2.5 线性规划的原对偶可行路径跟踪内点算法

一个算法用于求解某类问题, 如果它求得问题最优解所用的运算工作量是输入问题规模大小参数的多项式函数时, 那么称这个算法为具有多项式时间复杂性的算法(注: 这是指在最坏情况下, 即算法用于求解这类问题中的所有问题都具有以上性质); 如果所需的运算工作量是输入问题规模大小参数的指数函数时, 那么称这个算法为具有指数时间复杂性的算法. 如果一个算法用于求解某类问题, 在概率意义下具有多项式复杂性, 那么称这个算法在平均意义下具有多项式复杂性. 显然, 随着m或n的增大, 指数增长远快于多项式增长.

在1972年, Klee和Minty给出了一个线性规划的例子, 用单纯形算法求解它时, 需要找遍所有的顶点才能得到其最优解, 即需要迭代2^m次才找到最优解. 因此, 求解线性规划的单纯形算法的复杂性是指数时间复杂性的. 当时的一个开问题是: 对于求解线性规划是否存在具有多项式复杂性的算法? 前苏联数学家Khachiyan于1979年提出的椭球算法正面的回答了这一问题, 但实际的计算表明: 这一算法并不实用.

2.5 线性规划的原对偶可行路径跟踪内点算法

1984年, 美国贝尔实验室工作的Karmarkar提出了求解线性规划的投影内点算法(算法的主要思想相同于求解非线性规划的内罚函数法, 但Karmarkar提出的算法不同于传统的算法), 不但证明了算法的多项式复杂性, 而且算法具有好的计算效果. 在几何方面上, 单纯形算法可看做从可行域边界上的一个顶点开始迭代, 不断地产生使目标函数值逐步改善的顶点序列, 直到算法终止; 而提出的内点算法是从可行域内部的一个可行点(称为可行内点或严格可行点)开始迭代, 不断地产生使目标函数值逐步改善的可行内点序列, 直到算法终止. 由于提出的内点算法产生的所有迭代点都是可行内点, 因此被称为可行内点算法. 上个世纪八十年代末, 为了改进内点算法的实际计算效果, 提出了不可行内点算法, 即在迭代的过程中不保证迭代点的可行性, 但保持所有迭代点为内点(即非负变量在迭代过程中保持是正的), 最终使最优性和可行性同时达到. 由于内点算法具有理论上的多项式复杂性以及实际计算上的有效性, 因而被广为研究, 已经成为求解各种优化问题的最流行的方法之一. 内点算法分为三类: 路径跟踪算法、仿射尺度算法和势函数下降法, 其中原对偶路径跟踪内点算法无论在理论上还是在数值计算方面, 获得了更好的发展. 本书只简单介绍原对偶可行路径跟踪内点算法.

2.5 线性规划的原对偶可行路径跟踪内点算法

考虑一般线性规划问题(2.2.9)和对偶线性规划(2.2.10). 记

$$\begin{aligned} \mathscr{F}^0 &:= \{x \in \mathbb{R}^n \mid Ax = b, x > 0\}, \\ \mathscr{D}^0 &:= \{(y, s) \in \mathbb{R}^m \times \mathbb{R}^n \mid A^\top y + s = c, s > 0\}, \end{aligned}$$

假设 2.5.1 *(a)* $\mathscr{F}^0 \neq \emptyset$且$\mathscr{D}^0 \neq \emptyset$. *(b)* 矩阵$A$的行向量组线性无关.

根据定理2.2.9, 可知: 在假设2.5.1条件下, 原线性规划问题(2.2.9)和对偶线性规划(2.2.10)均是可解的. 因此, 由定理2.2.12, 可通过解最优性条件

$$\begin{cases} Ax = b, \\ A^\top y + s = c, \\ x \geq 0, \quad s \geq 0, \quad x^\top s = 0, \end{cases} \tag{2.5.1}$$

来求解问题(2.2.9)和问题(2.2.10)的最优解. 以下记

$$\Omega^0 := \{(x, y, s) \mid x \in \mathscr{F}^0, (y, s) \in \mathscr{D}^0\}.$$

2.5 线性规划的原对偶可行路径跟踪内点算法

对任意的向量$u \in \mathbb{R}^l$, 记U表示以向量u的分量为主对角元素的对角矩阵. 容易得到: 如果$x \geq 0$和$s \geq 0$, 那么$x^\top s = 0$当且仅当$Xs = 0$. 考虑等式与不等式组:

$$\begin{cases} Ax = b, \\ A^\top y + s = c, \\ x > 0, \quad s > 0, \quad Xs = \mu e, \end{cases} \tag{2.5.2}$$

其中, μ是一个大于零的参数且e是全部分量为1的n维向量. 在假设2.5.1条件下可以证明:

- 对任意给定的$\mu > 0$, 等式与不等式组(2.5.2)有唯一的解, 记为$(x(\mu), y(\mu), s(\mu)) \in \Omega^0$;
- 解函数$(x(\mu), y(\mu), s(\mu))$相对于参数$\mu > 0$是连续函数;
- 令$\mu \downarrow 0$, 那么解序列$\{(x(\mu), y(\mu), s(\mu))\}$的极限点是问题(2.5.1)的解, 因此, 它们分别是原线性规划问题(2.2.9)和对偶线性规划(2.2.10)的最优解.

