{ij}, & \left\lceil \frac{k}{n} \right\rceil = i, \quad k - \left\lfloor \frac{k}{n} \right\rfloor \times n = j, \quad i \in T \quad \text{且} \quad j \in N, \\ 0, & \text{否则.} \end{cases}$$

该常量对应的是原池化问题中混流在输入点和输出点的成本大小.

下面各常数向量的引入则是为说明原池化问题的约束条件, 具体如下.

p_i 为常数向量, $p_i = (p_i(k)) \in \mathbb{R}^{n^2}$ ($i = 1, \dots, n$) 表示所有与点 i 相关的流量情况, 即当 $i = 1, \dots, n - n_3$, 即 $i \in N \setminus T$ 时, p_i 表示从点 i 发出的所有流的流量总和, 这样对 $k = 1, \dots, n^2$, 每个元素有

$$p_i(k) = \begin{cases} 1, & \left\lceil \frac{k}{n} \right\rceil = i \text{ 且有边存在,} \\ 0, & \text{否则;} \end{cases}$$

而当 $i = n - n_3 + 1, \dots, n$, 即 $i \in T$ 时, p_i 表示指向点 i 的所有流的流量总和, 这样对 $k = 1, \dots, n^2$, 每个元素有

$$p_i(k) = \begin{cases} 1, & k - \left\lfloor \frac{k}{n} \right\rfloor \cdot n = i \text{ 且有边存在,} \\ 0, & \text{否则.} \end{cases}$$

a_i 为常数向量, $a_i = (a_i(k)) \in \mathbb{R}^{n^2}$ 表示的是从节点 i 发出的流量总和, 即当 $i = n - n_1 + 1, \dots, n - n_3$, 等价于 $i \in I$ 时,

$$a_i(k) = \begin{cases} 1, & \left\lceil \frac{k}{n} \right\rceil = i \text{ 且有边存在,} \\ 0, & \text{否则.} \end{cases}$$

对剩下的 i , 则有 $a_i = 0$.

d_i 为常数向量, $d_i = (d_i(k)) \in \mathbb{R}^{n^2}$ 表示指向节点 i 的流量总和, 即当 $i = n - n_1 + 1, \dots, n - n_3$, 等价于 $i \in I$ 时,

$$d_i(k) = \begin{cases} 1, & k - \left\lfloor \frac{k}{n} \right\rfloor \times n = i \text{ 且有边存在,} \\ 0, & \text{否则.} \end{cases}$$

对剩下的 i , 则有 $d_i = 0$.

H_{1i} 为常数矩阵, $H_{1i} = (H_{1i}(j, k)) \in \mathbb{R}^{n \times n^2}$ ($i = 1, \dots, n$), 这里每一行表示对应的点, 每一列表示对应的边. 所以, 它的作用是计算对输入集 S 中的每个点指向全部点 i 的流的总和, 也即对每个固定的 i ,

$$H_{1i}(j, k) = \begin{cases} 1, & j = \left\lceil \frac{k}{n} \right\rceil, \quad k - \left\lfloor \frac{k}{n} \right\rfloor \times n = i, \quad \left(\left\lfloor \frac{k}{n} \right\rfloor, k - \left\lfloor \frac{k}{n} \right\rfloor \times n \right) \in A \quad \text{且} \quad \left\lfloor \frac{k}{n} \right\rfloor \leq n_1, \\ 0, & \text{否则.} \end{cases}$$

H_{2i} 为常数矩阵, $H_{2i} = (H_{2i}(j, k)) \in \mathbb{R}^{n \times n^2}$ ($i = 1, \dots, n$), 它的定义同 H_{1i} 类似, 但它的作用是计算对中间集 I 中的每个点指向全部点 i 的流的总和, 也即对每个固定的 i ,

$$H_{2i}(j, k) = \begin{cases} 1, & j = \left\lceil \frac{k}{n} \right\rceil, \quad k - \left\lfloor \frac{k}{n} \right\rfloor \times n = i, \quad \left(\left\lfloor \frac{k}{n} \right\rfloor, k - \left\lfloor \frac{k}{n} \right\rfloor \times n \right) \in A \quad \text{且} \quad n_1 + 1 \leq \left\lfloor \frac{k}{n} \right\rfloor \leq n_1 + n_2, \\ 0, & \text{否则.} \end{cases}$$

H_{3i} 为常数矩阵, $H_{3i} = e_i p_i^T \in \mathbb{R}^{n \times n^2}$ ($i = 1, \dots, n$), 这里 e_i 是一个单位向量, 而 p_i 同之前的定义一样. H_{3i} 的作用是先挑出对应的元素, 即当 $i = n - n_1 + 1, \dots, n - n_3$ 时从中选择的是 w^k 的相关元素, 而当 $i = n - n_3 + 1, \dots, n$ 时从中选择的是 q^k 的相关元素, 最终得到的是指向节点 i 的流量的总和.

