

DOI:10.16356/j.1005-2615.2020.02.020

## 基于演化计算的线性规划原对偶内点法中的 初始点选取算法

贾伟<sup>1</sup>, 雍龙泉<sup>1,2</sup>, 李娜<sup>1</sup>

(1. 陕西理工大学数学与计算机科学学院, 汉中, 723001; 2. 陕西省工业自动化重点实验室, 汉中, 723001)

**摘要:** 采用原对偶内点法求解线性规划问题, 对初始点要求严格。根据初始可行内点的准则, 定义了相应的达成度函数, 并由达成度函数定义了适应值函数, 从而提出了基于演化计算的线性规划原对偶内点法中的初始点选取算法。该算法基于和声搜索演化算法实现, 经数值实验测试, 结果表明, 对所选取测试的典型线性规划问题, 算法都能求得大部分问题的初始可行内点。

**关键词:** 线性规划; 原对偶内点法; 初始点选取; 演化计算; 和声搜索

**中图分类号:** TP18; O221      **文献标志码:** A      **文章编号:** 1005-2615(2020)02-0334-07

### Initial Point Selection in Primal-Dual Interior Point Method for Linear Programming Based on Evolutionary Algorithm

JIA Wei<sup>1</sup>, YONG Longquan<sup>1,2</sup>, LI Na<sup>1</sup>

(1. School of Mathematics and Computer Science, Shaanxi University of Technology, Hanzhong, 723001, China;  
2. Shaanxi Key Laboratory of Industrial Automation, Hanzhong, 723001, China)

**Abstract:** The primal-dual interior point method requires strict initial feasible points in solving linear programming problems. The degree function is given based on the criterion of the initial feasible point in the primal-dual interior point method, and the fitness function in the evolutionary algorithm is designated by the degree function. The initial feasible point selection of the primal-dual interior point method for linear programming based on the evolutionary computation is proposed. It is taken by the harmony search. Numerical experiments show that the algorithm can find an initial feasible point of most given problems, thus validating the effectiveness of the algorithm.

**Key words:** linear programming; primal-dual interior point method; initial point selection; evolutionary computation; harmony search

线性规划(Linear programming, LP)是一类重要的数学规划问题,它应用广泛、发展较快、方法较成熟,已成为辅助人们进行科学管理的一种数学方法。到目前为止,求解线性规划的常见方法有:单纯形法及对偶单纯形法<sup>[1-3]</sup>,可行内点法及不可行

内点算法<sup>[4-8]</sup>。近年来, Bai等<sup>[9-10]</sup>对核函数内点法进行了深入的研究,并取得了系列研究成果; Ye等<sup>[11-12]</sup>把内点算法应用于物流选址及路径优化、库存管理、投资组合优化等问题,并获得了巨大的商业价值。近年来,采用高阶牛顿法求解线性规划也

**基金项目:** 国家自然科学基金(11401357)资助项目; 陕西省教育厅科研基金(17JK0146)资助项目; 陕西理工大学科研基金(SLGKY16-15)资助项目。

**收稿日期:** 2020-01-16; **修订日期:** 2020-03-21

**通信作者:** 贾伟, 男, 讲师, E-mail: jiawei@snut.edu.cn。

**引用格式:** 贾伟, 雍龙泉, 李娜. 基于演化计算的线性规划原对偶内点法中的初始点选取算法[J]. 南京航空航天大学学报, 2020, 52(2): 334-340. JIA Wei, YONG Longquan, LI Na. Initial point selection in primal-dual interior point method for linear programming based on evolutionary algorithm[J]. Journal of Nanjing University of Aeronautics & Astronautics, 2020, 52(2): 334-340.

成为一个热点<sup>[13-14]</sup>。

单纯形算法的特点是从一个基可行解旋转到另一个基可行解,从几何上而言就是从可行域的一个顶点迭代到另一个顶点,直至到达最优顶点为止。单纯形方法简单、实用,针对小规模线性规划求解非常有效。1972年,Klee和Minty举例说明了单纯形算法的时间复杂性是指数型,因此线性规划问题是否存在多项式型有效算法便成了计算机科学家和数学家迫切需要研究的问题。前苏联数学家Khachian在1979年给出了一个求解线性规划的椭球算法,计算复杂性为 $O(n^6L^2)$ ;印度数学家Karmarkar在1984年给出了线性规划的一个新的多项式算法,其计算复杂性为 $O(n^{3.5}L^2)$ ,Monteiro和Adler在1989年给出了求解线性规划一个原对偶内点算法,计算复杂性为 $O(n^{3.5}L)$ ,其迭代次数为 $O(n^{0.5}L)$ 。这些多项式算法的特点是从可行域内部一个适当的点出发,通过有限次迭代到达最优解,这便是内点算法的思想。