所以, 点列$\{(x(\mu), y(\mu), s(\mu))\}$构成了一条通向原线性规划(2.2.9)与其对偶规划(2.2.10) 最优解的路径, 称为中心路径.

2.5 线性规划的原对偶可行路径跟踪内点算法

粗略地讲，原对偶可行路径跟踪内点算法就是：任意选取可行内点$(x^0, y^0, s^0) \in \Omega^0$并且置$\mu_0 := (x^0)^\top s^0/n > 0$，沿着中心路径寻找新的迭代点$(x^+, y^+, s^+) \in \Omega^0$并使得$\mu_+ := (x^+)^\top s^+/n$下降，重复迭代直到算法终止。一般地，人们迭代地求解下面的方程组

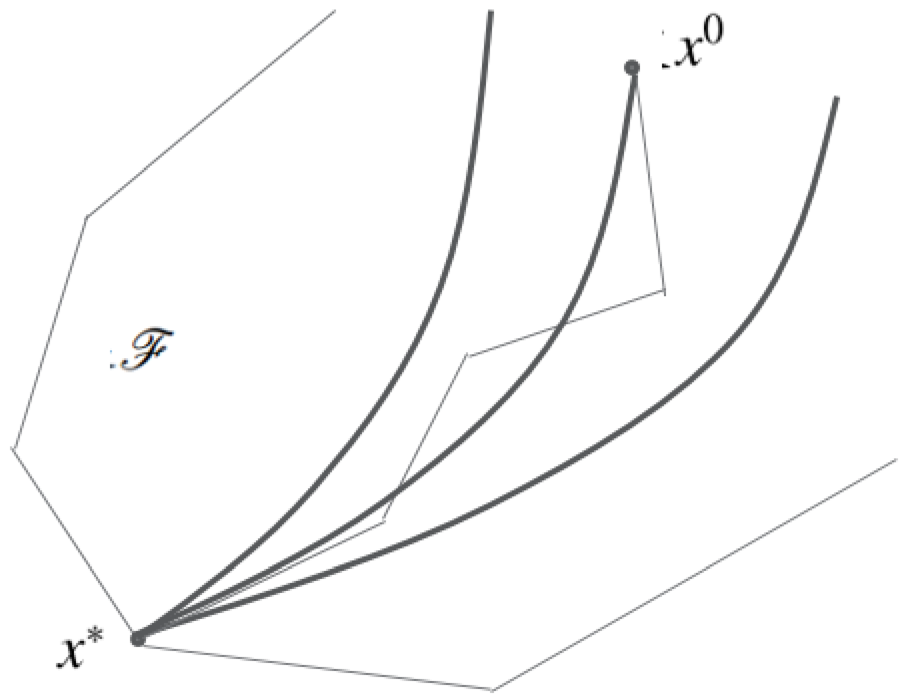
$$\begin{cases} Ax - b & = & 0, \\ A^\top y + s - c & = & 0, \\ Xs - \mu e & = & 0, \end{cases}$$

并且使得新迭代点既为可行内点又在中心路径附近。保证新迭代点具有以上性质的方法有很多。方法之一是使迭代点属于邻域

$$\mathcal{N}(\beta) := \left\{(x, y, s) \mid (x, y, s) \in \Omega^0, \|Xs - \mu e\| \le \beta\mu\right\},$$

其中$\beta \in (0, 1)$是一个给定的参数且$\|\cdot\|$表示$\|\cdot\|_2$。如果初始点和所有迭代点都在邻域$\mathcal{N}(\beta)$内，那么对应的算法称为小步长算法。

2.5 线性规划的原对偶可行路径跟踪内点算法



2.5 线性规划的原对偶可行路径跟踪内点算法

本节介绍一个小步长的原对偶可行路径跟踪内点算法. 具体地, 在找新迭代点时, 首先求解下述方程组

$$\left\{\begin{array}{rcccl} A\Delta x & & & = & 0, \\ & A^\top \Delta y & + \quad \Delta s & = & 0, \\ S\Delta x & & + \quad X\Delta s & = & \sigma\mu e - Xs \end{array}\right. \tag{2.5.3}$$

来获得扰动的牛顿方向$(\Delta x, \Delta y, \Delta s)$, 其中$\sigma \in (0,1)$是一个参数; 然后按以下方式进行迭代:

$$x^+ = x + \Delta x, \quad y^+ = y + \Delta y, \quad s^+ = s + \Delta s, \tag{2.5.4}$$

并且在迭代过程中, 使用$\mu := \mu(x,s) = x^\top s/n$来校正参数$\mu$.