3.2 向量化模型

基于以上定义和符号, 得到池化问题标准模型的向量化形式. 对决策变量 $x, w^1, \dots, w^k$, 有目标函数: $c^T x$. 且必须满足以下约束条件:

- 容量约束: $p_i^T x \leq b_i, i = 1, \dots, n$;
- 质量平衡约束: $a_i^T x - d_i^T x = 0, i = 1, \dots, n$;
- 中间节点物流性质约束: $q^{kT} H_{1i} x + w^{kT} H_{2i} x - w^k H_{3i} x = 0, i = n - n_1 + 1, \dots, n - n_3$;
- 输出集节点物流性质约束: $q^{kT} H_{1i} x + w^{kT} H_{2i} x - q^k H_{3i} x \leq 0, i = n - n_3 + 1, \dots, n$;
- 流的界限约束: $x \geq 0$.

为了将上述有关混流性质的约束表示成统一的形式, 这里引入两个矩阵 Q_{1i} 和 Q_{2i} , 使得对每个 $k \in K$ 有

$$Q_{1i} = \begin{cases} 0, & i = 1, \dots, n_1, \\ H_{1i}, & i = n_1 + 1, \dots, n - n_3, \\ H_{1i} - H_{3i}, & i = n - n_3 + 1, \dots, n, \end{cases}$$

$$Q_{2i} = \begin{cases} 0, & i = 1, \dots, n_1, \\ H_{2i} - H_{3i}, & i = n_1 + 1, \dots, n - n_3, \\ H_{2i}, & i = n - n_3 + 1, \dots, n. \end{cases}$$

这样可以得到该约束的最终形式, 即

$$\begin{cases} (q^k)^T Q_{1i} x + (w^k)^T Q_{2i} x = 0, & i = 1, \dots, n - n_3, \quad k \in K, \\ (q^k)^T Q_{1i} x + (w^k)^T Q_{2i} x \leq 0, & i = n - n_3 + 1, \dots, n, \quad k \in K. \end{cases}$$

因此, 池化问题的向量化模型 (P1) 可以表述如下:

$$\begin{aligned} \text{(P1)} \quad & \min_{x, w} c^T x \\ \text{s.t.} \quad & p_i^T x \leq b_i, \quad i = 1, \dots, n, \\ & a_i^T x - d_i^T x = 0, \quad i = 1, \dots, n, \\ & (q^k)^T Q_{1i} x + (w^k)^T Q_{2i} x = 0, \quad i = 1, \dots, n - n_3, \quad k = 1, \dots, |K|, \\ & (q^k)^T Q_{1i} x + (w^k)^T Q_{2i} x \leq 0, \quad i = n - n_3 + 1, \dots, n, \quad k = 1, \dots, |K|, \\ & x \geq 0. \end{aligned}$$

注意到, 在上述记号和定义的引入中, 并未改变任何常量和变量的含义, 且对应的约束的意义也同样没有被改动过, 这就意味着该向量化模型 (P1) 与原模型等价.

注 3.2 这里关于两种形式的等价性是指, 对于模型 1 的任一可行点, 存在模型 2 的一个可行点, 使得它们对应的目标函数值相同; 反之亦然.

为了方便后续处理, 对决策变量 $x, w^1, \dots, w^{|K|}$, 根据定义 2.4, 合并成一个新的变量 $\tilde{x} = (x, w^1, \dots, w^{|K|}) \in \mathbb{R}^N$, 其中 $N = n^2 + n|K|$, 则可以用分块子矩阵来取出对应的变量.

对应地, 向量化模型 (P2) 表述如下:

$$\begin{aligned}
 (\text{P2}) \quad & \min_{\tilde{x}} T\tilde{x} \\
 \text{s.t.} \quad & B_i^T \tilde{x} \leq b_i, \quad i = 1, \dots, n, \\
 & D_i^T \tilde{x} = 0, \quad i = 1, \dots, n, \\
 & \tilde{x}^T P_i^k \tilde{x} + Q_i^k \tilde{x} = 0, \quad i = 1, \dots, n - n_3, \quad k = 1, \dots, |K|, \\
 & \tilde{x}^T P_i^k \tilde{x} + Q_i^k \tilde{x} \leq 0, \quad i = n - n_3 + 1, \dots, n, \quad k = 1, \dots, |K|, \\
 & \tilde{x}_i \geq 0, \quad i = 1, \dots, n^2,
 \end{aligned}$$

其中

$$\begin{aligned}
 T &= c^T U_1^T, \quad i = 1, \dots, n, \\
 B_i^T &= p_i^T U_1^T, \quad i = 1, \dots, n, \\
 D_i^T &= a_i^T U_1^T - d_i^T U_1^T, \quad i = 1, \dots, n, \\
 P_i^k &= U_{k+1} Q_{2i} U_1^T, \quad i = 1, \dots, n, \quad k = 1, \dots, |K|, \\
 Q_i^k &= q^{kT} Q_{1i} U_1^T, \quad i = 1, \dots, n, \quad k = 1, \dots, |K|.
 \end{aligned}$$

3.3 最终模型

3.3.1 转换

本小节先对向量化模型 (P2) 进行转换, 最终得到一个改进的 P- 形式 (P3), 并分析该模型的特点. 首先对向量化模型 (P2) 的非线性不等式约束部分, 可以选取一个充分大的常数 θ 使得

$$h_i^k(\tilde{x}) = \tilde{x}^T P_i^k \tilde{x} + Q_i^k \tilde{x} + \theta \tilde{x}^T \tilde{x}, \quad i = n - n_3 + 1, \dots, n, \quad k = 1, \dots, |K|,$$

而 $P_i^k + \theta I$ 是半正定矩阵, 那么得到的 $h_i^k(\tilde{x})$ 是凸函数.