内点算法对初始点的要求比较严格,既要满足原可行性和对偶可行性,又要求在中心路径邻域内。文献[5]中给出了寻找初始可行内点的方法:构造一个比原问题更为复杂的辅助线性规划(增加了若干复杂参数),通过求解辅助线性规划而得到原问题的初始内点(事实上,求解辅助线性规划的难度比求解原问题还复杂)。文献[15]中初步给出了原对偶内点法的两个较为简单的线性规划算例,并通过改变参数研究了其收敛过程,但该方法不具有普适性,且对变量较多的线性规划难以找到初始可行内点。

本文依据文献[5]中的初始可行内点的约束准则,设计了演化计算中种群个体的分量;定义了4个准则的达成度函数;建立了演化计算的适应值函数;给出了基于演化计算选取初始可行内点的算法描述,并采用和声搜索算法来进行实现。通过求解9个线性规划问题(包含6个唯一解、2个多个解、1个无解的算例),结果表明,针对存在解的8个线性规划问题,其中7个都找到了初始可行内点。

## 1 内点算法中初始点选取准则

考虑线性规划及对偶问题

$$\begin{aligned} \text{(LP)} \quad & \begin{cases} \min & c^T x \\ \text{s.t.} & Ax = b \quad x \geq 0 \end{cases} \\ \text{(DP)} \quad & \begin{cases} \max & b^T y \\ \text{s.t.} & A^T y + z = c \quad z \geq 0 \end{cases} \end{aligned} \quad (1)$$

式中: $c \in \mathbb{R}^n, x \in \mathbb{R}^n, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, z \in \mathbb{R}^n, y \in \mathbb{R}^m, m \leq n, \text{rank}(A) = m$ 。

采用文献[5,15]中的原对偶内点法求解线性规划时,对初始点的要求较为严格。

要求初始点 $w^{(0)} = (\theta, \delta, x^{(0)}, y^{(0)}, z^{(0)})$ 及参数 $\theta$ 与 $\delta$ 满足以下准则:

$$\text{准则0} \quad 0 < \theta < 0.5; 0 < \delta < \sqrt{n}; \frac{\theta^2 + \delta^2}{2(1-\theta)} \leq$$

$$\theta \left( 1 - \frac{\delta}{\sqrt{n}} \right); \text{(对参数的要求)}$$

$$\text{准则1} \quad Ax^{(0)} = b, x^{(0)} > 0; \text{(原可行性)}$$

$$\text{准则2} \quad A^T y^{(0)} + z^{(0)} = c, z^{(0)} > 0; \text{(对偶可行性)}$$

$$\text{准则3} \quad \|f(w^{(0)}) - \mu_0 e\| \leq \theta \mu_0; \text{(在中心路径邻域内)}$$

$$\begin{aligned} \text{其中: } f(w^{(0)}) &= XZe, X = \text{diag}(x_1, x_2, \dots, x_n), \\ Z &= \text{diag}(z_1, z_2, \dots, z_n), e = (1, 1, \dots, 1)^T, \mu_0 = \\ & \frac{(x^{(0)})^T z^{(0)}}{n}, \|\cdot\| \text{表示2范数。} \end{aligned}$$

直接寻找满足上述准则的初始点较为困难,下面采用进化算法,建立适应值函数,利用和声搜索算法(Harmony search, HS)优化适应值函数,从而获得满足条件的初始点。

## 2 基于演化计算的初始点选取算法

对初始点的选取,若采用Monte Carlo随机产生的方式,是很难得到符合满足上述准则的初始点。演化计算(Evolutionary computation, EC)是一类受自然进化和演变启示而提出的各种计算方法。常见的演化计算有遗传算法(Genetic algorithm, GA)、粒子群优化算法(Particle swarm optimization, PSO)、蚁群算法(Ant colony optimization, ACO)、差分进化算法(Differential evolution, DE)及和声搜索算法等,更多的演化计算详见文献[16]。演化计算能在可接受的时间内获得问题的近似最优解,且具有鲁棒性强等优势,已被广泛应用于旅行商问题、背包问题、生产调度等优化问题,通过种群内个体之间的学习逼近问题的最优解。

