



线性规划的改进行旋转算法

刘燕武¹, 涂燕¹, 周晓阳^{2,3*}, 汪寿阳³, 张忠桢⁴

1. 武汉理工大学安全科学与应急管理学院, 武汉 430070;

2. 西安交通大学管理学院, 西安 710049;

3. 中国科学院数学与系统科学研究院, 北京 100190;

4. 武汉理工大学管理学院, 武汉 430070

E-mail: lywwhut@163.com, tuyan_belle@163.com, zhouxiaoyang@xjtu.edu.cn, sywang@amss.ac.cn, zhangzz321@126.com

收稿日期: 2021-06-30; 接受日期: 2023-04-21; 网络出版日期: 2023-09-25; * 通信作者

国家自然科学基金 (批准号: 72271194) 资助项目

摘要 行旋转算法是求解线性不等式组的一种直接、有效的算法, 为大数据时代众多与线性不等式组紧密相关的问题提供了统一、高效的解决思路. 求解线性规划问题本质上是求解线性不等式组, 因而行旋转算法可以作为基础算法直接应用于求解线性规划. 不同于以单纯形法为代表的列旋转算法, 线性规划的行旋转算法以行几何 (或行向量) 为基础, 其核心思想是在保证最优性条件始终成立的前提下求解约束条件对应的线性不等式组. 改进的行旋转算法保持了原算法的所有特色. 该算法的改进之处在于利用约束条件变量的部分系数构成的非奇异矩阵的逆矩阵 (称为特征逆矩阵) 和原始数据计算出枢轴行和枢轴列, 从而完成一次旋转运算. 特征逆矩阵的阶数一般要比约束的数目和变量的数目小很多, 在每次迭代过程中只需要计算原算法算表中的小部分必要元素, 因而能够显著提高计算效率.

关键词 线性规划 改进行旋转算法 逆矩阵 分块行旋转运算 线性不等式组

MSC (2020) 主题分类 90C05, 90-08

1 引言

在资源稀缺的常态下, 寻找能够获得问题最优解决方案并进行后优化分析的有效方法一直是最优化领域研究者孜孜不倦追求的目标. 自 20 世纪 40 年代以来, 线性规划一直是最优化领域最重要的分支^[11,12], 在经济学、工程学和金融学等众多领域具有大量应用 (参见文献 [3–6,9,25]). 旋转算法和内点算法是求解线性规划问题最主要的两类算法. 1947 年, Dantzig 第一个发展出求解一般线性规划问题的单纯形法 (参见文献 [8,15]). 在此后相当长的时间内, 单纯形法及其变体 (统称为旋转算法) 成为求解线性规划问题唯一有效的方法. 1984 年, Karmarkar^[16] 提出了具有里程碑意义的内点算法 (interior point method, IPM). 自 Karmarkar 的创造性工作之后, 内点算法成为研究热点, 在理论上展现出很好

英文引用格式: Liu Y W, Tu Y, Zhou X Y, et al. The revised row pivoting method for linear programming (in Chinese). Sci Sin Math, 2023, 53: 1509–1532, doi: 10.1360/SSM-2021-0123

的时间复杂性^[14, 20, 23]. 单纯形法则在实践中展示出更强的竞争力, 在后优化分析和整数规划中具有显著优势. 两类算法在求解不同类型的大规模线性规划问题上各具特色 (参见文献 [15, 17]).

以单纯形法及其变体为代表的旋转算法实际上都是列旋转算法, 以列几何 (列向量) 为基础. 由于单纯形法巨大的成功和影响力, 列旋转算法成为最优化领域的标准, 几乎成为旋转算法的代名词. 然而, 与人们的想象不同, Dantzig^[12] 最初研究的却是基于行几何的算法. 在经过艰苦的尝试后, Dantzig 认为基于行几何的算法沿着外部边沿运动, 看起来毫无希望. 幸运的是, Dantzig 用列几何 (列向量) 代替行几何 (行向量), 发现基于列几何的单纯形法是求解线性规划的一种有效方法. Dantzig 之所以一开始考虑行向量, 其实非常自然. 行向量与目标函数、约束等式和约束不等式直接相关, 而列向量与它们并没有直接的联系. 行向量实际上是表示线性规划问题更直接、更自然的方法, 因此探索行几何是否是求解线性规划的可行途径是一个具有方法论意义的问题. 文献 [19] 首次提出了线性规划的行旋转算法, 从理论上肯定地回答了 Dantzig 早期关于“行几何是否是求解线性规划的可行途径”的探索. 行旋转算法的提出改变了线性规划旋转算法实际上只包含列旋转算法的现状, 丰富了线性规划旋转算法的内涵.

Dantzig 进一步指出, 线性规划表面上涉及约束最优化问题, 实际上却是求解线性不等式组^[8, 11]. Dantzig 这段关于线性规划本质的哲学思考, 具有重要的方法论意义. 实际上, 很多优化问题本质上都是求解线性不等式组. 在某种程度上, 线性不等式组是比线性方程组更为基础的问题. 线性方程组可以等价表示为线性不等式组, 而线性不等式组却不能等价表示成线性方程组. 但是, 由于没有发展出像线性方程组 Gauss 消元法那样直接、有效地求解线性不等式组的算法, 线性不等式组的基础性作用大打折扣. 如果能够发展出直接、有效地求解线性不等式组的算法, 那么许多与线性不等式组紧密相关的问题就可以以该算法为基础以一种统一的方式求解. 因此, 探索求解线性不等式组的直接、有效的算法也是一件具有方法论意义的事情.

通过引入行向量最大线性无关组构成的基这一核心概念, 发现从行几何视角能充分发掘 Farkas 引理求解线性不等式组的巨大潜力, 并设计出一种求解线性不等式组的直接、有效的行旋转算法. 求解线性规划可以看作是保证最优性条件成立的前提条件下求解其约束条件对应的线性不等式组, 所以基于线性不等式组的行旋转算法可以自然地发展出求解线性规划的行旋转算法. 因此, 线性规划的行旋转算法回答了 Dantzig 对线性规划的两个哲学思考: 一是行几何是求解线性规划的可行途径; 二是可以运用求解一般线性不等式组的方法求解线性规划问题.

线性规划的行旋转算法具有如下特色: (1) 由于目标函数和约束条件都可以由行向量直接表示, 所以不需要增加任何辅助变量, 对应的矩阵规模比其他旋转算法矩阵规模小; (2) 由于行旋转算法充分揭示了行向量之间的内在联系, 因此可以自动识别冗余 (不) 等式和矛盾 (不) 等式, 不需要对系数矩阵作任何假设; (3) 由于等式是不等式的特殊形式, 因此行旋转算法可以直接处理等式约束, 并在预处理阶段直接消除所有等式约束; (4) 由于从行向量角度看, 上下界不等式具有直接的联系, 因此行旋转算法可以同时处理上下界不等式; (5) 由于 (不) 等式约束对应行向量总可以由正负单位行向量构成的基直接表示, 所以行旋转算法可以直接启动. 行旋转算法的这些计算特色是由行几何直接衍生出来的, 不需要额外设计任何特殊处理方法, 这是与列旋转算法的一个重要区别 (参见文献 [1, 2, 7, 10, 18, 21, 22, 24]).

线性规划的行旋转算法具有一些显著的计算特色, 但却是一种全表格算法, 即每经过一次旋转, 算表中所有元素都要更新. 事实上, 在行旋转算法的每次迭代过程中, 并不需要计算所有元素, 只需要记录迭代到当前状态的逆矩阵信息 (称为当前迭代的特征逆矩阵) 和模型的初始信息, 就可以随时计算出所有必要元素的信息. 因此, 在实际行旋转运算过程中只需计算当前迭代的特征逆矩阵, 并基于特征逆矩阵和模型初始信息计算必要元素, 这便是线性规划的改进行旋转算法的基本思路. 由于相应的

计算量远小于全表格算表计算量, 因此改进行旋转算法能够显著提高原算法的计算效率.

2 基础概念及其方法论意义

基础概念是构建理论的重要基石, 其一般性和普适性对于理论的一般性和普适性有重要影响.

2.1 基础概念

线性规划行旋转算法的标准型以及基、基本解和偏差等基础概念与单纯形法有很大的不同. 不加特别说明, 本文中系数向量均为行向量, 与变量相关的向量为列向量. 本文中线性规划问题的标准型如下:

$$\begin{aligned} \min z &= \mathbf{c}\mathbf{x} \\ \text{s.t. } \mathbf{a}_i\mathbf{x} &= b_i, \quad i = 1, 2, \dots, l, \\ \mathbf{a}_i\mathbf{x} &\geq b_i, \quad i = l+1, l+2, \dots, m, \end{aligned} \quad (2.1)$$

其中, $\mathbf{a}_i = [a_{i1}, a_{i2}, \dots, a_{in}]$ ($i = 1, 2, \dots, m$) 和 $\mathbf{c} = [c_1, c_2, \dots, c_n]$ 是 n 维行向量, $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ 是 n 维列向量, 标量 b_i ($i = 1, 2, \dots, m$) 是任意实数, l 和 m 是非负整数, $0 \leq l \leq m$. 如果 $l = 0$, 则模型 (2.1) 约束条件中只有不等式, 没有等式; 如果 $l = m$, 则模型 (2.1) 约束条件中只有等式, 没有不等式; 如果 $0 < l < m$, 则模型 (2.1) 约束条件中同时包含等式和不等式.

行旋转算法对模型 (2.1) 中约束条件和变量没有任何额外限定, 显然任何线性规划问题都可以在不添加任何辅助变量的条件下直接方便地转化为形如模型 (2.1) 的标准型. 需要特别指出的是, 模型 (2.1) 中约束条件

$$\begin{aligned} \mathbf{a}_i\mathbf{x} &= b_i, \quad i = 1, 2, \dots, l, \\ \mathbf{a}_i\mathbf{x} &\geq b_i, \quad i = l+1, l+2, \dots, m \end{aligned} \quad (2.2)$$

就是行旋转算法中线性不等式组的标准型. 行旋转算法对 (2.2) 中 (不) 等式约束和变量没有任何限制, 因此, 任意线性不等式组都可以不添加任何辅助变量、简单直接地转化为形如模型 (2.2) 的标准型. 如果 (2.2) 中解集为空集、则称不等式组 (2.2) 不相容; 反之, 则称不等式组 (2.2) 相容. 如果 (2.2) 的解集为空集, 表明模型 (2.1) 没有可行解.

定义 2.1 对于任意给定的 $\mathbf{x}$, $\sigma_i(\mathbf{x}) = \mathbf{a}_i\mathbf{x} - b_i$ ($i = 1, 2, \dots, m$) 称为等式 $\mathbf{a}_i\mathbf{x} = b_i$ ($i = 1, 2, \dots, l$) 或不等式 $\mathbf{a}_i\mathbf{x} \geq b_i$ ($i = l+1, l+2, \dots, m$) 在 $\mathbf{x}$ 下的偏差.

在下文中, 将 $\sigma_i(\mathbf{x})$ 简写为 σ_i . 对于等式, 如果 $\sigma_i = 0$, 则称对应等式在当前 $\mathbf{x}$ 下为满足等式; 否则, 称为违反等式. 对于不等式, 如果 $\sigma_i \geq 0$, 则称对应不等式在当前 $\mathbf{x}$ 下为满足不等式; 否则, 称为违反不等式.