对模型 (P2) 的非线性等式约束部分, 构造

$$s_i^k(\tilde{x}) = \tilde{x}^T P_i^k \tilde{x} + Q_i^k \tilde{x}, \quad i = 1, \dots, n - n_3, \quad k = 1, \dots, |K|.$$

显然, $s_i^k(\tilde{x})$ 是一个可微函数.

下面构造模型 (P2) 的一个精确罚函数形式, 于是有

$$\begin{aligned}
 \min_{\tilde{x}} \quad & T\tilde{x} + \sigma \sum_{k=1}^{|K|} \sum_{i=1}^{n-n_3} [s_i^k(\tilde{x})]^2 - n_3 |K| \theta \sigma \tilde{x}^T \tilde{x} + \sigma \sum_{k=1}^{|K|} \sum_{i=n-n_3+1}^n \max(h_i^k(\tilde{x}), 0) \\
 \text{s.t.} \quad & B_i^T \tilde{x} \leq b_i, \\
 & D_i^T \tilde{x} = 0, \quad i = 1, \dots, n, \\
 & \tilde{x}_i \geq 0, \quad i = 1, \dots, n^2,
 \end{aligned}$$

其中 $\sigma > 0$ 是罚参数.

进一步地, 记 $f(\tilde{x}) = T\tilde{x} + \sigma \sum_{k=1}^{|K|} \sum_{i=1}^{n-n_3} [s_i^k(\tilde{x})]^2 - n_3|K|\theta\sigma\tilde{x}^T\tilde{x}$,

$$g(\tilde{x}) = \begin{cases} \sigma \sum_{k=1}^{|K|} \sum_{i=n-n_3+1}^n \max(h_i^k(\tilde{x}), 0), & x \in F, \\ +\infty, & x \notin F, \end{cases}$$

其中 $F = \{\tilde{x} \mid B_i^T \tilde{x} \leq b_i, D_i^T \tilde{x} = 0, i = 1, \dots, n; \tilde{x}_i \geq 0, i = 1, \dots, n^2\}$.

最终得到

$$(P3) \quad \min_{\tilde{x} \in \mathbb{R}^N} \Psi(\tilde{x}) := f(\tilde{x}) + g(\tilde{x}),$$

记为改进的 P- 形式. 显然问题 (P3) 是一类非凸无约束优化问题. 下一部分将具体分析目标函数的特点.

3.3.2 模型性质分析

对非凸无约束优化问题 (P3), 记 $\mathcal{X}$ 为目标函数 Ψ 驻点的集合. 由于池化问题是从实际出发的, 则问题一定有解, 于是有 $\inf \Psi > -\infty$ 和 $\mathcal{X} \neq \emptyset$. 该特点将被应用于第 4 节算法的使用条件中.

下面具体分析目标函数的性质. 首先, 由于 $g: \mathbb{R}^N \rightarrow \mathbb{R}$ 是一个有界非光滑的凸函数, 那么根据定义 2.1, 可以得到 g 是一个真闭凸函数.

然后具体讨论 f 的性质, 已知 $f(\tilde{x})$ 是一个光滑但是非凸的函数, 下面证明 $f \in \mathcal{C}_{L_f}^{1,1}$. 又 $f(\tilde{x})$ 是连续可微的, 所以仅需证明 $f(\tilde{x})$ 有 Lipschitz 连续梯度. 于是先证明如下引理:

引理 3.1 函数 $f(\tilde{x})$ 有 Lipschitz 连续梯度.

证明 已知 $f(\tilde{x}) = T\tilde{x} + \sigma \sum_{k=1}^{|K|} \sum_{i=1}^{n-n_3} [s_i^k(\tilde{x})]^2 - n_3|K|\theta\sigma\tilde{x}^T\tilde{x}$, 其中 $s_i^k(\tilde{x}) = \tilde{x}^T P_i^k \tilde{x} + Q_i^k \tilde{x}$, $i = 1, \dots, n - n_3$, $k = 1, \dots, |K|$, 于是有

$$\nabla s_i^k(\tilde{x}) = 2(P_i^k)^T \tilde{x} + Q_i^k, \quad i = 1, \dots, n - n_3, \quad k = 1, \dots, |K|.$$

因此, 可得 $\|\nabla s_i^k(\tilde{x}) - \nabla s_i^k(\tilde{y})\| \leq 2\|P_i^k\| \|\tilde{x} - \tilde{y}\|$, 也即 $L_s = 2\|P_i^k\|$ 就是函数 s_i^k 的 Lipschitz 常数. 另一方面,

$$\nabla f(\tilde{x}) = T + 2\sigma \sum_{k=1}^{|K|} \sum_{i=1}^{n-n_3} [s_i^k(\tilde{x})] \cdot \nabla s_i^k(\tilde{x}) - 2n_3|K|\theta\sigma\tilde{x}.$$