### 2.1 演化计算中种群个体的设计

采用演化计算选取初始点,首先考虑演化计算种群中个体的构成分量。一种方案可以将初始点 $w^{(0)} = (\theta, \delta, x^{(0)}, y^{(0)}, z^{(0)})$ 直接作为种群个体,但它们受到线性规划中矩阵 $A$ 和向量 $b, c$ 的有关等式约束(即满足原可行性及对偶可行性)。对种群中个体分量的选择,做如下分析。

对非齐次方程组 $Ax = b, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m, m \leq n, \text{rank}(A) = m, r = n - m$ 中,设 $\eta_0$ 为其一个

特解,  $\xi_1, \xi_2, \dots, \xi_r$  为其导出组  $Ax=0$  的基础解系, 则其通解为  $x = \eta_0 + k_1\xi_1 + k_2\xi_2 + \dots + k_r\xi_r$ , 其中  $k_1, k_2, \dots, k_r$  为任意常数。当  $k_1, k_2, \dots, k_r$  取适当值时,  $x = \eta_0 + k_1\xi_1 + k_2\xi_2 + \dots + k_r\xi_r > 0$ , 从而满足准则1。

利用方程  $A^T y + z = c$  构造  $BY = c$ , 这里  $B = [A^T, I_{n \times n}]$ ,  $Y = \begin{bmatrix} y \\ z \end{bmatrix}$ 。对  $BY = c$ , 设  $\gamma_0$  为其一个特解,  $\gamma_1, \gamma_2, \dots, \gamma_s$  为其导出组  $BY = 0$  的基础解系, 则其通解为  $Y = \gamma_0 + l_1\gamma_1 + l_2\gamma_2 + \dots + l_s\gamma_s$ , 其中  $s = (m+n) - \text{rank}(B)$ ,  $l_1, l_2, \dots, l_s$  为任意常数, 则  $Y$  的前  $m$  个分量就是  $y$ , 而  $Y$  剩余的  $n$  个分量就是  $z$ ; 同理, 当  $l_1, l_2, \dots, l_s$  取适当的值时, 就可使得  $z > 0$ , 及满足准则2。

定义种群优化算法中个体构成分量为  $(\theta, \delta, k_1, k_2, \dots, k_r, l_1, l_2, \dots, l_s)$ 。其中  $0 < \theta < 0.5, 0 < \delta < \sqrt{n}, k_i \in \mathbf{R}, i = 1, 2, \dots, r, l_j \in \mathbf{R}, j = 1, 2, \dots, s$ , 且个体分量维数为  $\text{dim} = 2 + r + s$ 。

## 2.2 适应值函数的定义

在演化算法求解问题的过程中, 演化算法依据个体适应值来挑选父代, 采用不同的算子来产生新的种群子代, 一般子代的适应值优于父代的适应值, 在不断的迭代演化过程中, 最终得到符合要求的个体子代, 求得问题的解。可见, 演化计算求解问题, 关键问题之一就是将问题转换为一个最优化问题——设计合适的种群个体适应值函数。

为了构造演化计算的适应值函数, 对上述4个准则, 定义如下几个达成度函数

$$f_0(\mathbf{w}^{(0)}) = \begin{cases} 1 & \frac{\theta^2 + \delta^2}{2(1-\theta)} \leq \theta \left(1 - \frac{\delta}{\sqrt{n}}\right) \\ 0 & \text{其他} \end{cases} \quad (2)$$

$$f_1(\mathbf{w}^{(0)}) = \sum_{i=1}^n \sigma(x_i) \quad \sigma(x_i) = \begin{cases} 1 & x_i > 0 \\ 0 & \text{其他} \end{cases} \quad (3)$$

$$f_2(\mathbf{w}^{(0)}) = \sum_{i=1}^n \chi(z_i) \quad \chi(z_i) = \begin{cases} 1 & z_i > 0 \\ 0 & \text{其他} \end{cases} \quad (4)$$

$$f_3(\mathbf{w}^{(0)}) = \begin{cases} 1 & \|f(\mathbf{w}^{(0)}) - \mu_0 e\| \leq \theta \mu_0 \\ 0 & \text{其他} \end{cases} \quad (5)$$