模型 (2.2) 的矩阵形式如下:

$$\begin{aligned} \mathbf{A}^1\mathbf{x} &= \mathbf{b}^1, \\ \mathbf{A}^2\mathbf{x} &\geq \mathbf{b}^2, \end{aligned}$$

其中,

$$\mathbf{A}^1 = \begin{bmatrix} \mathbf{a}_1 \\ \vdots \\ \mathbf{a}_l \end{bmatrix}, \quad \mathbf{A}^2 = \begin{bmatrix} \mathbf{a}_{l+1} \\ \vdots \\ \mathbf{a}_m \end{bmatrix}, \quad \mathbf{b}^1 = \begin{bmatrix} b_1 \\ \vdots \\ b_l \end{bmatrix}, \quad \mathbf{b}^2 = \begin{bmatrix} b_{l+1} \\ \vdots \\ b_m \end{bmatrix}.$$

令

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}^1 \\ \mathbf{A}^2 \end{bmatrix} \quad \text{和} \quad \mathbf{b} = \begin{bmatrix} \mathbf{b}^1 \\ \mathbf{b}^2 \end{bmatrix}.$$

设 k 是所有行向量 $\mathbf{a}_i$ ($i = 1, 2, \dots, m$) 中线性无关向量的最大数目, 即 $k = \text{rank}(\mathbf{A})$. 由于行旋转算法不对约束条件作任何限制, 因此 $k \leq \min\{m, n\}$. 任何 k 个线性无关的 $\mathbf{a}_i$ 构成矩阵 $\mathbf{A}$ 的一个行基向量组, 由这 k 个行基向量构成的矩阵称为矩阵 $\mathbf{A}$ 的一个行基矩阵, 简称行基. 假设 $\mathbf{B}$ 是矩阵 $\mathbf{A}$ 的一个行基, 行基中行向量对应的 (不) 等式称为基 (不) 等式, 非基中向量对应的 (不) 等式称为非基 (不) 等式. 定义 β 为 $\mathbf{B}$ 中行向量 $\mathbf{a}_i$ 的指标集, 则 $\mathbf{B} = \mathbf{A}_\beta$. 指标集 β 有如下两个子集:

$$\begin{aligned} \beta_1 &= \{i \in \beta : 1 \leq i \leq l\}, \\ \beta_2 &= \{i \in \beta : l+1 \leq i \leq m\}, \end{aligned}$$

β_1 和 β_2 分别表示基等式和基不等式的指标集.

定义 β 相对 $\{1, 2, \dots, m\}$ 的补集如下:

$$v = \{1, 2, \dots, m\} \setminus \beta,$$

并定义 $v_1 = \{i \in v : 1 \leq i \leq l\}$, $v_2 = \{i \in v : l+1 \leq i \leq m\}$, v_1 和 v_2 分别表示非基等式和非基不等式的指标集.

显然矩阵 $\mathbf{A}$ 的非基行向量构成的子矩阵 $\mathbf{N} = \mathbf{A}_v$, 且 $v = v_1 \cup v_2$. $\mathbf{A}$ 中任何非基行向量都可以唯一表示为 $\mathbf{B}$ 中行基向量的线性组合, 因此存在唯一一组标量 w_{ij} 使下式成立:

$$\mathbf{a}_i = \sum_{j \in \beta} w_{ij} \mathbf{a}_j, \quad i \in v.$$

基于上式, 不等式组 (2.2) 可以等价表示为如下形式:

$$\begin{cases} \mathbf{a}_i \mathbf{x} = b_i, & i \in \beta_1, \\ \mathbf{a}_i \mathbf{x} \geq b_i, & i \in \beta_2, \\ \mathbf{a}_i \mathbf{x} = \sum_{j \in \beta} w_{ij} \mathbf{a}_j \mathbf{x} = b_i, & i \in v_1, \\ \mathbf{a}_i \mathbf{x} = \sum_{j \in \beta} w_{ij} \mathbf{a}_j \mathbf{x} \geq b_i, & i \in v_2. \end{cases} \quad (2.3)$$

(2.3) 表明任何线性不等式组都可以等价表示为由行基向量组表示的线性不等式组.

基等式和基不等式构成了不等式组 (2.2) 的一个子系统, 称为基不等式组. 通过求解基不等式组对应的方程组可以求得基不等式组的一个解, 称这个解为线性不等式组的行基本解. 如果这个行基本解满足 (2.1) 中所有约束条件, 则称这个行基本解为 (2.1) 的可行行基本解, 对应的行基称为可行行基. 行基及其对应的行基本解是线性规划行旋转算法中的核心概念, 因此将行基本解正式定义如下:

定义 2.2 对于任意给定的行基 B , 模型 (2.1) 的一个行基本解是线性方程组 $a_i x = b_i$ ($i \in \beta$) 的一个解.

在下文中如不加特别说明, 基均指行基, 基本解均指行基本解. 行旋转算法对矩阵的秩没有任何限定, 所以矩阵 A 的秩 $k \leq n$. 当 $k = n$ 时, 定义 2.2 中的方程组在当前基 B 下有唯一解; 当 $k < n$ 时, 定义 2.2 中的方程组在当前基 B 下有无穷多个解. 进一步考虑在当前基下存在无穷多个基本解的情形. 不妨设 $\bar{x}$ 是这无穷多个基本解中的任意一个, 则由 (2.3) 和定义 2.2 有

$$\begin{cases} a_i \bar{x} = b_i, & i \in \beta_1, \\ a_i \bar{x} = b_i, & i \in \beta_2, \\ a_i \bar{x} = \sum_{j \in \beta} w_{ij} a_j \bar{x} = \sum_{j \in \beta} w_{ij} b_j = b_i, & i \in v_1, \\ a_i \bar{x} = \sum_{j \in \beta} w_{ij} a_j \bar{x} = \sum_{j \in \beta} w_{ij} b_j \geq b_i, & i \in v_2. \end{cases} \quad (2.4)$$

从 (2.4) 可以看出, 在给定基 B 的条件下, 与该基对应的无穷多个基本解对约束条件有完全相同的作用, 每个非基 (不) 等式在当前基对应的任意基本解下偏差不变, 因此只需要计算出无穷多个基本解中的任意一个即可. 这样, 无论当前基 B 的秩 $k = n$ 还是 $k < n$, 对于每一个给定的基, 只需计算一个基本解.

2.2 基础概念的方法论意义

基础概念和基础定理是任何数学理论的基石, 基础概念和基础定理的一般性和普适性在很大程度上决定了数学理论的一般性和普适性. 从上面标准型、行基和行基本解的定义可以看出, 这些基础概念具有自然、普适、直接的特点, 基础概念之间具有一致性和自洽性, 基础概念与基础定理之间也具有内在一致性. 具体说明如下:

(1) 线性不等式组的标准型直接、简单、自然. 任意线性不等式组都可以在不添加任何辅助变量的情形下直接转为本文中的标准型, 原汁原味地保存了原线性不等式组的初始信息, 不需要对原线性不等式组作任何假设. 线性不等式组标准型的这种特点也直接遗传给其衍生问题线性规划、二次规划等问题. 在行旋转算法视角下, 线性不等式组的衍生问题的标准型都具有直接、简单、自然的特点.

(2) 对线性不等式组行基及其对应的行基本解的定义具有自然、一般、普适的特点. 对于每个线性 (不) 等式, 其系数向量是行向量, 用行向量 (行几何) 描述线性 (不) 等式很自然; 本文对线性不等式组行基的秩不作任何假设, 这与本文中对线性不等式组不作任何假设的特点是内在统一的; 本文定义的线性不等式组行基和行基本解的概念可以用来描述任意线性不等式组 (包括其特例线性方程组), 具有很好一般性和普适性.

(3) 这些基础概念与行旋转算法的基础定理—Farkas 引理—是内在统一的. Farkas 引理是最优化领域中最基础、最强大的理论之一, 在众多情形下有着广泛的应用. 线性不等式组行旋转算法是 Farkas 引理在求解线性不等式组方面巨大潜力的直接应用. Farkas 引理描述了两组线性不等式组之间解的关系, 该理论对线性不等式组的系数矩阵不作任何假设, 这与本文的基础概念内在统一.

(4) 基础概念和基础理论的无缝衔接使得行旋转算法成为求解任意线性不等式组的直接、有效的算法. 行基和行基本解既能直接、有效地描述任意线性不等式组, 又能无缝衔接 Farkas 引理, 能够充分应用 Farkas 引理揭示线性不等式组解的关系的巨大潜力, 因而二者结合就发展出能够求解任意线性不等式组的行旋转算法.

(5) 基础概念和 Farkas 引理的有机结合使得线性不等式组的行旋转算法能够自动揭示线性不等式组的本质结构, 从而使得线性不等式组行旋转算法具有良好的先天计算特色.

(6) 线性不等式组的行旋转算法具有两方面的统一性. 一是算法理论和算法实现都坚持统一的行向量 (行几何) 视角; 二是作为可以直接、有效地求解任意线性不等式组的一般方法, 该算法能够作为那些可以视为特殊线性不等式组的诸多问题的统一的基础算法.

线性规划行旋转算法是线性不等式组行旋转算法的自然扩展, 它将线性规划视为满足最优性条件的线性不等式组, 从而实现了 Dantzig 从行几何出发运用求解线性不等式组的一般方法求解线性规划问题的哲学思考.

3 理论基础

文献 [19] 中包含关于线性规划行旋转算法所有理论的完整证明. 本节将不加证明地给出这些基础理论, 为理解线性规划的改进行旋转算法提供逻辑基础.

Farkas 引理是行旋转算法理论的起点, 该定理表述如下:

定理 3.1 (Farkas 引理) 假设 y_i ($i = 1, 2, \dots, m$) 是定义在实数域上的未知标量, 则以下两组不等式有且仅有一组有解:

$$\begin{aligned} \mathbf{a}_i \mathbf{x} &\geq 0, \quad i = 1, 2, \dots, m, \\ \mathbf{c} \mathbf{x} &< 0 \end{aligned}$$

和

$$\begin{aligned} y_1 \mathbf{a}_1 + y_2 \mathbf{a}_2 + \dots + y_m \mathbf{a}_m &= \mathbf{c}, \\ y_1, y_2, \dots, y_m &\geq 0. \end{aligned}$$

Farkas 引理的结论可以推广到包含线性方程的线性不等式组的情形, 相应推论表述如下:

推论 3.1 λ_i ($i = 1, 2, \dots, m$) 是定义在实数域上的未知标量, 假设方程组 $\mathbf{a}_i \mathbf{x} = b_i$ ($i = 1, 2, \dots, m$) 是相容的且 $\bar{\mathbf{x}}$ 是该方程组的解, 则以下两组线性不等式有且仅有一组有解:

$$\begin{aligned} \mathbf{a}_i \mathbf{x} &= b_i, \quad i = 1, 2, \dots, l, \\ \mathbf{a}_i \mathbf{x} &\geq b_i, \quad i = l+1, l+2, \dots, m, \\ \mathbf{c} \mathbf{x} &< \mathbf{c} \bar{\mathbf{x}} \end{aligned}$$

和

$$\begin{aligned} \lambda_1 \mathbf{a}_1 + \lambda_2 \mathbf{a}_2 + \dots + \lambda_l \mathbf{a}_l + \lambda_{l+1} \mathbf{a}_{l+1} + \dots + \lambda_m \mathbf{a}_m &= \mathbf{c}, \\ \lambda_{l+1}, \lambda_{l+2}, \dots, \lambda_m &\geq 0. \end{aligned}$$

推论 3.1 可以进一步推导出线性规划问题最优性条件, 相应表述如下:

定理 3.2 (最优性条件) 假设 $\mathbf{x}^*$ 是模型 (2.1) 的一个可行解. 如果

$$\mathbf{c} = \sum_{i \in \beta_1 \cup v_1} \lambda_i \mathbf{a}_i + \sum_{i \in \alpha} \lambda_i \mathbf{a}_i, \quad \lambda_i \geq 0, \quad i \in \alpha,$$

其中 $\alpha = \{i \mid \mathbf{a}_i \mathbf{x}^* = b_i, i \in \{l+1, l+2, \dots, m\}\}$, 则 $\mathbf{x}^*$ 是模型 (2.1) 的一个最优解.

下面的推论进一步给出了基本可行解与最优基本可行解之间的关系:

推论 3.2 对于某个给定的基 B , 假设 x^* 是模型 (2.1) 相应的基本可行解. 如果

$$c = \sum_{i \in \beta} \lambda_i a_i, \quad \lambda_i \geq 0, \quad i \in \beta_2,$$

则 x^* 是模型 (2.1) 的最优基本可行解.

定理 3.2 和推论 3.2 就是行旋转算法实现基本解始终满足最优性条件的理论基础. 行旋转算法的基本原理是保证最优性条件成立的前提下从非可行基本解通过迭代逐渐达到可行基本解或获得无可行解的信息, 因此行旋转算法是从非可行基开始的. 行旋转算法的这个特征使得该算法可以直接从满足最优性条件的 n 个非零元素处在不同位置的 n 维正负单位行向量直接启动, 原理如下:

$$c = \sum_{j \in \omega_1} c_j e_j + \sum_{j \in \omega_2} (-c_j)(-e_j),$$

其中, $\omega_1 = \{j : c_j \geq 0\}$, $\omega_2 = \{j : c_j < 0\}$, e_j 表示第 j 个元素为 1 的 n 维单位行向量.