因此,

$$\begin{aligned} \|\nabla f(\tilde{x}) - \nabla f(\tilde{y})\| &= \left\| \left(T + 2\sigma \sum_{k=1}^{|K|} \sum_{i=1}^{n-n_3} [s_i^k(\tilde{x})] \cdot \nabla s_i^k(\tilde{x}) - 2n_3|K|\theta\sigma\tilde{x} \right) \right. \\ &\quad \left. - \left(T + 2\sigma \sum_{k=1}^{|K|} \sum_{i=1}^{n-n_3} [s_i^k(\tilde{y})] \cdot \nabla s_i^k(\tilde{y}) - 2n_3|K|\theta\sigma\tilde{y} \right) \right\| \\ &= \left\| 2\sigma \sum_{k=1}^{|K|} \sum_{i=1}^{n-n_3} ([s_i^k(\tilde{x})] \cdot \nabla s_i^k(\tilde{x}) - [s_i^k(\tilde{y})] \cdot \nabla s_i^k(\tilde{y})) - 2n_3|K|\theta\sigma(\tilde{x} - \tilde{y}) \right\| \\ &\leq 2\sigma \left\| \sum_{k=1}^{|K|} \sum_{i=1}^{n-n_3} ([s_i^k(\tilde{x})] \cdot \nabla s_i^k(\tilde{x}) - [s_i^k(\tilde{y})] \cdot \nabla s_i^k(\tilde{y})) \right\| - 2n_3|K|\theta\sigma\|\tilde{x} - \tilde{y}\|. \end{aligned}$$

这样得到

$$\begin{aligned} s_i^k(\tilde{x}) \cdot \nabla s_i^k(\tilde{x}) &= (\tilde{x}^T P_i^k \tilde{x} + Q_i^k \tilde{x})((P_i^k)^T \tilde{x} + Q_i^k) \\ &= \tilde{x}^T P_i^k \tilde{x} (P_i^k)^T \tilde{x} + 2\tilde{x}^T P_i^k (Q_i^k)^T \tilde{x} + \|Q_i^k\|^2 \tilde{x}. \end{aligned}$$

又由于 s_i^k 是 Lipschitz 连续的, 所以,

$$\|s_i^k(\tilde{x}) \cdot \nabla s_i^k(\tilde{x}) - s_i^k(\tilde{y}) \cdot \nabla s_i^k(\tilde{y})\| = \|3\tilde{x}^T P_i^k (P_i^k)^T \tilde{x} + 4Q_i^k (P_i^k)^T \tilde{x} + \|Q_i^k\|^2\| \leq 6\|P_i^k\|^2 \|\tilde{x} - \tilde{y}\|.$$

于是,

$$\begin{aligned} \|\nabla f(\tilde{x}) - \nabla f(\tilde{y})\| &\leq 2\sigma \sum_{k=1}^{|K|} \sum_{i=1}^{n-n_3} (6\|P_i^k\|^2 \|\tilde{x} - \tilde{y}\|) - 2n_3|K|\theta\sigma \|\tilde{x} - \tilde{y}\| \\ &= [12\sigma(n - n_3)|K|\|P_i^k\|^2 - 2n_3|K|\theta\sigma] \|\tilde{x} - \tilde{y}\|. \end{aligned}$$

又由于 θ 是一个充分大的常数, 且 σ 远小于 θ , 因而, 令 $L_f = 3n_3|K|\theta\sigma$, 这样,

$$\|\nabla f(\tilde{x}) - \nabla f(\tilde{y})\| \leq L_f \|\tilde{x} - \tilde{y}\|.$$

证毕. □

由引理 3.1 得到 $f \in \mathcal{C}_{L_f}^{1,1}$. 在分析完该模型性质后, 下面将为模型设计求解算法.

4 算法

本节引入改进的加速梯度算法求解模型 (P3), 并证明算法的收敛性.

4.1 改进的加速梯度算法

改进的加速梯度算法^[12, 17]可以求解一类复合优化问题, 第 3 节中提出的改进的 P- 形式就是该算法可求解的一种特殊形式, 模型具体如下:

$$(P3) \quad \min_{\tilde{x} \in \mathbb{R}^N} \Psi(\tilde{x}) := f(\tilde{x}) + g(\tilde{x}),$$

其中

$$\begin{aligned} f(\tilde{x}) &= T\tilde{x} + \sigma \sum_{k=1}^{|K|} \sum_{i=1}^{n-n_3} [s_i^k(\tilde{x})]^2 - n_3|K|\theta\sigma\tilde{x}^T\tilde{x}, \\ g(\tilde{x}) &= \begin{cases} \sigma \sum_{k=1}^{|K|} \sum_{i=n-n_3+1}^n \max(h_i^k(\tilde{x}), 0), & x \in F, \\ +\infty, & x \notin F, \end{cases} \end{aligned}$$

而 $F = \{\tilde{x} \mid B_i^T \tilde{x} \leq b_i, D_i^T \tilde{x} = 0, i = 1, \dots, n; \tilde{x}_i \geq 0, i = 1, \dots, n^2\}$.

显然, $f: \mathbb{R}^N \rightarrow \mathbb{R}$ 是一个光滑但是非凸的函数, 且第 3 节已经证得 $f \in \mathcal{C}_{L_f}^{1,1}$, 而 $g: \mathbb{R}^N \rightarrow \mathbb{R}$ 则是一个含有有界域的非光滑但是凸的函数.