如果上述条件都满足, 则  $f_0(\mathbf{w}^{(0)}) + f_1(\mathbf{w}^{(0)}) + f_2(\mathbf{w}^{(0)}) + f_3(\mathbf{w}^{(0)}) = 1 + n + n + 1$ , 于是构造差项  $(1 + n + n + 1) - [f_0(\mathbf{w}^{(0)}) + f_1(\mathbf{w}^{(0)}) + f_2(\mathbf{w}^{(0)}) + f_3(\mathbf{w}^{(0)})]$ , 使其最小。

有了上面的分析, 定义演化计算的适应值函数为

$$\min F(\mathbf{w}^{(0)}) \triangleq (1 + n + n + 1) - [f_0(\mathbf{w}^{(0)}) +$$

$$f_1(\mathbf{w}^{(0)}) + f_2(\mathbf{w}^{(0)}) + f_3(\mathbf{w}^{(0)})]$$

当且仅当  $F(\mathbf{w}^{(0)}) = 0$  时, 则找到可行内点。

## 2.3 演化计算寻找初始点算法描述

通过演化计算求解对偶内点法的初始点选取问题的算法流程描述如下。

**算法1** 演化计算寻找可行初始内点算法

**Step 1** 问题数据  $A, b, c, m, n$  初始化。

**Step 2** 种群演化计算参数初始化, 如种群大小 popsize 等。

**Step 3** 初始化种群。

**Step 4** 求种群中每个个体的适应值。

**Step 5** 选取种群中最优适应值 bestfit 和对应个体 bestx。

**Step 6** 如果 bestfit  $\leq 0$  (满足约束要求) 或超过最大迭代次数, 则迭代结束; 转 Step 12。

**Step 7** 依据种群优化算法, 生成新个体。

**Step 8** 计算新个体的适应值。

**Step 9** 淘汰适应度差的个体。

**Step 10** 选取种群中最优适应值 bestfit 和对应个体 bestx。

**Step 11** 返回 Step 6。

**Step 12** 如果 bestfit  $\leq 0$ , 则根据 bestx 构造可行初始点  $\mathbf{w}^{(0)} = (\theta, \delta, x^{(0)}, y^{(0)}, z^{(0)})$ 。

## 3 和声搜索算法求解

和声搜索算法是2001年韩国学者 Geem 等人提出的一种新颖的智能优化算法, 具有通用、简单、易于实现、群体协同搜索等优点<sup>[17-20]</sup>, 故本文选取了和声搜索作为演化计算的实现实例, 来实现算法1, 具体步骤描述如下所示。

**算法2** 和声搜索求解初始点选取算法

**Step 1** 问题数据  $A_{m \times n}, b, c$  初始化, 问题维数 dim。

**Step 2** 参数初始化: ①和声记忆库的大小 HMS; ②和声记忆库取值概率 HMCR; ③音调微调概率 PAR; ④音调微调带宽  $b_w$ ; ⑤最大创作次数  $T_{\max}$ 。

**Step 3** 随机初始化和声记忆库  $HM_{HMS \times \text{dim}}$ 。

**Step 4** 求种群中每个个体的适应度 fitness。

**Step 5** 选取种群中最优适应值 bestfit 和其个体 bestx。

**Step 6** 设当前迭代次数  $t = 1$ 。

**Step 7** 如果 bestfit  $\leq 0$  或迭代次数  $t > T_{\max}$ , 则迭代结束; goto Step 28。

**Step 8**  $i = 1$

**Step 9** while  $i > \text{dim}$ , 则 goto Step 22。

**Step 10** 产生 $[0,1]$ 之间的随机数  $\text{rand1}$ 。  
**Step 11** 如果  $\text{rand1} \geq \text{HMCR}$ , 则 goto Step 14。  
**Step 12** 随机产生  $x_{\text{new}}(i)$ 。  
**Step 13** Goto Step 20。  
**Step 14** 产生 $[1, \text{HMS}]$ 之间的随机整数  $d$ 。  
**Step 15**  $x_{\text{new}}(i) = \text{HM}(d, i)$ 。  
**Step 16** 产生 $[0,1]$ 之间的随机数  $\text{rand2}$ 。  
**Step 17** 如果  $\text{rand2} \geq \text{PAR}$ , 则 goto Step 20。  
**Step 18** 产生 $[-1,1]$ 之间的随机数  $\text{rand3}$ 。  
**Step 19**  $x_{\text{new}}(i) = x_{\text{new}}(i) + b_w * \text{rand3}$ 。  
**Step 20**  $i = i + 1$ 。  
**Step 21** Goto Step 9。  
**Step 22** 计算新个体的适应值  $\text{newfit} = f(x_{\text{new}})$ 。