在上面的表示中, 所有系数均大于等于 0, 满足最优性条件. 同时, 所有约束条件系数行向量也可以由这组基向量直接表示出来:

$$a_i = \sum_{j \in \omega_1} a_{ij} e_j + \sum_{j \in \omega_2} (-a_{ij})(-e_j), \quad i \in \{1, 2, \dots, m\}.$$

显然, 正负单位行向量对应关于变量 x_i ($i \in \{1, 2, \dots, n\}$) 的不等式约束. 如果约束条件中存在这样的约束, 则直接将对应行向量作为基向量; 如果约束条件中不存在这样的约束, 则引入关于相应变量的人工不等式. 后面将看到行旋转算法可以非常方便地解决变量上下界问题, 所以在初始算表中只有变量约束之外的其他约束作为显性表示的非基不等式. 从初始基出发, 根据适当的迭代规则, 能够保证向更好的基迭代, 直至找到一个可行基或获得没有可行解的信息为止.

在实际计算模型 (2.1) 的过程中, 根据确定初始基的需要, 考虑是否需要在约束条件中加入人工不等式. 这种确定初始基的方法既没有改变原模型 (2.1), 也没有增加问题的计算规模, 却带来了很多好处: (1) 直接启动算法, 不需要像其他旋转算法需要经过第一阶段才能真正启动算法; (2) 可以直接计算出初始基本解; (3) 直接保证最优性条件成立, 并依据适当的迭代规则保证最优性条件始终成立; (4) 无论 $\text{rank}(A) = k$ 是否小于 n , 行旋转算法启动后, 在每次迭代过程中, 向量 c 和非基约束对应行向量 a_i 均能由当前基中 n 个基向量唯一线性表示; (5) 如果需要, 行旋转算法可以随时计算出当前基下对应的基本解. 在本文后面的改进行旋转算法计算步骤部分和算例演示部分中将具体展现这些特征.

下面的定理建立了可行解与最优解以及基本可行解与最优基本可行解之间的关系.

定理 3.3 (线性规划基本定理) 给定线性规划标准型 (2.1).

- (1) 如果存在一个可行解, 则存在一个基本可行解;
- (2) 如果存在一个最优解, 则存在一个最优基本可行解.

定理 3.3 与单纯形法的基本定理看起来很相似, 但是二者之间有很大的不同. 一是行旋转算法对基和基本解的定义与单纯形法不同, 行旋转算法基于行向量确定基和基本解, 单纯形法则基于列向量; 二是对行旋转算法而言, 定理 3.3 不仅是求解线性不等式组的基本定理, 而且是求解线性规划的基本定理, 单纯形法的基本定理仅针对线性规划问题. 由于线性规划的行旋转算法保证最优性条件始终成立, 因此该算法本质上就是检验约束条件所对应线性不等式组是否存在基本可行解. 一旦找到了一个基本可行解, 这个满足最优性条件的基本可行解当然是最优基本可行解.

模型 (2.1) 基数目的最大可能值为

$$\binom{m-l}{k-l} = \frac{(m-l)!}{(k-l)!(m-k)!}.$$

考虑到行旋转算法可以自动识别并删除冗余 (不) 等式, 这个数目还可能进一步减少. 由于不需要加入任何辅助变量, 所以行旋转算法下模型 (2.1) 中基数目的最大可能值远小于单纯形法下模型 (2.1) 中基数目的最大可能值. 模型 (2.1) 规模越大, 行旋转算法的这种优势越显著. 一般情形下, 行旋转算法在迭代过程中会经过其中很小一部分基到达一个基本可行解或获得无最优解的信息.

如果当前基是不可行基, 则必然存在当前基本解无法满足某些或某个 (不) 等式的情形, 需要进行基变换, 下面的定理处理存在违反不等式的情形.

定理 3.4 对于任意一个给定的基 B , 假设 $\mathbf{x}^1$ 是模型 (2.1) 相应的基本解, 并假设

$$\mathbf{c} = \sum_{j \in \beta} w_{0j} \mathbf{a}_j,$$

其中, $w_{0j} \geq 0, j \in \beta_2$. 假设对于某个 $r \in v_2$, 有 $\mathbf{a}_r \mathbf{x}^1 < b_r$ 且 $\mathbf{a}_r = \sum_{j \in \beta} w_{rj} \mathbf{a}_j$.

(1) 如果对于所有 $j \in \beta_2$ 都有 $w_{rj} \leq 0$, 则模型 (2.1) 没有可行解;

(2) 如果存在 $j \in \beta_2$ 使得 $w_{rj} > 0$ 且

$$\frac{w_{0s}}{w_{rs}} = \min \left\{ \frac{w_{0j}}{w_{rj}} : w_{rj} > 0, j \in \beta_2 \right\},$$

则

$$\mathbf{c} = \left(\frac{w_{0s}}{w_{rs}} \right) \mathbf{a}_r + \sum_{j \in \beta \setminus \{s\}} \left(w_{0j} - \left(\frac{w_{0s}}{w_{rs}} \right) w_{rj} \right) \mathbf{a}_j,$$

其中, $w_{0j} - \left(\frac{w_{0s}}{w_{rs}} \right) w_{rj} \geq 0, j \in \beta_2 \setminus \{s\}$ 且 $\mathbf{c} \mathbf{x}^2 - \mathbf{c} \mathbf{x}^1 = \frac{w_{0s}}{w_{rs}} (b_r - \mathbf{a}_r \mathbf{x}^1) \geq 0$, 这里, $\mathbf{x}^2$ 是线性方程组 $\mathbf{a}_i \mathbf{x} = b_i$ ($i \in \{r\} \cup \beta \setminus \{s\}$) 的解.

(3) 如果存在某个 $s \in \beta_2$ 使得 $w_{rs} > 0$ 且对于任意 $j \in \beta_2 \setminus \{s\}$ 都有 $w_{rj} \leq 0$, 则 $\mathbf{a}_s \mathbf{x} \geq b_s$ 是模型 (2.1) 的一个冗余不等式.

定理 3.4(1) 是矛盾不等式出现的判定条件; 定理 3.4(2) 进行正常迭代; 定理 3.4(3) 是定理 3.4(2) 的特例, 是冗余不等式的判定条件. 由于行旋转算法是从可行域外搜索基本可行解的, 定理 3.4(2) 表明在达到可行基本解之前目标函数值是单调非递减的.

定理 3.4(3) 用于判断基不等式是冗余不等式, 下面的推论用于判断非基不等式是冗余不等式.

推论 3.3 对于任意一个给定的基 B , 假设 $\mathbf{x}^1$ 是模型 (2.1) 相应的基本解, 假设存在 $r \in v_2$ 使得 $\mathbf{a}_r \mathbf{x}^1 \geq b_r$ 且 $\mathbf{a}_r = \sum_{j \in \beta} w_{rj} \mathbf{a}_j$. 如果对于所有 $j \in \beta_2$ 都有 $w_{rj} \geq 0$, 则 $\mathbf{a}_r \mathbf{x} \geq b_r$ 是模型 (2.1) 的一个冗余不等式.

接下来的定理处理在当前基下存在违反等式的情形.

定理 3.5 对于任意一个给定的基 B , 假设 $\mathbf{x}^1$ 是模型 (2.1) 相应的基本解, 并假设

$$\mathbf{c} = \sum_{j \in \beta} w_{0j} \mathbf{a}_j,$$

其中, $w_{0j} \geq 0, j \in \beta_2$. 对于某个 $r \in v_1$, 假设

$$\mathbf{a}_r = \sum_{j \in \beta} w_{rj} \mathbf{a}_j.$$

(1) 如果 $\mathbf{a}_r \mathbf{x}^1 < b_r$ 且对于所有 $j \in \beta_2$ 都有 $w_{rj} \leq 0$, 或者 $\mathbf{a}_r \mathbf{x}^1 > b_r$ 且对于所有 $j \in \beta_2$ 都有 $w_{rj} \geq 0$, 则模型 (2.1) 没有可行解;

(2) 如果 $\mathbf{a}_r \mathbf{x}^1 \leq b_r$ 且

$$\frac{w_{0s}}{w_{rs}} = \min \left\{ \frac{w_{0j}}{w_{rj}} : w_{rj} > 0, j \in \beta_2 \right\}$$

或者 $\mathbf{a}_r \mathbf{x}^1 \geq b_r$ 且

$$\frac{w_{0s}}{w_{rs}} = \max \left\{ \frac{w_{0j}}{w_{rj}} : w_{rj} < 0, j \in \beta_2 \right\},$$

则

$$\mathbf{c} = \left(\frac{w_{0s}}{w_{rs}} \right) \mathbf{a}_r + \sum_{j \in \beta \setminus \{s\}} \left(w_{0j} - \left(\frac{w_{0s}}{w_{rs}} \right) w_{rj} \right) \mathbf{a}_j,$$

其中, $w_{0j} - \left(\frac{w_{0s}}{w_{rs}} \right) w_{rj} \geq 0, j \in \beta \setminus \{s\}$ 且

$$\mathbf{c} \mathbf{x}^2 - \mathbf{c} \mathbf{x}^1 = \frac{w_{0s}}{w_{rs}} (b_r - \mathbf{a}_r \mathbf{x}^1) \geq 0,$$

其中, $\mathbf{x}^2$ 是线性方程组 $\mathbf{a}_i \mathbf{x} = b_i$ ($i \in \{r\} \cup \beta \setminus \{s\}$) 的一个解.

(3) 如果 $\mathbf{a}_r \mathbf{x}^1 = b_r$ 且对于所有 $j \in \beta_2$ 都有 $w_{rj} = 0$, 则 $\mathbf{a}_r \mathbf{x} = b_r$ 是模型 (2.1) 的一个冗余等式.

定理 3.5(1) 给出了判定矛盾等式的条件; 定理 3.5(2) 给出了对违反等式进行正常迭代的规则; 定理 3.5(3) 给出了判定冗余等式的条件. 根据定理 3.5(2) 进行违反等式的正常迭代, 模型 (2.1) 的目标函数值单调非递减.

4 行旋转运算公式

行旋转运算是指将一个非基行向量替换原基中的一个基行向量, 形成一个新的基行向量组, 并将非基行向量表示为新基向量组的线性组合的过程. 这两个向量分别称为出基行向量和入基行向量. 行旋转运算是线性规划行旋转算法的基础运算, 下面将展示行旋转运算的运算规则.

4.1 全表格形式的行旋转运算公式

文献 [19] 根据线性规划的行旋转算法的相关理论推导了完成一次基变换的全表格行旋转运算公式, 这里直接引用推导结果. 表 1 是当前算表, 表 2 是非基向量 $\mathbf{a}_r$ 变成基向量且基向量 $\mathbf{a}_s$ 变成非基向量 (简记为基变换 $\mathbf{a}_r \leftrightarrow \mathbf{a}_s$) 后的算表.

表 2 中的系数解释如下:

$$w'_{0j} = w_{0j} - \left(\frac{w_{0s}}{w_{rs}} \right) w_{rj}, \quad j \in \beta \setminus \{s\},$$

表 1 当前算表

	$\mathbf{a}_s$	$\mathbf{a}_j$	
$\mathbf{c}$	w_{0s}	w_{0j}	
$\mathbf{a}_r$	w_{rs}	w_{rj}	σ_r
$\mathbf{a}_i$	w_{is}	w_{ij}	σ_i

表 2 完成基变换 $a_r \leftrightarrow a_s$ 后的算表

	a_r	a_j	
c	$\frac{w_{0s}}{w_{rs}}$	w'_{0j}	
a_s	$\frac{1}{w_{rs}}$	$-\frac{w_{rj}}{w_{rs}}$	$-\frac{\sigma_r}{w_{rs}}$
a_i	$\frac{w_{is}}{w_{rs}}$	w'_{ij}	σ'_i

$$w'_{ij} = w_{ij} - \left(\frac{w_{is}}{w_{rs}} \right) w_{rj}, \quad i \in v \setminus \{r\}, \quad j \in \beta \setminus \{s\},$$

$$\sigma'_i = \sigma_i - \left(\frac{w_{is}}{w_{rs}} \right) \sigma_r, \quad i \in v \setminus \{r\}.$$

全表格行旋转运算公式需要计算表中的所有元素. 从矩阵运算的角度看行旋转运算, 能够更好地揭示其本质. 下面的定理给出了基于分块矩阵的行旋转运算公式.