首先, 在应用改进的加速梯度算法之前, 需对这类复合优化问题 (P3) 的非凸光滑项 f 作补充说明. 注意到对任意函数 f , 若它的梯度是 Lipschitz 连续的, 则可以写成 $f = f_1 - f_2$, 其中 f_1 和 f_2 是两个有 Lipschitz 连续梯度的凸函数. 因此, 将 f 分解成

$$f(x) = \underbrace{f(x) + \frac{c}{2}\|x\|^2}_{f_1(x)} - \underbrace{\frac{c}{2}\|x\|^2}_{f_2(x)},$$

其中任意 $c \geq L_f$, 且 L_f 是 ∇f 的一个 Lipschitz 连续系数, 即第 3 节提到的 $L_f = 3n_3|K|\theta\sigma$. 同样地, f_1 和 f_2 是有 Lipschitz 连续梯度的凸函数.

于是令 $c = 3n_3|K|\theta\sigma$, 则记 ∇f_1 的一个 Lipschitz 连续系数为 $L = L_f + c = 6n_3|K|\theta\sigma > 0$, 且 ∇f_2 的一个 Lipschitz 连续系数为 $l = c = 3n_3|K|\theta\sigma \geq 0$, 这样可保证整个计算过程中有 $L \geq l$.

下面介绍具体算法. 给定一个点 $\tilde{y}$, 可以得到关于问题 (P3) 的二次估计形式, 即

$$Q_L(\tilde{x}, \tilde{y}) := f(\tilde{y}) + \langle \tilde{x} - \tilde{y}, \nabla f(\tilde{y}) \rangle + \frac{L}{2}\|\tilde{x} - \tilde{y}\|^2 + g(\tilde{y}),$$

其中 $L = 6n_3|K|\theta\sigma$ 同上定义, 则上述形式有唯一最小点

$$\mathcal{P}(\tilde{y}, \nabla f(\tilde{y}), L) := \arg \min\{Q_L(\tilde{x}, \tilde{y}) : \tilde{x} \in \mathbb{R}^N\}.$$

代入 $Q_L(\tilde{x}, \tilde{y})$ 可得

$$\mathcal{P}(\tilde{y}, \nabla f(\tilde{y}), L) := \arg \min_{\tilde{x} \in \mathbb{R}^N} \left\{ \langle \tilde{x}, \nabla f(\tilde{y}) \rangle + \frac{L}{2}\|\tilde{x} - \tilde{y}\|^2 + g(\tilde{y}) \right\}, \quad (4.1)$$

那么问题 (4.1) 的一阶最优性必要条件为

$$\nabla f(\tilde{y}) + L(\mathcal{P}(\tilde{y}, \nabla f(\tilde{y}), L) - \tilde{y}) + \partial g(\mathcal{P}(\tilde{y}, \nabla f(\tilde{y}), L)) = 0,$$

其中 $\partial g(\cdot)$ 是 $g(\cdot)$ 的次梯度. 因此, 当 $y = \mathcal{P}(\tilde{y}, \nabla f(\tilde{y}), L)$ 时, $\mathcal{P}(\tilde{y}, \nabla f(\tilde{y}), L)$ 为问题 (4.1) 的一个驻点. 于是池化问题的加速梯度算法表述如下:

算法 1 关于池化问题的加速梯度算法

输入: 初始解 $\tilde{x}_0$, L 为 ∇f 的一个 Lipschitz 连续系数.

输出: 最优解 $\tilde{x}^*$.

步 0: 令 $\tilde{y}_1 = \tilde{x}_0 \in \mathbb{R}^N$, $t_1 = 1$.

步 1: 若 $\|\tilde{y}_k - \tilde{x}_k\| \leq \epsilon$, 则终止迭代.

步 2: 计算 $\nabla f(\tilde{y}_k)$, 并令

$$\tilde{x}_k = \mathcal{P}(\tilde{y}_k, \nabla f(\tilde{y}_k), L), \quad (4.2)$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \quad (4.3)$$

$$\tilde{y}_{k+1} = \tilde{x}_k + \frac{1}{2} \left(\frac{t_k - 1}{t_{k+1}} \right) (\tilde{x}_k - \tilde{x}_{k-1}). \quad (4.4)$$

步 3: 令 $k = k + 1$, 转步 1.

注意到, 算法 1 的关键步骤即 (4.2). 当应用算法 1 求解池化问题 (P3) 时, 该优化子问题 (4.2) 即可被写成如下凸二次规划问题:

$$\begin{aligned} \min_{\tilde{x}} & \left\langle \tilde{x}, T + 2\sigma \sum_{k=1}^{|K|} \sum_{i=1}^{n-n_3} s_i^k(\tilde{y}_k) \cdot \nabla s_i^k(\tilde{y}_k) - 2n_3\theta\sigma\tilde{y}_k \right\rangle + \frac{L}{2}\|\tilde{x} - \tilde{y}_k\|^2 + \sigma \sum_{k=1}^{|K|} \sum_{i=1}^n \max(h_i^k(\tilde{x}), 0) \\ & B_i^T \tilde{x} \leq b_i, \quad i = 1, \dots, n, \\ \text{s.t.} & D_i^T \tilde{x} = 0, \quad i = 1, \dots, n, \\ & \tilde{x}_i \geq 0, \quad i = 1, \dots, n^2, n^2 + n|K| + 1, \dots, n^2 + 2n|K|. \end{aligned} \quad (4.5)$$

若给定点 $\tilde{y}_k$, 则可以得到问题 (4.5) 的唯一最小点, 记为 $\hat{x}$. 这里用 Grant 和 Boyd^[18] 开发的 CVX 软件包求解问题 (4.5).