**Step 23** 选择 HM 中适应值最差的和声  $\text{worstx}$  和其适应值  $\text{worstfit}$ 。  
**Step 24** 如果  $\text{newfit}$  优于  $\text{worstfit}$ , 则用  $\text{newfit}$ 、 $x_{\text{new}}$  替换  $\text{worstfit}$ 、 $\text{worstx}$ 。  
**Step 25** 选取种群中最优适应值  $\text{bestfit}$  和个体  $\text{bestx}$ 。  
**Step 26**  $t = t + 1$ 。  
**Step 27** 返回 Step 7。  
**Step 28** 如果  $\text{bestfit} \leq 0$ , 则根据  $\text{bestx}$  构造可行初始点  $w^{(0)} = (\theta, \delta, x^{(0)}, y^{(0)}, z^{(0)})$ 。

## 4 数值实验

### 4.1 测试问题

对线性规划问题, 选取了如下 9 个线性规划问题, 如表 1 所示。

表 1 测试算法的 9 个线性规划问题  
 Table 1 Nine linear programming problems tested for the algorithm

问题编号	矩阵 $A$ , 向量 $b$ , 向量 $c$	备注
LP01	$A = [1 \ 0 \ 0 \ 1/4 \ -8 \ -19; 0 \ 1 \ 0 \ 1/2 \ -1 \ 2 \ -1/2 \ 3; 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0];$ $b = [1.25 \ -8 \ 2]^T; c = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7]^T$	唯一解
LP02	$A = [2 \ -1 \ 1 \ 6 \ -5 \ -10; 1 \ 1 \ 2 \ 1 \ 2 \ 0 \ -1];$ $b = [6 \ 3]^T; c = [3 \ 4 \ 6 \ 7 \ 1 \ 0 \ 0]^T$	唯一解
LP03	$A = [-4 \ -2 \ 13 \ -2 \ 10 \ 0; 12 \ -1 \ 1 \ 2 \ 0 \ 10; 8 \ 1 \ -1 \ -1 \ 0 \ 0 \ 1];$ $b = [46 \ 24 \ 3]^T; c = [2 \ -1 \ -3 \ 2.5 \ -2 \ 0 \ 0]^T$	唯一解
LP04	$A = [-1 \ 2 \ 1 \ 0 \ 0; 3 \ 2 \ 0 \ 1 \ 0; 1 \ -1 \ 0 \ 0 \ 1];$ $b = [4 \ 14 \ 3]^T; c = [-3 \ -2 \ 0 \ 0 \ 0]^T$	多个解
LP05	$A = [1 \ -2 \ 1 \ 0 \ 0; 0 \ 1 \ -3 \ 1 \ 0; 0 \ 1 \ -1 \ 0 \ 1];$ $b = [2 \ 1 \ 2]^T; c = [0 \ -1 \ -2 \ 0 \ 0]^T$	无解
LP06	$A = [-1 \ 2 \ -2 \ 3 \ -3 \ 1 \ 0 \ 1; 1 \ 1 \ -3 \ 2 \ -2 \ 0 \ 0 \ 1; 0 \ 0 \ -2 \ 3 \ -3 \ 0 \ 1 \ 1];$ $b = [1 \ 4 \ 2]^T; c = [4 \ 3 \ 5 \ 1 \ 2 \ 0 \ 0 \ 0]^T$	多个解
LP07	$A = [1 \ 1 \ 1 \ 0 \ 0 \ 0; 0 \ 0 \ 0 \ 1 \ 1 \ 1; 1 \ 0 \ 1 \ 0 \ 1 \ 0];$ $b = [1 \ 5 \ 8 \ 12]^T; c = [5 \ 2 \ 3 \ 7 \ 9 \ 1]^T$	唯一解
LP08	$A = [1 \ 1 \ 2; 2 \ 1 \ 3]; b = [3 \ 5]^T; c = [2 \ 1 \ 4]^T$	唯一解
LP09 (Klee-Minty 问题, $n=3, \delta=1/4$ 时)	$A = [1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0; 1/4 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0; 0 \ 1/4 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0; 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1 \ 0 \ 0;$ $-1/4 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1 \ 0; 0 \ -1/4 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1];$ $b = [1 \ 1 \ 1 \ 0 \ 0 \ 0]^T; c = [0 \ 0 \ -1 \ 0 \ 0 \ 0 \ 0 \ 0]^T$	唯一解