4.2 分块矩阵形式的行旋转运算公式

定理 4.1 (分块矩阵的行旋转运算) 设一个 $m \times n$ 矩阵的分块矩阵为

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix},$$

其中 A_{11} 为 k 阶可逆方阵. 如果沿着 A_{11} 的主对角线元素进行 k 次行旋转运算, 则进行这 k 次行旋转运算后得到的矩阵为

$$\begin{pmatrix} A_{11}^{-1} & -A_{11}^{-1}A_{12} \\ A_{21}A_{11}^{-1} & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{pmatrix}.$$

证明 用 a_i 表示原矩阵的第 i 行, e_i 表示 n 阶单位矩阵的第 i 行, 则

$$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \end{pmatrix} = A_{11} \begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_k \end{pmatrix} + A_{12} \begin{pmatrix} e_{k+1} \\ e_{k+2} \\ \vdots \\ e_n \end{pmatrix}.$$

将上式两边左乘 A_{11}^{-1} 并整理得

$$\begin{pmatrix} e_1 \\ e_2 \\ \vdots \\ e_k \end{pmatrix} = A_{11}^{-1} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_k \end{pmatrix} - A_{11}^{-1}A_{12} \begin{pmatrix} e_{k+1} \\ e_{k+2} \\ \vdots \\ e_n \end{pmatrix}.$$

原矩阵的后 $m - k$ 行可以表示为

$$\begin{aligned}
 \begin{pmatrix} \mathbf{a}_{k+1} \\ \mathbf{a}_{k+2} \\ \vdots \\ \mathbf{a}_m \end{pmatrix} &= \mathbf{A}_{21} \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \vdots \\ \mathbf{e}_k \end{pmatrix} + \mathbf{A}_{22} \begin{pmatrix} \mathbf{e}_{k+1} \\ \mathbf{e}_{k+2} \\ \vdots \\ \mathbf{e}_n \end{pmatrix} \\
 &= \mathbf{A}_{21} \left[\mathbf{A}_{11}^{-1} \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_k \end{pmatrix} - \mathbf{A}_{11}^{-1} \mathbf{A}_{12} \begin{pmatrix} \mathbf{e}_{k+1} \\ \mathbf{e}_{k+2} \\ \vdots \\ \mathbf{e}_n \end{pmatrix} \right] + \mathbf{A}_{22} \begin{pmatrix} \mathbf{e}_{k+1} \\ \mathbf{e}_{k+2} \\ \vdots \\ \mathbf{e}_n \end{pmatrix} \\
 &= \mathbf{A}_{21} \mathbf{A}_{11}^{-1} \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_k \end{pmatrix} + (\mathbf{A}_{22} - \mathbf{A}_{21} \mathbf{A}_{11}^{-1} \mathbf{A}_{12}) \begin{pmatrix} \mathbf{e}_{k+1} \\ \mathbf{e}_{k+2} \\ \vdots \\ \mathbf{e}_n \end{pmatrix}.
 \end{aligned}$$

这样经过 k 次行旋转运算后, 前 k 行的两个子块变为 $\mathbf{A}_{11}^{-1}$ 和 $-\mathbf{A}_{11}^{-1} \mathbf{A}_{12}$, 后 $m - k$ 行对应的子块变为 $\mathbf{A}_{21} \mathbf{A}_{11}^{-1}$ 和 $\mathbf{A}_{22} - \mathbf{A}_{21} \mathbf{A}_{11}^{-1} \mathbf{A}_{12}$. $\square$

如果将这个定理的结果表述为表格形式, 则更容易直观地反映行旋转运算的本质.

对比表 3 和 4 可以得到这样的启示: 由于初始表信息已知, 旋转运算并不需要重新计算所有系数, 只需要记录迭代到当前算表相应的特征逆矩阵 $\mathbf{A}_{11}^{-1}$ 就可以随时计算出当前表中所有其他元素. 将这个启示推广到线性规划问题的行旋转算法, 就得到改进的行旋转算法.

文献 [19] 已经说明, 在保证最优性条件始终成立的前提下, 线性规划的行旋转算法总能从一组 n 个非零元素处在不同位置的正负单位行向量构成的基直接启动. 为了说明问题的方便, 不妨假设初始表表 5 中的 $\mathbf{e}'_1, \dots, \mathbf{e}'_k$ 是基向量, $\mathbf{a}_1, \dots, \mathbf{a}_k$ 是非基向量, 相应地将表 5 中的系数矩阵进行划分并用矩阵符号表示. 其中, $\mathbf{e}'_i = \mathbf{e}_i$ 或 $\mathbf{e}'_i = -\mathbf{e}_i$ ($i \in \{1, 2, \dots, n\}$), $\mathbf{c}_1$ 和 $\mathbf{c}_2$ 分别表示在用当前基向量组表示成本向量 $\mathbf{c}$ 时前 k 个系数和后 $n - k$ 个系数构成的行向量, $\boldsymbol{\sigma}_1$ 和 $\boldsymbol{\sigma}_2$ 分别表示在当前基本解下前 k 个等式 (或不等式) 和后 $m - k$ 个等式 (或不等式) 对应偏差构成的列向量, z_0 是当前基本解对应的目标函数值.

表 3 分块矩阵形式的初始表

	$\mathbf{e}_1, \dots, \mathbf{e}_k$	$\mathbf{e}_{k+1}, \dots, \mathbf{e}_n$
$\mathbf{a}_1$		
$\vdots$	$\mathbf{A}_{11}$	$\mathbf{A}_{12}$
$\mathbf{a}_k$		
$\mathbf{a}_{k+1}$		
$\vdots$	$\mathbf{A}_{21}$	$\mathbf{A}_{22}$
$\mathbf{a}_m$		

表 4 沿主对角线完成 k 次行旋转运算后的算表

	$a_1, \dots, a_k$	$e_{k+1}, \dots, e_n$
e_1		
$\vdots$	A_{11}^{-1}	$-A_{11}^{-1}A_{12}$
e_k		
a_{k+1}		
$\vdots$	$A_{21}A_{11}^{-1}$	$A_{22} - A_{21}A_{11}^{-1}A_{12}$
a_m		

表 5 改进行旋转算法的分块矩阵形式的初始表

	$e'_1, \dots, e'_k$	$e'_{k+1}, \dots, e'_n$	
c	c_1	c_2	z_0
a_1			
$\vdots$	A_{11}	A_{12}	σ_1
a_k			
a_{k+1}			
$\vdots$	A_{21}	A_{22}	σ_2
a_m			

运用定理 4.1 介绍的分块矩阵行旋转运算公式, 容易得到经过 k 次行旋转运算后的结果 (见表 6). 表 5 和 6 直观地展示了线性规划的改进行旋转算法的基本原理. 在改进的行旋转算法中, 只要 A_{11}^{-1} 已知, 其他元素在需要时可以运用 A_{11}^{-1} 和初始表中相应信息随时计算出来, 因此该算法的主要工作是计算 A_{11}^{-1} . 在实际运算中, 旋转次序常常不是按下标次序进行的, 但是向量下标是人为规定的, 总可以按照实际行旋转运算次序对向量下标重新排序, 形成前 k 个 a_i 依次替代前 k 个 e'_i 的情形. 因此, 计算 A_{11}^{-1} 的关键是如实地反映迭代到当前基向量组之前的所有基变换的历史信息. 第 5 节将提出能够有效地计算 A_{11}^{-1} 的迭代方法.

4.3 其他公式

在一些情形下, 模型 (2.1) 中的不等式约束有上下界限制. 行旋转算法可以方便有效地处理不等式

表 6 改进行旋转算法的分块矩阵形式的当前表

	$a_1, \dots, a_k$	$e'_{k+1}, \dots, e'_n$	
c	$c_1 A_{11}^{-1}$	$c_2 - c_1 A_{11}^{-1} A_{12}$	$z_0 - c_1 A_{11}^{-1} \sigma_1$
e'_1			
$\vdots$	A_{11}^{-1}	$-A_{11}^{-1} A_{12}$	$-A_{11}^{-1} \sigma_1$
e'_k			
a_{k+1}			
$\vdots$	$A_{21} A_{11}^{-1}$	$A_{22} - A_{21} A_{11}^{-1} A_{12}$	$\sigma_2 - A_{21} A_{11}^{-1} \sigma_1$
a_m			

的上下界, 其计算量与不等式仅含有上界或下界相似. 不失一般性, 假设 $u_i \geq \mathbf{a}_i \mathbf{x} \geq l_i$ 是模型 (2.1) 中具有上下界的约束不等式, 其中 u_i 和 l_i 分别表示该不等式的上界和下界. 为了表述的方便, 称 $-\mathbf{a}_i \mathbf{x} \geq -u_i$ 和 $\mathbf{a}_i \mathbf{x} \geq l_i$ 分别为上界不等式和下界不等式, 它们对应行向量分别为 $-\mathbf{a}_i$ 和 $\mathbf{a}_i$. 考虑 $-\mathbf{a}_i$ 或 $\mathbf{a}_i$ 为基向量时的情形: 如果 $-\mathbf{a}_i$ 为基向量, 即 $-\mathbf{a}_i \mathbf{x} = -u_i$, 则 $\mathbf{a}_i \mathbf{x} \geq l_i$ 一定成立; 如果 $\mathbf{a}_i$ 为基向量, 即 $\mathbf{a}_i \mathbf{x} = l_i$, 则 $-\mathbf{a}_i \mathbf{x} \geq -u_i$ 一定成立. 因此, $-\mathbf{a}_i \mathbf{x} \geq -u_i$ 和 $\mathbf{a}_i \mathbf{x} \geq l_i$ 中有一个为基不等式, 另一个一定成立不必考虑. 再考虑 $\mathbf{a}_i$ 和 $-\mathbf{a}_i$ 同时为非基向量时的情形: 如果已知 $\sigma_{l_i} = \mathbf{a}_i \mathbf{x} - l_i$, 则 $\sigma_{u_i} = -\mathbf{a}_i \mathbf{x} + u_i = -(\sigma_{l_i} + l_i) + u_i = u_i - l_i - \sigma_{l_i}$; 如果已知 $\sigma_{u_i} = -\mathbf{a}_i \mathbf{x} + u_i$, 则 $\sigma_{l_i} = \mathbf{a}_i \mathbf{x} - l_i = u_i - l_i - \sigma_{u_i}$. 因此, 同为非基不等式的上下界不等式之间偏差有如下关系:

$$\sigma_{l_i} + \sigma_{u_i} = u_i - l_i. \quad (4.1)$$

显然, 当非基向量 $\mathbf{a}_i$ 和 $-\mathbf{a}_i$ 由同一组基向量线性表示时, 它们的表示系数必然大小相同, 正负相反. 因此, 当上下界不等式同为非基不等式时, 已知其中一个不等式的信息, 另一个不等式的信息完全已知, 这样只需要显性地表示其中一个不等式.

这样在任何情形下, 只需要显性表达上下界不等式中的一个即可. 变量上下界只是含有上下界的不等式的特殊情形, 因此也可以与一般的上下界不等式采用相同的处理方法.

行旋转算法总是从 n 个非零元素处在不同位置的 n 维正负单位行向量作为初始基向量组启动. n 维单位行向量对应变量的系数行向量. 不失一般性, 不妨设这 n 个初始基向量对应的基不等式为

$$x_i \geq l_i \quad \text{或} \quad -x_i \geq -u_i, \quad i \in \{1, 2, \dots, n\}.$$

算法一旦启动, 可以随时根据需要运用以下公式计算当前基对应的基本解:

$$x_i = l_i + \sigma_{l_i} \quad \text{或} \quad x_i = u_i - \sigma_{u_i}, \quad i \in \{1, 2, \dots, n\}. \quad (4.2)$$

需要指出的是, 基不等式在当前基本解下偏差为 0.