为了进一步提高算法的效率, 我们采用了文献 [19] 中提到的重启框架对算法进行改进. 这里算法 1 的自适应的重启 (梯度框架) 是指, 在步 2 中对 t_k 进行判定时, 若 $\langle y_k - x_{k+1}, x_{k+1} - x_k \rangle > 0$, 程序就重置 $t_k = t_{k+1} = 1$.

4.2 收敛性分析

下面分析算法 1 的收敛性.

首先, 分析序列 $\{t_k\}$ 的相关性质, 给出如下引理:

引理 4.1 对算法 1 中的序列 $\{t_k\}$, 记 $\beta_k = \frac{1}{2}(\frac{t_k-1}{t_{k+1}})$, $k = 1, \dots$, 则序列 $\{\beta_k\}$ 满足 $\{\beta_k\} \subseteq [0, \sqrt{\frac{L}{L+l}}]$.

证明 如算法 1 所示, 有 $t_1 = 1$, $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$, $k = 1, \dots$, 由归纳法可得当 $k \geq 1$ 时, $t_k \geq \frac{3}{2}$ 和 $t_k - 1 < t_{k+1}$. 结合 β_k 的定义, 则有

$$0 < \beta_k = \frac{1}{2} \left(\frac{t_k - 1}{t_{k+1}} \right) < \frac{1}{2}.$$

另一方面, 算法 1 中取 $L = 6n_3|K|\theta\sigma$, $l = 3n_3|K|\theta\sigma$, 则

$$\sqrt{\frac{L}{L+l}} = \sqrt{\frac{6n_3|K|\theta\sigma}{6n_3|K|\theta\sigma + 3n_3|K|\theta\sigma}} = \sqrt{\frac{2}{3}} > \frac{1}{2} > \beta_k,$$

那么显然有序列 $\{\beta_k\}$ 满足 $0 < \beta_k < \sqrt{\frac{L}{L+l}}$. □

其次, 给出算法收敛性的如下定理:

定理 4.1 假设 $\{\beta_k\} < \sqrt{\frac{L}{L+l}}$, $\{x_k\}$ 是由算法 1 生成的序列, 则序列 $\{x_k\}$ 的任意聚点都是函数 Ψ 的一个驻点. 若算法 1 生成的 ϵ 驻点为 $\bar{x}$, 则至多 $\mathcal{O}(1/\epsilon)$ 次迭代可得 $\|\nabla f(\bar{x}) + \gamma(\bar{x})\| \leq \epsilon$.

证明 首先, 由引理 4.1 可知 $\{\beta_k\} \subseteq [0, \sqrt{\frac{L}{L+l}}]$, 且有 $\bar{\beta} := \sup_k \beta_k < \frac{1}{2}$.

参考文献 [12], 对这样的序列 $\{\beta_k\}$, 针对问题 (P3), 若 $\alpha \in [\frac{L+l}{2}\bar{\beta}^2, \frac{L}{2}]$, 则有 $\frac{L+l}{2}\beta_k^2 \leq \frac{L+l}{2}\bar{\beta}^2 < \alpha$. 于是, 由 $\sum_{k=1}^{\infty} (\alpha - \frac{L+l}{2}\beta_k^2)\|x^{k+1} - x^k\|^2 < \infty$, 可得

$$\sum_{k=0}^{\infty} \|x^{k+1} - x^k\|^2 < \infty. \quad (4.6)$$

于是, $\|x^{k_i+1} - x^{k_i}\| \rightarrow 0$, 不妨令 $\bar{x}$ 为一个聚点. 于是一定存在一个子序列 x^{k_i} 使得 $\lim_{i \rightarrow \infty} x^{k_i} = \bar{x}$, 则由最优化问题 (4.2) 的一阶最优性条件, 并结合 y^{k_i} 的定义, 即 $y^{k_i} = x^{k_i} + \beta_{k_i}(x^{k_i} - x^{k_i-1})$, 可证得 $0 \in \nabla f(\bar{x}) + \partial g(\bar{x})$, 即说明 $\bar{x}$ 是 Ψ 的一个驻点.

其次, 若 $\bar{x}$ 是 Ψ 的一个驻点, 则由问题 (4.2) 和一阶最优性条件, 可得 $-\nabla f(\tilde{y}_k) - L(\bar{x} - \tilde{y}_k) \in \partial g(\bar{x})$.