### 4.2 算法参数设置

算法参数设置: ①和声记忆库的大小  $\text{HMS} = 10$ ; ②和声记忆库取值概率  $\text{HMCR} = 0.9$ ; ③音调微调概率  $\text{PAR} = 0.3$ ; ④音调微调带宽  $b_w = 0.01$ ; ⑤最大创作次数  $T_{\text{max}} = 10\ 000$ 。

测试问题中, 种群个体  $(\theta, \delta, k_1, k_2, \dots, k_r, l_1, l_2, \dots, l_s)$  的搜索范围分别设为  $0 < \theta < 0.5, 0 < \delta < \sqrt{n}, -10 < k_i < 10 (i = 1, 2, 3, \dots, r), -10 <$

$l_j < 10 (j = 1, 2, 3, \dots, s)$ 。

### 4.3 实验结果数据统计

为了消除随机数对算法的影响, 对每个问题, 算法独立执行 100 次, 表 2 中统计出了在 100 次中成功找到初始可行内点的次数, 最小迭代次数、平均迭代次数及平均耗时。N 表示没有找到可行内点。

实验结果表明, 对 8 个存在最优解的线性规划

表2 实验测试结果统计表

Table 2 Statistical table of experimental test results

问题编号	LP01	LP02	LP03	LP04	LP05	LP06	LP07	LP08	LP09
成功次数	100	95	0	89	0	86	28	99	97
最小迭代次数	75	98	N	70	N	237	196	12	184
平均迭代次数	408.70	879.03	N	444.92	N	1 997.23	719.64	967.66	1 183.53
平均耗时	0.32	0.63	N	0.22	N	1.76	0.44	0.27	1.16

问题,除了LP03外,其他7个问题都找到多个满足条件的初始可行内点,且大部分问题(除了LP03、LP07外)在100次独立运行中,都有超过80%的比例找到初始可行内点,且最小迭代次数和平均的迭代次数都远远小于 $T_{max}$ 。

### 5 应用于Klee-Minty问题

线性规划问题

$$\begin{cases} \max x_n \\ \text{s.t. } 0 \leq x_1 \leq 1 \\ \delta x_{i-1} \leq x_i \leq 1 - \delta x_{i-1} \quad i = 2, 3, \dots, n \\ x_i \geq 0 \quad i = 1, 2, \dots, n \end{cases} \quad (6)$$

式中: $0 \leq \delta \leq 0.5$ 。该问题特点是对具有 $n$ 个变量的问题,单纯形方法起点取为原点时,需要 $2^n - 1$ 次迭代才能找到其最优解,即遍历所有顶点才能到达最优解。

当 $n = 3$ 时,取 $\delta = \frac{1}{4}$ ,即LP09,对应的线性规划

划为

$$\begin{cases} \max x_3 \\ \text{s.t. } 0 \leq x_1 \leq 1 \\ \frac{1}{4} x_1 \leq x_2 \leq 1 - \frac{1}{4} x_1 \\ \frac{1}{4} x_2 \leq x_3 \leq 1 - \frac{1}{4} x_2 \quad x_1, x_2, x_3 \geq 0 \end{cases} \Rightarrow \begin{cases} \max x_3 \\ \text{s.t. } x_1 \leq 1 & x_2 + \frac{1}{4} x_1 \leq 1 \\ x_2 - \frac{1}{4} x_1 \geq 0 & x_3 + \frac{1}{4} x_2 \leq 1 \\ x_3 - \frac{1}{4} x_2 \geq 0 & x_1, x_2, x_3 \geq 0 \end{cases} \quad (7)$$

其最优解为 $x_1 = 0, x_2 = 0, x_3 = 1$ ,最优目标值为1。若初始点取为 $x_1 = 0, x_2 = 0, x_3 = 0$ ,采用单纯形方法,则在可行域上的单纯形迭代过程为 $x^{(0)} \rightarrow x^{(1)} \rightarrow x^{(2)} \rightarrow x^{(3)} \rightarrow x^{(4)} \rightarrow x^{(5)} \rightarrow x^{(6)} \rightarrow x^{(7)}$ ,如图1所示。