5 算法步骤

本节的算法步骤分为构造初始表、预处理和主迭代三个阶段. 预处理是将非基等式转换为基等式的过程. 由于线性规划问题若有可行解, 必须满足所有等式约束, 因此非基等式对应向量一旦入基就不再换出. 主迭代则是将偏差为负的非基不等式与基不等式交换, 一直到所有非基不等式偏差非负或发现没有可行解为止. 主迭代采用最大负偏差规则确定入基向量, 用最小比值规则确定出基向量. 由于非基等式一旦变为基等式后就不再换出, 因此最小比值规则不必考虑基等式对应向量所在列的元素. 最小比值规则确保最优性条件始终成立, 最大负偏差规则保证目标函数值不断向最优值接近. 对于迭代规则更深入的阐述, 可参见文献 [19].

线性规划的改进行旋转算法的计算步骤如下.

步骤 1 构造初始表.

如果 $c_j \geq 0$, 以 $x_j \geq l_j$ (或引入人工不等式 $x_j \geq -M$, M 为充分大的正数) 为初始基本不等式, $\mathbf{e}_j$ 是初始基向量, 初始基本解中 $x_j^0 = l_j$ (或 $x_j^0 = -M$); 如果 $c_j < 0$, 以 $-x_j \geq -u_j$ (或 $-x_j \geq -M$) 为初始基本不等式, $-\mathbf{e}_j$ 是初始基向量, 初始基本解中 $x_j^0 = u_j$ (或 $x_j^0 = M$). 由此可建立初始表 (见表 7).

在表 7 中, $\mathbf{e}'_j = \mathbf{e}_j$ 或 $\mathbf{e}'_j = -\mathbf{e}_j$ ($j = 1, 2, \dots, n$) 是初始基向量. 第 j 列的元素是向量 $\mathbf{c}$ 和 $\mathbf{a}_i$ ($i = 1, 2, \dots, m$) 关于当前基向量组线性表示时对应第 j 个基向量的表示系数. 具体而言, 当 $c_j \geq 0$ 时,

表 7 初始表

	e'_1	e'_2	$\cdots$	e'_n	
c	c'_1	c'_2	$\cdots$	c'_n	z_0
a_1	a'_{11}	a'_{12}	$\cdots$	a'_{1n}	σ_1
a_2	a'_{21}	a'_{22}	$\cdots$	a'_{2n}	σ_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
a_m	a'_{m1}	a'_{m2}	$\cdots$	a'_{mn}	σ_m

$e'_j = e_j, c'_j = c_j, a'_{ij} = a_{ij}$; 当 $c_j < 0$ 时, $e'_j = -e_j, c'_j = -c_j, a'_{ij} = -a_{ij}$ ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$), 最后一列的 σ_i ($i = 1, 2, \dots, m$) 是将初始基本解代入各非基 (不) 等式得到的偏差.

步骤 2 预处理.

用 w_{0j} 表示表 6 中目标函数行的元素, 用 w_{ij} 表示表 6 中第 i 行第 j 列的元素 ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$), 用 β_2 表示与当前基相应的基不等式的指标集.

对于 $i = 1, 2, \dots, l$, 如果 a_i 对应等式的偏差大于 0, 则将 a_i 行所有元素反号 (相当于将等式约束 $a_i x = b_i$ 换为 $-a_i x = -b_i$) 后转 (1); 当偏差小于 0 时转 (1); 当偏差等于 0 时转 (2).

(1) 若 a_i 行没有与基不等式对应的正元素, 则线性规划没有可行解, 停止; 否则按最小比值规则确定枢轴元素, 即如果

$$\frac{w_{0s}}{w_{is}} = \min \left\{ \frac{w_{0j}}{w_{ij}} : w_{ij} > 0, j \in \beta_2 \right\},$$

则以 w_{is} 为枢轴运用行旋转运算公式计算相应的特征逆矩阵, 并基于该特征逆矩阵计算当前基下相应非基 (不) 等式偏差以及 c 在当前基下的表示系数.

(2) 若 a_i 行与基不等式对应的元素均为 0, 则 $a_i x = b_i$ 冗余; 否则, 当该行有与基不等式对应的正元素时, 按最小比值规则确定枢轴, 运用行旋转运算公式计算相应的特征逆矩阵, 并基于该特征逆矩阵计算当前基下相应非基 (不) 等式偏差以及 c 在当前基下的表示系数.

步骤 3 主迭代.

步骤 3.1 计算当前表非基不等式的偏差.

非基向量 $e'_1, e'_2, \dots, e'_k$ 的偏差构成的向量用 k 维列向量 w_{n+1}^1 表示, 即

$$w_{n+1}^1 = -A_{11}^{-1} \sigma_1;$$

非基向量 $-e'_1, -e'_2, \dots, -e'_k$ 的偏差用 $\bar{w}_{n+1}^1$ 表示, 即

$$\bar{w}_{n+1}^1 = u^1 - l^1 - w_{n+1}^1,$$

其中, u^1 是变量上界 $u_1, u_2, \dots, u_k$ 构成的列向量, l^1 是变量下界 $l_1, l_2, \dots, l_k$ 构成的列向量.

非基向量 $a_{k+1}, a_{k+2}, \dots, a_m$ 的偏差用 w_{n+1}^2 表示, 即

$$w_{n+1}^2 = \sigma_2 - A_{21} A_{11}^{-1} \sigma_1 = \sigma_2 + A_{21} w_{n+1}^1.$$

如果 $w_{n+1}^1, \bar{w}_{n+1}^1$ 和 w_{n+1}^2 的所有分量 (非基向量偏差) 大于或等于 0, 则当前基本解是最优基本可行解, 停止. 否则, 以偏差为负且最小的非基向量入基.

如果 $-e'_j$ ($j \in \{1, 2, \dots, k\}$) 的偏差为负且最小, 则将显性表示的非基向量 e'_j 换为 $-e'_j$, 具体做法如下: 将表 5 中第 j 列的所有元素反号, 然后将此列乘以 $l_j - u_j$ 加到偏差列; 将表 6 中 A_{11}^{-1} 的第 j 行反

号. 所以下面假设最负的分量出现在 $\mathbf{w}_{n+1}^1$ 和 $\mathbf{w}_{n+1}^2$ 中, 即非基向量 $\mathbf{e}'_1, \mathbf{e}'_2, \dots, \mathbf{e}'_k, \mathbf{a}_{k+1}, \mathbf{a}_{k+2}, \dots, \mathbf{a}_m$ 中某个向量的偏差为负且最小, 设其为 $w_{r(n+1)}$ ($r \in \{1, 2, \dots, m\}$).

步骤 3.2 计算当前表基向量的“成本”.

基向量 $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$ 成本用 $\mathbf{w}_0^1$ 表示, 即 $\mathbf{w}_0^1 = \mathbf{c}_1 \mathbf{A}_{11}^{-1}$; 基向量 $\mathbf{e}'_{k+1}, \mathbf{e}'_{k+2}, \dots, \mathbf{e}'_n$ 的成本用 $\mathbf{w}_0^2$ 表示, 即 $\mathbf{w}_0^2 = \mathbf{c}_2 - \mathbf{c}_1 \mathbf{A}_{11}^{-1} \mathbf{A}_{12} = \mathbf{c}_2 - \mathbf{w}_0^1 \mathbf{A}_{12}$.

步骤 3.3 计算枢轴行元素以及基向量成本与枢轴行正元素的比值.

用 $\mathbf{a}_r^1$ 表示表 5 中第 r 行前 k 个元素构成的行向量, 即 $\mathbf{a}_r^1 = [a'_{r1}, a'_{r2}, \dots, a'_{rk}]$, 用 $\mathbf{a}_r^2$ 表示表 5 中第 r 行后 $n-k$ 个元素构成的行向量, 即 $\mathbf{a}_r^2 = [a'_{r(k+1)}, a'_{r(k+2)}, \dots, a'_{rn}]$. 用 $\mathbf{w}_r^1$ 表示表 6 中第 r 行前 k 个元素构成的行向量, 即 $\mathbf{w}_r^1 = [w_{r1}, w_{r2}, \dots, w_{rk}]$, 用 $\mathbf{w}_r^2$ 表示表 6 中第 r 行后 $n-k$ 个元素构成的行向量, 即 $\mathbf{w}_r^2 = [w_{r(k+1)}, w_{r(k+2)}, \dots, w_{rn}]$. 如果枢轴向量 $\mathbf{a}_r$ 由基向量 (不包括基等式对应向量) 唯一线性表示的系数中没有正元素, 则线性规划没有可行解. 否则,

(1) 如果 $r \in \{1, 2, \dots, k\}$, 则 $\mathbf{w}_r^1$ 是 $\mathbf{A}_{11}^{-1}$ 的第 r 行, $\mathbf{w}_r^2 = -\mathbf{w}_r^1 \mathbf{A}_{12}$, 计算 $w_{0s}/w_{rs} = \min\{w_{0j}/w_{rj} : w_{rj} > 0, j \in \beta_2\}$, 其中 β_1 为基等式相应行向量对应的指标集;

(2) 如果 $r \in \{k+1, k+2, \dots, m\}$, 则 $\mathbf{w}_r^1 = \mathbf{a}_r^1 \mathbf{A}_{11}^{-1}$, $\mathbf{w}_r^2 = \mathbf{a}_r^2 - \mathbf{w}_r^1 \mathbf{A}_{12}$, 计算 $w_{0s}/w_{rs} = \min\{w_{0j}/w_{rj} : w_{rj} > 0, j \in \beta_2\}$.

如果枢轴行没有正元素 (枢轴行向量由当前基向量组线性表示时基不等式对应向量的表示系数均小于等于 0), 则线性规划没有可行解. 否则, 以取得最小比值的正元素为枢轴元素, 设其为第 s 列的元素 w_{rs} .

步骤 3.4 更新 $\mathbf{A}_{11}^{-1}$.

(1) 枢轴元素 w_{rs} 是表 6 中 $\mathbf{A}_{11}^{-1}$ 中的元素.

此时 $r, s \in \{1, 2, \dots, k\}$. 在 $\mathbf{A}_{11}^{-1}$ 中以 w_{rs} 为枢轴进行一次旋转运算, 然后删去枢轴行和枢轴列. 令 $k := k - 1$, 转步骤 3.1.

(2) 枢轴元素 w_{rs} 是表 6 中 $-\mathbf{A}_{11}^{-1} \mathbf{A}_{12}$ 中的元素.

此时 $r \in \{1, 2, \dots, k\}$, $s \in \{k+1, k+2, \dots, n\}$. 表 5 中 $\mathbf{A}_{12}$ 的第 $s-k$ 列为 $\mathbf{p}_s = [a'_{1s}, a'_{2s}, \dots, a'_{ks}]^T$. 表 6 中 $-\mathbf{A}_{11}^{-1} \mathbf{A}_{12}$ 的第 $s-k$ 列为 $\mathbf{w}_s = -\mathbf{A}_{11}^{-1} \mathbf{p}_s$. 将 $\mathbf{w}_s$ 附着在 $\mathbf{A}_{11}^{-1}$ 后面, 即构造 $k \times (k+1)$ 矩阵 $[\mathbf{A}_{11}^{-1}, \mathbf{w}_s]$, 并以 $\mathbf{w}_s$ 中的 w_{rs} 为枢轴进行一次旋转运算, 然后删去最后一列. 这样, $\mathbf{e}'_r$ 成为基向量, $\mathbf{e}'_s$ 成为非基向量, 转步骤 3.1.

(3) 枢轴元素 w_{rs} 是表 6 中 $\mathbf{A}_{21} \mathbf{A}_{11}^{-1}$ 中的元素.