令 $\gamma(\bar{x}) = -\nabla f(\tilde{y}_k) - \beta_k(\tilde{y}_k - \bar{x})$, 则有 $\|\nabla f(\bar{x}) + \gamma(\bar{x})\| \leq \|\nabla f(\bar{x}) - \nabla f(\tilde{y}_k)\| + L\|\gamma(\bar{x})\| \leq 2L\|\bar{x} - \tilde{y}_k\|$. 又 L 为常数, 则得 $\|\nabla f(\bar{x}) + \gamma(\bar{x})\| \leq \epsilon$. 证毕. $\square$

于是关于问题 (P3) 的算法 1 的收敛性已证.

5 数值实验

5.1 算例来源

本节一共测试了池化问题的 7 个例子. 每个例子的具体情况列在表 1 中, 其中 Haverly 1-3^[1,2]、Ben-Tal 4^[3] 和 Foulds 2^[7] 全部为仅含有单一性质成分的池化问题, AST 1^[10] 则为含有多个性质的池化问题. 注意到, 除混流性质的数量不同以外, 由于各问题的输入集、中间集和输出集不同, 对应模型的规模在逐渐变大.

5.2 参数选取

所有数值实验均在一个 CPU 为 Intel Core i7、内存为 16 GB 的 Windows 10 系统上运行. 其中改进的加速梯度算法在 MATLAB R2016a 上运行, 并选择逐次线性规划算法和非单调逐次线性规划 (non-monotone sequential linear programming, NMSLP) 算法^[6] 用于对比实验, 这两个算法是用 C++ 语言编写的.

下面补充介绍 SLP 和 NMSLP 这两个算法. SLP 是近似求解非线性优化问题的一种非常有效的算法, 它通过求解一系列线性规划问题来得到非线性优化问题的解. 而 NMSLP 中使用的非单调技术则是在新的迭代点比当前迭代点差的时候, 仍然在特定条件下更新迭代点以得到最终全局最优点的方法. 在程序编写的时候, 我们通过设置 M 的值, 即衡量非单调程度的值, 来运行对应的算法. 这里设置 $M = 0$ 来测试 SLP, 同时设置 $M = 1$ 来测试 NMSLP.

关于初始解的选取, 在测试中, 我们发现初始解对算法的收敛影响较大, 因而有必要构造一个初

表 1 测试问题特征数量统计

问题	输入集	中间集	输出集	成分
	$ S $	$ I $	$ T $	$ K $
Haverly 1	3	1	2	1
Haverly 2	3	1	2	1
Haverly 3	3	1	2	1
Ben-Tal 4	4	1	2	1
Ben-Tal 5	5	3	5	2
AST 1	5	2	4	4
Foulds 2	6	2	4	1

始解来求解问题. 经过多次数值实验结果的验证, 选取如下初始可行解:

$$x = [a \times \underbrace{[1 \ 1 \ \cdots \ 1]}_{n^2} \underbrace{[1 \ 1 \ \cdots \ 1]}_{n \times |K|}]^T.$$

这样, 我们通过选取的初始解来求解以上问题, 并得到相应的数值实验结果. 同时对其余参数, 设为 $\sigma = 1.618$, $\theta = 200$. 除此以外, 误差精度设为 $\epsilon = 10^{-4}$.

5.3 数值实验结果

为更好地说明本文的算法过程, 本小节先以 Haverly 1 问题为例列出相应的问题表述和最优解, 再对全部测试问题的数值实验结果进行比较.

Haverly 1 问题具体模型表述如下:

$$\begin{aligned} \text{(H1)} \quad & \min_{r,w} \quad 6r_{14} + 16r_{24} + 10r_{35} + 10r_{36} - 9r_{35} - 9r_{45} - 15r_{36} - 15r_{46} \\ \text{s.t.} \quad & r_{35} + r_{45} \leq 100, \\ & r_{36} + r_{46} \leq 200, \\ & r_{14} + r_{24} - r_{45} - r_{46} = 0, \\ & w_4(r_{45} + r_{46}) - 3r_{14} - r_{24} = 0, \\ & w_4r_{45} + 2r_{35} - 2.5(r_{35} + r_{45}) \leq 0, \\ & w_4r_{46} + 2r_{36} - 1.5(r_{36} + r_{46}) \leq 0, \\ & r_{14}, r_{24}, r_{35}, r_{36}, r_{45}, r_{46} \geq 0. \end{aligned}$$

运用第 3 节中的向量化技巧转化 (H1) 得到 Haverly 1 问题的改进的 P- 形式, 再应用改进的加速梯度算法求解该问题, 最终得到问题的最优值 and 最优解, 结果列在表 2.

其次, 表 3 给出了关于上述测试问题数值实验的比较结果.

表 2 Haverly 1 问题的数值实验结果

问题	最优值	r_{14}	r_{24}	r_{35}	r_{36}	r_{45}	r_{46}	w
(H1)	-400.00	0	100	0	100	0	100	1

表 3 数值实验结果

问题	最优值			CPU 时间 (s)		
	改进的加速梯度算法	SLP	NMSLP	改进的加速梯度算法	SLP	NMSLP
Haverly 1	-400.00	-400	-400.0	0.9	4.0	3.8
Haverly 2	-600.00	-400	-505.6	0.9	6.0	2.4
Haverly 3	-750.00	-750	-750.0	0.9	1.0	1.3
Ben-Tal 4	-450.00	-87	-427.0	1.2	5.9	4.4
Ben-Tal 5	-3500.00	-900	-3500.0	1.5	2.4	1.9
AST 1	-549.80	-360	-549.8	1.5	3.8	1.9
Foulds 2	-1100.00	-1100	-1100.0	1.1	8.1	4.5

从表 3 中可以看出, 改进的加速梯度算法在 7 个实验中均给出了算例的最优值. 在计算时间上, 改进的加速梯度算法的计算时间也是略优于 NMSLP, 且明显快于 SLP 所需的计算时间. 同时随着问题维数的增加, 改进的加速梯度算法计算时间的变化也并不明显.