采用本文的方法,找到一组满足条件的初始可行内点为:

$$\theta = 0.498\ 601\ 355\ 657\ 197, \delta = 0.388\ 126\ 553\ 590\ 515,$$

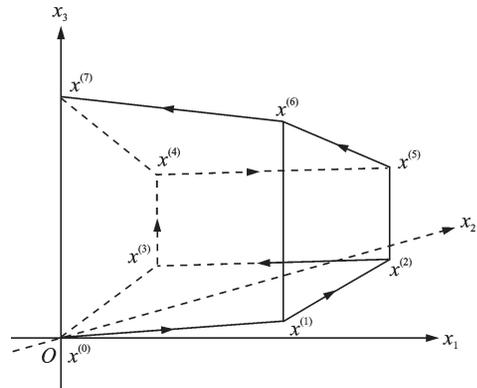


图1 单纯形法迭代过程示意图

Fig.1 Schematic diagram of simplex iterative process

$$x^{(0)} = (0.39, 0.57, 0.61, 0.61, 1.35, 1.87, 1.01, 1.86)^T,$$

$$y^{(0)} = (-9.47, -4.06, 3.12, -4.58, 2.77)^T,$$

$$z^{(0)} = (16.64, 11.12, 6.27, 9.47, 4.06, 3.12, 4.58, 2.77)^T.$$

采用文献[5, 15]中的原对偶内点算法求解,图2给出了部分变量的收敛图及目标函数的收敛图。

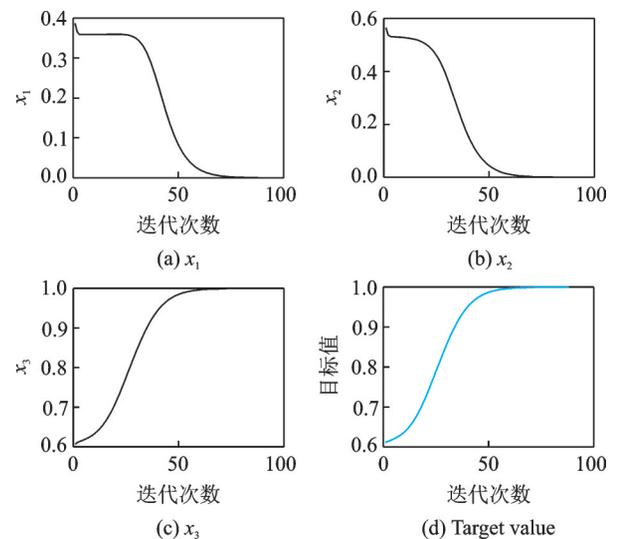


图2 原对偶内点算法求解的收敛图及目标函数的收敛图

Fig.2 Convergence graph of the primal-dual interior point algorithm

此外,对于Klee-Minty问题,当 $n$ 分别取5、10、20, $\delta = \frac{1}{4}$ 的情况,经程序运行测试,结果数据统计如表3所示。

表 3 Klee-Minty 问题测试结果数据统计表

Table 3 Statistical table of test results of Klee-Minty problem

$n$	$\delta=0.25$		
	5	10	20
成功次数	100	100	99
最小迭代次数	1 139	14 287	457 666
平均迭代次数	3 657.72	32 781.08	1 039 628
平均耗时	0.118 4	0.910 3	38.658 5

## 6 结 论

线性规划内点法的难点在于初始可行内点要满足多个准则,本文对每一个准则分别构造了相应的达成度函数,建立了演化计算的适应值函数,通过优化该适应值函数获得线性规划原对偶内点算法的初始内点。计算结果表明,对 8 个存在最优解的线性规划问题,其中 7 个问题都找到了多个初始可行内点(1 个问题没有找到);并把该方法应用于求解 Klee-Minty 问题。

基于演化计算的初始点选取,算法虽具有通用性,且可以找到大部分问题的初始可行内点,但由于演化算法本身具有的随机性,并不能保证算法的每一次运行都能找到满足全部准则的初始可行内点,而且问题规模越大,算法的迭代次数也较多,甚至可能寻找失败。可是,即使算法返回失败的结果,其寻找的初始点也符合准则中大部分要求。后继将对这些情况进行进一步的研究,以期获得更好的结果。