此时 $r \in \{k+1, k+2, \dots, m\}$, $s \in \{1, 2, \dots, k\}$. 表 5 中 $\mathbf{A}_{21}$ 的第 $r-k$ 行为 $\mathbf{q}_r = [a'_{r1}, a'_{r2}, \dots, a'_{rk}]$. 表 6 中 $\mathbf{A}_{21} \mathbf{A}_{11}^{-1}$ 的第 $r-k$ 行为 $\mathbf{w}_r = \mathbf{q}_r \mathbf{A}_{11}^{-1}$. 将 $\mathbf{w}_r$ 附着在 $\mathbf{A}_{11}^{-1}$ 下面, 即构造 $(k+1) \times k$ 矩阵

$$\begin{bmatrix} \mathbf{A}_{11}^{-1} \\ \mathbf{w}_r \end{bmatrix},$$

并以 $\mathbf{w}_r$ 中的 w_{rs} 为枢轴进行一次旋转运算, 然后删去最后一行. 这样, $\mathbf{a}_r$ 成为基向量, $\mathbf{a}_s$ 成为非基向量, 转步骤 3.1.

(4) 枢轴元素 w_{rs} 是表 6 中 $\mathbf{A}_{22} - \mathbf{A}_{21} \mathbf{A}_{11}^{-1} \mathbf{A}_{12}$ 中的元素.

此时 $r \in \{k+1, k+2, \dots, m\}$, $s \in \{k+1, k+2, \dots, n\}$. 构造 $(k+1) \times (k+1)$ 矩阵

$$\begin{bmatrix} \mathbf{A}_{11}^{-1} & \mathbf{w}_s \\ \mathbf{w}_r & w_{rs} \end{bmatrix},$$

其中, w_s 和 w_r 的计算见 (2) 和 (3), $w_{rs} = a'_{rs} - w_r p_s$, a'_{rs} 为表 5 中第 r 行第 s 列的元素. 这样, a_r 成为基向量, e'_s 成为非基向量. 令 $k := k + 1$, 转步骤 3.1.

改进行旋转算法计算步骤看起来复杂, 实际上思路非常简单. 首先将所有非基等式变成基等式. 如果存在矛盾等式, 则线性规划问题没有可行解; 如果存在冗余等式, 则消除冗余等式. 一旦非基等式全部变为基等式后, 则所对应向量不再出基. 接下来进入主迭代阶段, 在偏差为负且最小的非基不等式与基不等式之间进行基变换, 直到找到最优基本可行解或获得无可行解信息为止.

一般而言, 完成一次行旋转运算分为以下几个步骤: (1) 计算非基向量对应的偏差, 以偏差为负且最小的非基向量入基, 确定枢轴行; (2) 计算基不等式的成本, 即向量 c 关于当前基向量组的线性表示系数; (3) 计算枢轴行上与基不等式对应的元素, 并按最小比值规则确定枢轴元素; (4) 根据枢轴元素的位置更新特征逆矩阵. 因此, 每一次旋转运算过程中必须计算的元素是在当前基下非基 (不) 等式偏差、枢轴行和枢轴列部分元素、成本向量元素. 所有这些必要计算的元素, 都可以随时通过当前特征逆矩阵 A_{11}^{-1} 和模型的初始信息计算出来. 改进行旋转算法编程的关键在于正确确定 A_{11}^{-1} 和枢轴元素 w_{rs} 的位置关系, 从而能够正确地运用旋转运算更新特征逆矩阵 A_{11}^{-1} .

6 算例

考虑如下线性规划问题:

$$\begin{aligned} \min z &= 9x_1 + x_2 - x_3 + x_4 + 2x_5 - 2x_6 + 13x_7 + 6x_8 \\ \text{s.t. } x_1 + 4x_2 + 2x_3 + x_4 - 4x_5 + 9x_6 - 3x_7 - 5x_8 &= -28, \\ 2x_1 + 11x_2 - x_3 + 2x_4 - 2x_5 + 12x_6 + 3x_7 - 7x_8 &= -16, \\ x_1 + 7x_2 - 3x_3 + x_4 + 2x_5 + 3x_6 + 6x_7 - 2x_8 &= 12, \\ x_1 - 2x_2 - x_3 + x_4 + 7x_5 - 2x_6 + 4x_7 + 7x_8 &\geq -1, \\ 2x_1 + x_2 - 3x_3 - x_4 + 12x_5 + 2x_6 - x_7 + 6x_8 &\geq 47, \\ 2x_1 + x_2 + x_3 + 2x_4 + 2x_5 - 2x_6 + 7x_7 - 8x_8 &\geq 1, \\ 4x_1 + 6x_2 - x_3 + 3x_4 - 8x_5 + 3x_6 + 6x_7 + x_8 &\geq 6, \\ 4 \geq x_1 \geq 0, \quad 1 \geq x_2 \geq -1, \quad 2 \geq x_3 \geq 0, \quad 2 \geq x_4 \geq 0, \\ 5 \geq x_5 \geq 0, \quad 1 \geq x_6 \geq -1, \quad 4 \geq x_7 \geq 2, \quad 2 \geq x_8 \geq 0. \end{aligned}$$

利用改进行旋转算法求解以上线性规划问题的完整过程如下.

步骤 1 构建初始表.

目标函数系数为 $c = [9, 1, -1, 1, 2, -2, 13, 6]$, 根据目标函数系数向量各元素的正负情形, 确定初始基不等式组为 $x_1 \geq 0, x_2 \geq -1, -x_3 \geq -2, x_4 \geq 0, x_5 \geq 0, -x_6 \geq -1, x_7 \geq 2, x_8 \geq 0$, 相应初始基向量为

$$\begin{aligned} e_1 &= [1, 0, 0, 0, 0, 0, 0, 0], & e_2 &= [0, 1, 0, 0, 0, 0, 0, 0], \\ -e_3 &= -[0, 0, 1, 0, 0, 0, 0, 0], & e_4 &= [0, 0, 0, 1, 0, 0, 0, 0], \\ e_5 &= [0, 0, 0, 0, 1, 0, 0, 0], & -e_6 &= -[0, 0, 0, 0, 0, 1, 0, 0], \\ e_7 &= [0, 0, 0, 0, 0, 0, 1, 0], & e_8 &= [0, 0, 0, 0, 0, 0, 0, 1], \end{aligned}$$

相应初始基本解为 $\boldsymbol{x}^0 = [0, -1, 2, 0, 0, 1, 2, 0]^T$. 作为基不等式组的上(下)界不等式相应的下(上)界不等式一定成立, 不必考虑. 变量上界和下界向量分别为

$$\boldsymbol{u} = [4, 1, 2, 2, 5, 1, 4, 2]^T \quad \text{和} \quad \boldsymbol{l} = [0, -1, 0, 0, 0, -1, 2, 0]^T.$$

非基(不)等式对应的系数向量分别为

$$\begin{aligned} \boldsymbol{a}_1 &= [1, 4, 2, 1, -4, 9, -3, -5], & \boldsymbol{a}_2 &= [2, 11, -1, 2, -2, 12, 3, -7], \\ \boldsymbol{a}_3 &= [1, 7, -3, 1, 2, 3, 6, -2], & \boldsymbol{a}_4 &= [1, -2, -1, 1, 7, -2, 4, 7], \\ \boldsymbol{a}_5 &= [2, 1, -3, -1, 12, 2, -1, 6], & \boldsymbol{a}_6 &= [2, 1, 1, 2, 2, -2, 7, -8], \\ \boldsymbol{a}_7 &= [4, 6, -1, 3, -8, 3, 6, 1]. \end{aligned}$$

显然, 目标函数系数向量和非基(不)等式相应系数向量可以直接由基向量组线性表示, 将基本解代入各非基(不)等式得到相应偏差. 该线性规划问题的初始表如下:

变量上下界向量和初始表保存了线性规划问题的所有原始信息. 在初始表中, $\boldsymbol{A}_{11}^{-1} = [\cdot]$ 为空矩阵, 所以 $\boldsymbol{A}_{11}^{-1}$ 阶数 $k = 0$.

步骤 2 预处理 (将非基等式变为基等式).

非基向量 $\boldsymbol{a}_1$ 对应的等式在当前基本解下偏差为正, 将 $\boldsymbol{a}_1$ 变为 $-\boldsymbol{a}_1$, 并将对应行所有系数反号, 根据最小比值规则确定枢轴元素 $w_{16} = 9$, 即向量 $-\boldsymbol{a}_1$ 入基, 向量 $-\boldsymbol{e}_6$ 出基 (简记为 $-\boldsymbol{a}_1 \leftrightarrow -\boldsymbol{e}_6$). 由于初始表对应 $\boldsymbol{A}_{11}^{-1}$ 为空矩阵, 所以更新后的 $\boldsymbol{A}_{11}^{-1}$ 仅包含对枢轴 w_{16} 进行旋转运算后对应的元素, 即 $\boldsymbol{A}_{11}^{-1} = [1/9] = [0.11]$, 阶数 $k = 1$. $-\boldsymbol{a}_1$ 变为基向量后, 对应等式变为基等式, 在以后的运算中不再出基. 基变换 $-\boldsymbol{a}_1 \leftrightarrow -\boldsymbol{e}_6$ 确定当前特征逆矩阵 $\boldsymbol{A}_{11}^{-1} = [1/9] = [0.11]$ 后, 就可以根据前面部分提供的公式计算当前基下非基(不)等式偏差、枢轴行元素、成本向量元素, 确定枢轴元素, 以及更新 $\boldsymbol{A}_{11}^{-1}$ 所需要的元素. 计算结果如表 8 所示.

依据当前 $\boldsymbol{A}_{11}^{-1}$ 和相应初始信息计算得到非基(不)等式偏差, 显然当前基本解不是最优基本可行解, 需要进一步迭代. 需要指出的是, 表 8 中第 2 个偏差列记录上下界非基不等式没有显性表示的那个不等式对应的偏差 (-1.44 是 $\boldsymbol{e}_6$ 对应不等式在当前基本解下的偏差). 接下来将第 2 个非基等式变成基等式, 所以 $\boldsymbol{a}_2$ 对应行为枢轴行. 为了确定出基向量, 需要计算当前基下的成本向量元素. 根据最小比值规则, 确定 $w_{23} = 3.67$ 为枢轴元素, 计算出必要的元素以更新 $\boldsymbol{A}_{11}^{-1}$, 通过基变换 $\boldsymbol{a}_2 \leftrightarrow -\boldsymbol{e}_3$ 对应的行旋转运算更新 $\boldsymbol{A}_{11}^{-1}$ 的过程如表 9 所示. 从而得到更新后的特征逆矩阵如下:

表 8 第 1 次基变换 $-\boldsymbol{a}_1 \leftrightarrow -\boldsymbol{e}_6$ 结果

	$\boldsymbol{e}_1$	$\boldsymbol{e}_2$	$-\boldsymbol{e}_3$	$\boldsymbol{e}_4$	$\boldsymbol{e}_5$	$-\boldsymbol{a}_1$	$\boldsymbol{e}_7$	$\boldsymbol{e}_8$		
$\boldsymbol{c}$	9.22	1.89	0.56	1.22	1.11	0.22	12.33	4.89	σ_i	$\bar{\sigma}_i$
$-\boldsymbol{e}_6$			-0.22			0.11			3.44	-1.44
$\boldsymbol{a}_2$	0.67	5.67	[3.67]	0.67	3.33	-1.33	7.00	-0.33	-20.33	
$\boldsymbol{a}_3$									-20.33	
$\boldsymbol{a}_4$									13.89	
$\boldsymbol{a}_5$									-60.89	
$\boldsymbol{a}_6$									5.89	
$\boldsymbol{a}_7$									-9.33	

表 9 第 2 个特征逆矩阵计算过程

	$-\mathbf{a}_1$	$-\mathbf{e}_3$	$\Rightarrow$		$-\mathbf{a}_1$	$\mathbf{a}_2$
$-\mathbf{e}_6$	0.11	-0.22		$-\mathbf{e}_6$	0.03	-0.06
$\mathbf{a}_2$	-1.33	[3.67]		$-\mathbf{e}_3$	0.36	0.27

$$\mathbf{A}_{11}^{-1} = \begin{bmatrix} 0.03 & -0.06 \\ 0.36 & 0.27 \end{bmatrix}, \quad k = 2.$$

依据当前 $\mathbf{A}_{11}^{-1}$ 和相应初始信息计算得到非基 (不) 等式偏差, 可以看到还有多个非基 (不) 等式在当前基本解下偏差为负, 显然当前基本解不是最优基本可行解, 需要进一步迭代. 接下来计算将第 3 个非基等式转化为基等式的相关信息 (见表 10).