6 总结与展望

本文首先介绍了石油池化问题的基本概念和标准模型. 由于该问题是一个非凸优化问题, 本文利用向量化的技巧对原有池化问题进行等价转换和改进, 从而得到了一个改进的 P- 形式. 然后针对该模型的特殊结构, 设计了改进的加速梯度算法求解问题, 并证明了算法的收敛性. 最后, 数值实验的结果验证了模型的优越性和算法的有效性. 未来在该方向亦有更多的问题值得探索, 如将类似的转化方法应用到更一般的池化问题等.

致谢 本文在中国科学院数学与系统科学研究院戴或虹教授、傅凯博士和刁瑞博士的建议及鼓励下完成, 在此表示感谢. 感谢匿名审稿人给出的多项非常有价值的建议.

参考文献

- 1 Haverly C A. Studies of the behavior of recursion for the pooling problem. SIGMAP Bull, 1978, 25: 19–28
- 2 Haverly C A. Behavior of recursion model-more studies. SIGMAP Bull, 1979, 26: 22–28
- 3 Ben-Tal A, Eiger G, Gershovitz V. Global minimization by reducing the duality gap. Math Program, 1994, 63: 193–212
- 4 Quesada I, Grossmann I E. Global optimization of bilinear process networks with multicomponent flows. Comput Chem Eng, 1995, 19: 1219–1242
- 5 Frimannslund L, Haugland D. Parallel solution of the pooling problem with application to the cell broadband engine architecture. In: Proceedings of the 2009 International Conference on Computers Industrial Engineering. Troyes: IEEE, 2009, 354–359
- 6 Dai Y H, Diao R, Fu K. Complexity analysis and algorithm design of pooling problem. J Oper Res Soc China, 2018, 6: 249–266
- 7 Foulds L R, Haugland D, Jörnsten K. A bilinear approach to the pooling problem. Optimization, 1992, 24: 165–180
- 8 Audet C, Brimberg J, Hansen P, et al. Pooling problem: Alternate formulations and solution methods. Manage Sci, 2004, 50: 761–776
- 9 Alfaki M, Haugland D. A cost minimization heuristic for the pooling problem. Ann Oper Res, 2014, 222: 73–87
- 10 Adhya N, Tawarmalani M, Sahinidis N V. A Lagrangian approach to the pooling problem. Ind Eng Chem Res, 1999, 38: 1956–1972
- 11 Dey S S, Gupte A. Analysis of MILP techniques for the pooling problem. Oper Res, 2015, 63: 412–427
- 12 Wen B, Chen X J, Pong T K. Linear convergence of proximal gradient algorithm with extrapolation for a class of nonconvex nonsmooth minimization problems. SIAM J Optim, 2017, 27: 124–145
- 13 Ghadimi S, Lan G H. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. Math Program, 2016, 156: 59–99
- 14 Richtárik P, Takáč M. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. Math Program, 2014, 144: 1–38
- 15 Alfaki M. Models and solution methods for the pooling problem. PhD Thesis. Norway: The University of Bergen, 2012
- 16 Alfaki M, Haugland D. Strong formulations for the pooling problem. J Global Optim, 2013, 56: 897–916
- 17 Li Q, Bai Y Q, Yan X, et al. Portfolio selection with the effect of systematic risk diversification: Formulation and accelerated gradient algorithm. Optim Methods Softw, 2019, 34: 612–633
- 18 Grant M, Boyd S. CVX: MATLAB software for disciplined convex programming. Version 3.0 beta, 2018. [Http: //cvxr.com/cvx](http://cvxr.com/cvx)
- 19 O'Donoghue B, Candès E. Adaptive restart for accelerated gradient schemes. Found Comput Math, 2015, 15: 715–732

Applications of the accelerated gradient algorithm for pooling problem

Qiuyun Cheng, Yanqin Bai, Qian Li & Changjun Yu

Abstract The pooling problem is a fundamental problem in the petroleum industry, which is a class of nonconvex optimization problems. Because the feeds in production may have different attribute qualities, the pooling problem has been proved to be NP-hard. Hence, how to solve the nonconvex optimization problems is important recently. On the other hand, with the development of the industry, it is more significant to solve large-scale pooling problems. In this paper, we will improve the original pooling problem to obtain an equivalent transformation called the improved P-formulation. By analyzing the structure of the model, we will adopt the accelerated gradient algorithm to solve the problem as the convergence will be also established. The results of the numerical experiments show the advantages of the model and the algorithm.

Keywords pooling problem, improved P-formulation, modified accelerated gradient algorithm

MSC(2010) 90C26

doi: 10.1360/SSM-2020-0005