### 参考文献:

- [1] 潘平奇. 线性规划计算(上)[M]. 北京: 科学出版社, 2016.  
PAN Pingqi. Linear programming calculation (Volume 1)[M]. Beijing: Science Press, 2016.
- [2] 潘平奇. 线性规划计算(下)[M]. 北京: 科学出版社, 2016.  
PAN Pingqi. Linear programming calculation (Volume 2)[M]. Beijing: Science Press, 2016.
- [3] 张建中, 许绍吉. 线性规划[M]. 北京: 科学出版社, 1999.  
ZHANG Jianzhong, XU Shaoji. Linear programming [M]. Beijing: Science Press, 1999.
- [4] 方述诚, 普森普拉. 线性优化及扩展理论与算法[M]. 北京: 科学出版社, 1994.
- [5] MONTEIRO R D C, ADLER I. Interior path following primal-dual algorithms. Part I: Linear programming[J]. Mathematical Programming, 1989, 44(1/2/3): 27-41.
- [6] DARVAY Z, TAKÁCS P R. New method for determining search directions for interior-point algorithms in linear optimization [J]. Optimization Letters, 2017 (1): 1-18.
- [7] KOJIMA M, MEGIDDO N, MIZUNO S. Primal-dual infeasible-interior-point algorithm for linear programming[J]. Mathematical Programming, 1993, 61(1/2/3): 263-280.
- [8] DANHUA Z, MINGWANG Z. A full-newton step infeasible interior-point algorithm for linear complementarity problem[J]. Nonlinear Analysis: Real World Applications, 2011, 12(1): 545-561.
- [9] BAI Y Q, LESAJA G, ROOS C, et al. A class of large-update and small-update primal-dual interior-point algorithms for linear optimization[J]. Journal of Optimization Theory & Applications, 2008, 138(3): 341-359.
- [10] BAI Y Q, GHAMI M E, ROOS C. A comparative study of kernel functions for primal-dual interior-point algorithms in linear optimization[J]. SIAM Journal on Optimization, 2004, 15(1): 101-128.
- [11] YE Yinyu. An  $O(n^3L)$  potential reduction algorithm for linear programming[J]. Math Programming, 1991 (50): 239-258.
- [12] YE Yinyu. On the complexity of approximating a KKT point of quadratic programming[J]. Math Programming, 1998(80): 195-211.
- [13] FOUTAYENI Y E, BOUANANI H E, KHALADI M. An efficient fifth-order method for linear optimization[J]. Journal of Optimization Theory and Applications, 2016, 170(1): 189-204.
- [14] 雍龙泉. 基于上方一致光滑逼近函数的高阶牛顿法求解线性规划[J]. 吉林大学学报(理学版), 2019, 57(2): 265-270.  
YONG Longquan. High order newton method for solving linear programming based on uniform smooth approximation function from above[J]. Journal of Jilin University (Science Edition), 2019, 57(2): 265-270.
- [15] 雍龙泉. 线性规划的原对偶内点算法数值实验初步[J]. 科学技术与工程, 2007(18): 4576-4579.  
YONG Longquan. Primary numerical experiment to the primal-dual interior point algorithm for linear programming [J]. Science Technology and Engineering, 2007(18): 4576-4579.
- [16] 李士勇, 李研, 林永茂. 智能优化算法与涌现计算[M]. 北京: 清华大学出版社, 2019.  
LI Shiyong, LI Yan, LIN Yongmao. Intelligent optimization algorithms and emergent computation[M].

- Beijing: Tsinghua University Press, 2019.
- [17] TUO Shouheng, ZHANG Junying, YONG Longquan, et al. A harmony search algorithm for high-dimensional multimodal optimization problems[J]. Digital Signal Processing: A Review Journal, 2015, 46(11): 151-163.
- [18] YONG Longquan, TUO Shouheng, GAO Kai. Harmony search with teaching-learning algorithm for non-negative linear least square[J]. Journal of Interdisciplinary Mathematics, 2017, 20(8): 1661-1677.
- [19] 雍龙泉, 拓守恒, 史加荣. 约束处理技术及应用[J]. 计算机科学与探索, 2018, 12(6): 1013-1020.
- YONG Longquan, TUO Shouheng, SHI Jiarong. Constraint-handling technique and applications[J]. Journal of Frontiers of Computer Science and Technology, 2018, 12(6): 1013-1020.
- [20] 黎延海, 拓守恒, 雍龙泉. 动态选择策略的和声教与学混合算法[J]. 计算机应用研究, 2019, 36(12): 3679-3684.
- LI Yanhai, TUO Shouheng, YONG Longquan. Hybrid algorithm based on harmony search and teaching-learning-based optimization using dynamic selection strategies[J]. Application Research of Computers, 2019, 36(12): 3679-3684.

(编辑:孙静)