从表 10 中可以看出, $\mathbf{a}_3$ 由 $-\mathbf{a}_1$ 和 $\mathbf{a}_2$ 唯一线性表示且对应偏差为 0. 也就是说, 在当前基下该等式为冗余等式, 可以直接删除.

步骤 3 主迭代 (非基不等式与基不等式替换).

删掉 $\mathbf{a}_3$ 对应行后, 非基等式转换为基等式阶段完成, 进入主迭代阶段. 在主迭代阶段, 基等式对应基向量不参与向量替换. 在当前基下, $\mathbf{a}_5$ 对应不等式的偏差为负且最小, $\mathbf{a}_5$ 对应行为枢轴行. 与前面逻辑相同, 需要基于当前 $\mathbf{A}_{11}^{-1}$ 和初始信息计算必要元素, 其结果如表 11 所示.

利用当前 $\mathbf{A}_{11}^{-1}$ 和初始信息可以计算出更新 $\mathbf{A}_{11}^{-1}$ 所需要的元素, 根据最小比值规则, $w_{55} = 9.76$ 为枢轴元素, 相应基变换为 $\mathbf{a}_5 \leftrightarrow \mathbf{e}_5$, 对应地计算新 $\mathbf{A}_{11}^{-1}$ 的矩阵如表 12 所示. 经过基变换 $\mathbf{a}_5 \leftrightarrow \mathbf{e}_5$ 对应的行旋转运算得到更新后的 $\mathbf{A}_{11}^{-1}$ 如下:

$$\mathbf{A}_{11}^{-1} = \begin{bmatrix} 0.06 & -0.04 & -0.02 \\ 0.46 & 0.36 & -0.09 \\ -0.11 & -0.10 & 0.10 \end{bmatrix}, \quad k = 3.$$

从上面更新 $\mathbf{A}_{11}^{-1}$ 的计算过程可以看出, 在用行旋转公式更新 $\mathbf{A}_{11}^{-1}$ 时, 必须保证矩阵行列的排列一定要与向量替换的先后顺序一致. 后面不再具体说明更新 $\mathbf{A}_{11}^{-1}$ 的计算过程, 而直接给出更新后的 $\mathbf{A}_{11}^{-1}$.

经过第 1 次主迭代后, 非基不等式在当前基本解下仍然存在负偏差. 在计算必要的元素后确定新的基变换为 $\mathbf{a}_7 \leftrightarrow \mathbf{e}_8$, 对应地计算新 $\mathbf{A}_{11}^{-1}$ 的矩阵如表 13 所示. 枢轴元素为 9.04, 经过基变换 $\mathbf{a}_7 \leftrightarrow \mathbf{e}_8$

表 10 第 2 次基变换 $\mathbf{a}_2 \leftrightarrow -\mathbf{e}_3$ 结果

	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{a}_2$	$\mathbf{e}_4$	$\mathbf{e}_5$	$-\mathbf{a}_1$	$\mathbf{e}_7$	$\mathbf{e}_8$		
$\mathbf{c}$	9.12	1.03	0.15	1.12	0.61	0.42	11.27	4.94	σ_i	$\bar{\sigma}_i$
$-\mathbf{e}_6$									2.21	-0.21
$-\mathbf{e}_3$									5.55	-3.55
$\mathbf{a}_3$	-0.00	0.00	1.00	-0.00	-0.00	1.00	0.00	0.00	0.00	
$\mathbf{a}_4$									16.97	
$\mathbf{a}_5$									-41.79	
$\mathbf{a}_6$									-2.12	
$\mathbf{a}_7$									-0.09	

表 11 第 1 次主迭代枢轴元素的确定

	e_1	e_2	a_2	e_4	e_5	$-a_1$	e_7	e_8		
c	9.12	1.03	0.15	1.12	0.61	0.42	11.27	4.94	σ_i	$\bar{\sigma}_i$
$-e_6$					-0.24				2.21	-0.21
$-e_3$					-0.91				5.55	-3.55
a_4									16.97	
a_5	1.15	-5.21	0.94	-1.85	[9.76]	1.03	-6.91	7.42	-41.79	
a_6									-2.12	
a_7									-0.09	

表 12 第 1 次主迭代对应特征逆矩阵算表

	$-a_1$	a_2	e_5
$-e_6$	0.03	-0.06	-0.24
$-e_3$	0.36	0.27	-0.91
a_5	1.03	0.94	[9.76]

表 13 第 2 次主迭代对应特征逆矩阵算表

	$-a_1$	a_2	a_5	e_8
$-e_6$	0.06	-0.04	-0.02	-0.39
$-e_3$	0.46	0.36	-0.09	0.78
e_5	-0.11	-0.10	0.10	-0.76
a_7	1.14	1.24	-0.84	[9.04]

对应的行旋转运算得到更新后的 A_{11}^{-1} 如下:

$$A_{11}^{-1} = \begin{bmatrix} 0.11 & 0.02 & -0.06 & -0.04 \\ 0.36 & 0.25 & -0.02 & 0.09 \\ -0.01 & 0.01 & 0.03 & -0.08 \\ -0.13 & -0.14 & 0.09 & 0.11 \end{bmatrix}, \quad k = 4.$$

经过第 2 次主迭代后, 非基不等式在当前基本解下仍然存在负偏差. 在计算必要的元素后确定新的基变换为 $a_6 \leftrightarrow e_4$, 对应地计算新 A_{11}^{-1} 的矩阵如表 14 所示. 枢轴元素为 3.94, 经过基变换 $a_6 \leftrightarrow e_4$

表 14 第 3 次主迭代对应特征逆矩阵算表

	$-a_1$	a_2	a_5	a_7	e_4
$-e_6$	0.11	0.02	-0.06	-0.04	0.14
$-e_3$	0.36	0.25	-0.02	0.09	-0.42
e_5	-0.01	0.01	0.03	-0.08	0.26
e_8	-0.13	-0.14	0.09	0.11	-0.09
a_6	0.83	0.90	-0.78	-1.23	[3.94]

对应的行旋转运算得到更新后的 $\mathbf{A}_{11}^{-1}$ 如下:

$$\mathbf{A}_{11}^{-1} = \begin{bmatrix} 0.08 & -0.02 & -0.03 & 0.00 & 0.04 \\ 0.45 & 0.35 & -0.10 & -0.05 & -0.11 \\ -0.06 & -0.05 & 0.08 & -0.00 & 0.07 \\ -0.11 & -0.12 & 0.07 & 0.08 & -0.02 \\ -0.21 & -0.23 & 0.20 & 0.31 & 0.25 \end{bmatrix}, \quad k = 5.$$

在计算必要的元素后, 非基向量 $-\mathbf{e}_4$ 对应的不等式偏差为负且最小, 确定新的基变换为 $-\mathbf{e}_4 \leftrightarrow \mathbf{e}_1$. 将初始信息表 15 中基变量 $\mathbf{e}_4$ 替换为 $-\mathbf{e}_4$, 将表 15 中对应列所有元素反号, 然后将此列乘以 $l_4 - u_4$ 加到偏差列. 在当前 $\mathbf{A}_{11}^{-1}$ 中将 $\mathbf{e}_4$ 对应系数全部反号, 变为 $-\mathbf{e}_4$ 对应系数, 即 $-\mathbf{e}_4$ 对应不等式显性表示, $\mathbf{e}_4$ 对应不等式隐性表示. 对应地计算新 $\mathbf{A}_{11}^{-1}$ 的矩阵如表 16 所示. 枢轴元素为 1.90, 经过基变换 $-\mathbf{e}_4 \leftrightarrow \mathbf{e}_1$ 对应行旋转运算并删除最后一列, 得到更新后的 $\mathbf{A}_{11}^{-1}$ 如下:

$$\mathbf{A}_{11}^{-1} = \begin{bmatrix} 0.06 & -0.03 & -0.02 & 0.02 & 0.05 \\ 0.41 & 0.31 & -0.07 & 0.01 & -0.06 \\ -0.04 & -0.02 & 0.06 & -0.04 & 0.03 \\ -0.07 & -0.08 & 0.04 & 0.03 & -0.06 \\ -0.11 & -0.12 & 0.10 & 0.16 & 0.13 \end{bmatrix}, \quad k = 5.$$

在计算必要的元素后, 非基向量 $\mathbf{e}_3$ 对应的不等式偏差为负且最小, 确定新的基变换为 $\mathbf{e}_3 \leftrightarrow \mathbf{e}_2$. 将初始信息表 15 中基变量 $-\mathbf{e}_3$ 替换为 $\mathbf{e}_3$, 将表 15 中对应列所有元素反号, 然后将此列乘以 $l_3 - u_3$

表 15 初始表

	$\mathbf{e}_1$	$\mathbf{e}_2$	$-\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$	$-\mathbf{e}_6$	$\mathbf{e}_7$	$\mathbf{e}_8$	
$\mathbf{c}$	9.00	1.00	1.00	1.00	2.00	2.00	13.00	6.00	σ_i
$\mathbf{a}_1$	1.00	4.00	-2.00	1.00	-4.00	-9.00	-3.00	-5.00	31.00
$\mathbf{a}_2$	2.00	11.00	1.00	2.00	-2.00	-12.00	3.00	-7.00	21.00
$\mathbf{a}_3$	1.00	7.00	3.00	1.00	2.00	-3.00	6.00	-2.00	-10.00
$\mathbf{a}_4$	1.00	-2.00	1.00	1.00	7.00	2.00	4.00	7.00	7.00
$\mathbf{a}_5$	2.00	1.00	3.00	-1.00	12.00	-2.00	-1.00	6.00	-54.00
$\mathbf{a}_6$	2.00	1.00	-1.00	2.00	2.00	2.00	7.00	-8.00	-1.00
$\mathbf{a}_7$	4.00	6.00	1.00	3.00	-8.00	-3.00	6.00	1.00	1.00

表 16 第 4 次主迭代对应特征逆矩阵算表

	$-\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_5$	$\mathbf{a}_7$	$\mathbf{a}_6$	$\mathbf{e}_1$
$-\mathbf{e}_6$	0.08	0.02	-0.03	0.00	0.04	0.10
$-\mathbf{e}_3$	0.45	0.35	-0.10	-0.05	-0.11	0.36
$\mathbf{e}_5$	-0.06	-0.05	0.08	-0.00	0.07	-0.24
$\mathbf{e}_8$	-0.11	-0.12	0.07	0.08	-0.02	-0.31
$-\mathbf{e}_4$	0.21	0.23	-0.20	-0.31	-0.25	[1.90]

加到偏差列. 在当前 $\mathbf{A}_{11}^{-1}$ 中将 $-\mathbf{e}_3$ 对应系数全部反号, 变为 $\mathbf{e}_3$ 对应系数, 即 $\mathbf{e}_3$ 对应不等式显性表示, $-\mathbf{e}_3$ 对应不等式隐性表示. 对应地计算新 $\mathbf{A}_{11}^{-1}$ 的矩阵如表 17 所示. 在当前基下确定枢轴元素为 1.68, 经过基变换 $\mathbf{e}_3 \leftrightarrow \mathbf{e}_2$ 对应行旋转运算并删除最后一列, 得到更新后的 $\mathbf{A}_{11}^{-1}$ 如下:

$$\mathbf{A}_{11}^{-1} = \begin{bmatrix} 0.17 & 0.05 & -0.04 & 0.02 & 0.03 \\ 0.25 & 0.18 & -0.04 & 0.01 & -0.04 \\ 0.03 & 0.03 & 0.05 & -0.04 & 0.02 \\ 0.03 & -0.00 & 0.03 & 0.04 & -0.08 \\ -0.20 & -0.18 & 0.12 & 0.16 & 0.15 \end{bmatrix}, \quad k = 5.$$

在计算必要的元素后, 非基向量 $-\mathbf{e}_8$ 对应的不等式偏差为负且最小, 确定新的基变换为 $-\mathbf{e}_8 \leftrightarrow \mathbf{a}_6$. 将初始信息表 15 中基变量 $\mathbf{e}_8$ 替换为 $-\mathbf{e}_8$, 将表 15 中对应列所有元素反号, 然后将此列乘以 $l_8 - u_8$ 加到偏差列. 在当前 $\mathbf{A}_{11}^{-1}$ 中将 $\mathbf{e}_8$ 对应系数全部反号, 变为 $-\mathbf{e}_8$ 对应系数, 即将 $-\mathbf{e}_8$ 对应不等式显性表示, $\mathbf{e}_8$ 对应不等式隐性表示. 对应地计算新 $\mathbf{A}_{11}^{-1}$ 的矩阵如表 18 所示. 在当前基下确定枢轴元素为 0.08, 经过基变换 $-\mathbf{e}_8 \leftrightarrow \mathbf{a}_6$ 对应行旋转运算并删除枢轴行和枢轴列, 得到更新后的 $\mathbf{A}_{11}^{-1}$ 如下:

$$\mathbf{A}_{11}^{-1} = \begin{bmatrix} 0.18 & 0.05 & -0.03 & 0.04 \\ 0.23 & 0.18 & -0.05 & -0.01 \\ 0.03 & 0.02 & 0.06 & -0.03 \\ -0.14 & -0.19 & 0.17 & 0.23 \end{bmatrix}, \quad k = 4.$$

基于当前特征逆矩阵和初始信息, 可以求出所有非基不等式偏差. 所有偏差均为正, 表明当前基为可行基, 当前解为最优基本可行解. 表 19 直观显示了第 6 次主迭代后当前基向量、非基向量和非基不等式对应的偏差.