引理 2.5.1 假设矩阵A行满秩, 那么方程组(2.5.3) 是可解的.

2.5 线性规划的原对偶可行路径跟踪内点算法

引理 2.5.2　令假设2.5.1条件成立且$(x, y, s) \in \mathcal{N}(\beta)$. 若$(\Delta x, \Delta y, \Delta s)$是方程组(2.5.3)的一个解, (x^+, y^+, s^+)是由(2.5.4)式所得, 且$\mu_+ = (x^+)^\top s^+/n$, 则下面的结论成立.

(i) $\mu_+ = \sigma\mu$.

(ii) 记$D := X^{-1/2}S^{1/2}$. 那么$\|D\Delta x\|^2 + \|D^{-1}\Delta s\|^2 \leq \frac{\beta^2+(1-\sigma)^2 n}{1-\beta}\mu$.

(iii) $\|X^+ s^+ - \mu_+ e\| \leq \frac{\beta^2+(1-\sigma)^2 n}{2(1-\beta)}\mu$.

2.5 线性规划的原对偶可行路径跟踪内点算法

原对偶可行路径跟踪内点算法的具体步骤如下:

算法 2.5.1 *(对偶可行路径跟踪内点算法)*

步0 *(选择起始参数)*　取$\sigma \in (0,1)$. 选择参数$\beta \in (0,1/2)$并使得$\beta^2+(1-\sigma)^2 n \le 2(1-\beta)\sigma\beta$. 令$\varepsilon \in (0,1)$充分小. 选取$(x^0, y^0, s^0) \in \mathcal{N}(\beta)$. 置$\mu_0 := \mu(x^0, s^0)$和$k := 0$.

步1 *(检测终止条件)*　若$\mu_k \le \varepsilon$, 则算法终止. 否则, 转步2.

步2 *(迭代步)*　取$\mu := \mu_k$和$(x, y, s) := (x^k, y^k, s^k)$. 解方程组(2.5.3), 设解记为$(\Delta x^k, \Delta y^k, \Delta s^k)$. 置

$$(x^{k+1}, y^{k+1}, s^{k+1}) := (x^k, y^k, s^k) + (\Delta x^k, \Delta y^k, \Delta s^k).$$

转步3.

步3 *(重复迭代)*　置$\mu_{k+1} := \mu(x^{k+1}, s^{k+1})$和$k := k+1$, 转步1.

2.5 线性规划的原对偶可行路径跟踪内点算法

定理 2.5.1 若假设*2.5.1*条件成立，则算法*2.5.1*是适定的．如果序列$\{(x^k, y^k, s^k)\}$是由算法*2.5.1* 产生的迭代序列，那么对任意的$k \in \{0, 1, 2, \ldots\}$，有

$$(x^k, y^k, s^k) \in \mathcal{N}(\beta) \quad 且 \quad \mu_{k+1} = \sigma^k \mu_0.$$

定义线性规划问题(2.2.9)的数据大小为

$$L := \lceil \log(1+\chi) \rceil + \lceil \log(1+\max_j |c_j|) \rceil + \lceil \log(1+\max_i |b_i|) \rceil + \lceil \log(m+n) \rceil,$$

其中，$\lceil \xi \rceil$表示不小于ξ的最小正整数，χ表示矩阵A的所有方子矩阵的行列式的绝对值中的最大值．

定理 2.5.2 令假设*2.5.1*条件成立．在算法*2.5.1*中，若取$\sigma := 1 - \frac{\delta}{\sqrt{n}}$，其中$\delta \in (0, 1)$，则下面的结论成立．

(i) 算法*2.5.1*的迭代步数不大于$\hat{k} := \lceil \ln(\varepsilon^{-1}\mu_0)\sqrt{n}/\delta \rceil$;

(ii) 如果$\mu_0 := 2^{O(L)}$并且$\varepsilon := 2^{O(L)}$，那么算法*2.5.1*最多$O(\sqrt{n}L)$步可找到原线性规划问题*(2.2.9)* 和对偶线性规划*(2.2.10)*的一对ε-最优解．

2.5 线性规划的原对偶可行路径跟踪内点算法

最后阐述一下可行内点算法的启动问题. 可行内点算法, 首先需要找到问题的一个可行内点, 算法才能开始迭代, 如何找到一个可行内点, 则是一个难问题. 自Karmarkar的内点算法提出后, 算法的启动问题一直困扰着人们, 虽然做了各种尝试, 例如: 大M法, 组合二阶段法等, 结果都不甚理想. 1994年, Ye, Todd, 和Mizuno (见文献[23]) 提出了求解线性规划的齐次自对偶内点算法, 不但完美地解决了可行内点算法的启动问题, 而且还得到了其它一些好的性质.

2.5 线性规划的原对偶可行路径跟踪内点算法—作业

引理 2.5.1 假设矩阵A行满秩，那么方程组(2.5.3) 是可解的.