表 17 第 5 次主迭代对应特征逆矩阵算表

	$-\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_5$	$\mathbf{a}_7$	$\mathbf{a}_6$	$\mathbf{e}_2$
$-\mathbf{e}_6$	0.06	-0.03	-0.02	0.02	0.05	0.43
$\mathbf{e}_3$	-0.41	-0.31	0.07	-0.01	0.06	[1.68]
$\mathbf{e}_5$	-0.04	-0.02	0.06	-0.04	0.03	0.26
$\mathbf{e}_8$	-0.07	-0.08	0.04	0.03	-0.06	0.42
$\mathbf{e}_1$	-0.11	-0.12	0.10	0.16	0.13	-0.35

表 18 第 6 次主迭代对应特征逆矩阵算表

	$-\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_5$	$\mathbf{a}_7$	$\mathbf{a}_6$
$-\mathbf{e}_6$	0.17	0.05	-0.04	0.02	0.03
$\mathbf{e}_2$	0.25	0.18	-0.04	0.01	-0.04
$\mathbf{e}_5$	0.03	0.03	0.05	-0.04	0.02
$-\mathbf{e}_8$	-0.03	0.00	-0.03	-0.04	[0.08]
$\mathbf{e}_1$	-0.20	-0.18	0.12	0.16	0.15

表 19 第 6 次主迭代后非基不等式对应偏差

	$-e_4$	e_3	a_2	$-e_8$	a_5	$-a_1$	e_7	a_7		
c									σ_i	$\bar{\sigma}_i$
$-e_6$									1.33	0.67
e_2									0.12	1.88
a_4									50.48	
e_5									2.78	2.22
e_1									3.62	0.38
a_6									0.58	

表 19 表明当前基本解是基本可行解, 由于在线性规划的行旋转运算中始终保持最优性条件成立, 因而当前基本解就是线性规划问题的最优基本可行解. 运用 (4.2) 可以计算出最优解如下:

$$\begin{aligned} x_1 &= 3.623006, & x_2 &= -0.8821898, & x_3 &= 0, & x_4 &= 2, \\ x_5 &= 2.775121, & x_6 &= -0.3326399, & x_7 &= 2, & x_8 &= 2, \end{aligned}$$

相应的最优目标函数值为 $z^* = 77.94038$.

7 结论

线性规划的行旋转算法是基于行几何 (行向量) 的一种求解线性规划问题的有效算法. 求解线性规划问题可以看作是在保证最优性条件始终成立的前提下求解其约束条件对应的线性不等式组, 因而线性规划问题可以看作特殊的线性不等式组. 线性规划的行旋转算法同时实现了 Dantzig^[11,12] 提出的“求解线性规划问题本质上是求解线性不等式组”和“从行几何求解线性规划问题”的思想, 改变了线性规划问题的旋转算法实际只包含列旋转算法的现状, 丰富了旋转算法的内涵.

线性规划的行旋转算法具有一些显著的计算特色, 但是仍然是一种全表格算法, 有进一步提高计算效率的空间. 通过分析分块矩阵的行旋转运算公式可以发现, 每次旋转运算并不需要计算当前算表中的所有元素, 只需要计算确定入基向量和出基向量的小部分必要元素即可; 只要知道当前算表对应的特征逆矩阵和模型的初始信息, 就可以随时计算出任何必要元素. 因此, 只需要记录当前算表对应的特征逆矩阵, 并基于该特征逆矩阵和模型初始信息计算出每次行旋转运算所需要的必要元素, 就可以得到与线性规划的行旋转算法完全相同的结果, 这就是线性规划的改进行旋转算法的基本思路. 本文的核心工作就是推导分块矩阵的行旋转运算公式, 并根据该公式设计出计算每张算表对应特征逆矩阵的迭代算法.

改进行旋转算法与改进单纯形算法类似, 仅计算枢轴行和枢轴列上的元素以及更新逆矩阵, 不计算其他元素. 但是, 改进单纯形算法中的逆矩阵阶数等于一般约束的数目, 是固定的. 本文的特征逆矩阵阶数在旋转运算过程中不断变化, 从理论上讲, 阶数不超过一般约束系数矩阵的秩, 实际上往往比秩小很多, 所以比全表格形式运算的计算量小很多. 线性规划规模越大, 特征逆矩阵的相对计算量越小, 改进行旋转算法的效率优势越明显.

Gale 曾经指出, 线性规划问题是线性不等式组的衍生问题, 但是由于没有求解线性不等式组的直接有效算法, 求解线性不等式组需要借助其衍生问题来解决, 这是一种非常尴尬的状态^[13]. 正是由于缺乏像求解线性方程组的 Gauss 消元法那样求解线性不等式组的直接、有效方法, 线性不等式组的基

基础性作用大打折扣. 线性不等式组的行旋转算法改变了没有直接、有效的算法求解线性不等式组的历史. 无论是线性规划的行旋转算法, 还是线性规划的改进行旋转算法, 都只是线性不等式组的行旋转算法的自然应用. 在最优化领域有很多分支都与线性不等式组密切相关, 行旋转算法不改变所求解问题的原始信息, 能够更好地揭示和运用所研究问题系数矩阵的本质结构, 常常能够产生简洁高效的解决方法. 行旋转算法可以高效地解决整数线性规划、大规模资产组合选择和灵敏度分析等问题; 行旋转算法可以以一种统一的方式有效求解包含运输问题、指派问题、最短路问题和最大流问题等在内的大规模一般网络优化模型; 行旋转算法揭示出非凸二次规划系数矩阵的独特结构, 从而为从某个局部最优解转换到其他更好的局部最优解提供了有价值的理论成果. 总之, 线性不等式组的行旋转算法具有重要的方法论意义, 可以作为一种基础算法以一种统一的方式求解与线性不等式组紧密相关的诸多领域中的问题.

致谢 作者衷心感谢审稿专家提出的宝贵意见. 感谢 Richard Cottle 教授在身患多种严重疾病、视力逐渐衰退的情形下对本文的倾心指导. 本文中对于基、基本解等核心概念的定义充分参考了 Cottle 教授的建议. 同时, 向编委和两位审稿人表示诚挚的谢意, 感谢他们对本文初稿提出的富有建设性的建议和评述, 使本文的工作得到较大的提升.

参考文献

- Andersen E D. Finding all linearly dependent rows in large-scale linear programming. *Optim Methods Softw*, 1995, 6: 219–227
- Andersen E D, Andersen K D. Presolving in linear programming. *Math Program*, 1995, 71: 221–245
- Bemporad A, Borrelli F, Morari M. Model predictive control based on linear programming—the explicit solution. *IEEE Trans Automat Control*, 2002, 47: 1974–1985
- Ben-Tal A, Nemirovski A. Robust solutions of linear programming problems contaminated with uncertain data. *Math Program*, 2000, 88: 411–424
- Bixby R E. Solving real-world linear programs: A decade and more of progress. *Oper Res*, 2002, 50: 3–15
- Candes E J, Tao T. Decoding by linear programming. *IEEE Trans Inform Theory*, 2005, 51: 4203–4215
- Cheng M C. General criteria for redundant and nonredundant linear inequalities. *J Optim Theory Appl*, 1987, 53: 37–42
- Cottle R W, George B, Dantzig: Operations research icon. *Oper Res*, 2005, 53: 892–898
- Dalalah D, Lev B. Duality of the improved algebraic method (DIAM). *Omega*, 2009, 37: 1027–1035
- Dantzig G B. Upper bounds, secondary constraints, and block triangularity in linear programming. *Econometrica*, 1955, 23: 174–183
- Dantzig G B. *Linear Programming and Extensions*. Princeton: Princeton University Press, 1963
- Dantzig G B. Linear programming. *Oper Res*, 2002, 50: 42–47
- Gale D. How to solve linear inequalities. *Amer Math Monthly*, 1969, 76: 589–599
- Gondzio J. Interior point methods 25 years later. *European J Oper Res*, 2012, 218: 587–601
- Illés T, Terlaky T. Pivot versus interior point methods: Pros and cons. *European J Oper Res*, 2002, 140: 170–190
- Karmarkar N. A new polynomial-time algorithm for linear programming. *Combinatorica*, 1984, 4: 373–395
- Klotz E, Newman A M. Practical guidelines for solving difficult linear programs. *Surv Oper Res Manag Sci*, 2013, 18: 1–17
- Kuhn H W. Solvability and consistency for linear equations and inequalities. *Amer Math Monthly*, 1956, 63: 217–232
- Liu Y, Tu Y, Zhang Z. The row pivoting method for linear programming. *Omega*, 2021, 100: 102354
- Lustig I J, Marsten R E, Shanno D F. Interior point methods for linear programming: Computational state of the art. *ORSA J Comput*, 1994, 6: 1–14
- Mattheiss T H. An algorithm for determining irrelevant constraints and all vertices in systems of linear inequalities. *Oper Res*, 1973, 21: 247–260
- Mészáros C, Suhl U H. Advanced preprocessing techniques for linear and quadratic programming. *OR Spectrum*, 2003, 25: 575–595
- Morshed M S, Noor-E-Alam M. Generalized affine scaling algorithms for linear programming problems. *Comput Oper Res*, 2020, 114: 104807
- Murav'eva O V. Determination of consistency and inconsistency radii for systems of linear equations and inequalities using the matrix l_1 norm. *Comput Math Math Phys*, 2018, 58: 840–849

The revised row pivoting method for linear programming

Yanwu Liu, Yan Tu, Xiaoyang Zhou, Shouyang Wang & Zhongzhen Zhang

Abstract The row pivoting method is a direct and efficient method for solving any system of linear inequalities and provides a uniform and efficient way to solve many kinds of problems which are closely associated with systems of linear inequalities in the big data era. Dantzig thought that solving a linear programming problem is really all about solving a linear inequality system; therefore the row pivoting method for a system of linear inequalities can be directly applied to solving a linear programming problem. Unlike column pivoting methods such as the simplex method, the row pivoting method for linear programming is based on row geometry (or row vectors), and its core idea is to solve a system of linear inequalities corresponding to the constraints of a linear programming problem while keeping the optimality condition always being true. The revised row pivoting method retains all the characteristics of the row pivoting method for linear programming. The improvement of the new method lies in applying the inverse matrix (called the characteristic inverse matrix) of the non-singular matrix formed by the coefficients of partial variables of the constraints and the original data to calculating the pivot row and the pivot column, which completes a pivoting operation. Since the order of the characteristic inverse matrix is generally much smaller than the number of constraints and variables, the revised pivoting method only needs to calculate a small fraction of necessary elements coming from the corresponding computational tableau in the row pivoting method for each iteration. Therefore, the revised method can significantly improve computational efficiency compared with the row pivoting method for linear programming.

Keywords linear programming, the revised row pivoting method, inverse matrix, block row pivoting operation, system of linear inequalities

MSC(2020) 90C05, 90-08

doi: 10.1360/SSM-2021-0123