求解 Fisher 市场均衡问题的内点算法

毕红梅，刘妙华，赵学军
（空军工程大学基础部，西安，710051）

摘要 Fisher 市场均衡是经济学中的经典问题，可以用线性权互补问题来表述。通过调整中心方向向可行点偏移得到新的搜索方向以保证可行性，再利用线性搜索寻找满足邻域条件的最大更新参数来设计求解 Fisher 市场均衡问题的算法，分析了算法的可行性，证明了算法的迭代复杂度。数值实验结果表明该算法对求解 Fisher 市场均衡问题是有效的。

关键词 Fisher 市场均衡；线性权互补问题；内点算法；迭代复杂度

DOI 10.3969/j.issn.2097-1915.2022.04.012

中图分类号 O221.1 **文献标志码** A **文章编号** 2097-1915(2022)04-0077-04

An Interior-Point Algorithm for Solving Fisher Market Equilibrium Problems

BI Hongmei, LIU Miaohua, ZHAO Xuejun
(Fundamentals Department, Air Force Engineering University, Xi'an 710051, China)

Abstract The Fisher market equilibrium is a classic problem in economics, which can be formulated as a linear weight complementarity problem. The new search direction is obtained by adjusting the center direction offset to the feasible point to ensure feasibility, and then the linear search is used to find the maximum update parameter that satisfies the neighborhood conditions to design an algorithm to solve Fisher market equilibrium problems. The feasibility of the algorithm is analyzed, and the iterative complexity of the algorithm is proved. Numerical experimental results show that the algorithm is effective for solving Fisher market equilibrium problems.

Key words Fisher market equilibrium; weighted linear complementarity problem; interior-point algorithm; iteration complexity

早些时候，Ye 提出了求解 Fisher 市场均衡问题的原-对偶内点算法[1]。但该算法跟踪的加权中心路径并不连续，从初始可行点出发还需要通过势函数下降算法才能寻找到满足邻域要求的可行点。近年来，Potra 将 Fisher 市场均衡问题抽象为更一般的线性权互补问题，根据已知可行点构造了一条光滑的中心路径，给出了求解线性权互补问题的内点算法[2]。当权向量为 0 时，权互补问题退化为一般的互补问题；对于权向量部分为 0、部分大于 0 的情形，算法设计和分析比一般的互补问题更为复杂。自此以后，求解权互补问题逐渐成为优化领域研究热点之一。文献[3～4]设计了光滑牛顿算法求解权互补问题。文献[5]在一定假设条件下提出了线性权互补问题宽邻域的路径跟踪算法。文献[6～8]提

收稿日期： 2021-11-13
基金项目： 2022 年度空军工程大学基础部科研启动基金
作者简介： 毕红梅（1982—），女，陕西三原人，讲师，硕士，研究方向为最优化算法。E-mail：bihongmei2021@163.com

引用格式： *毕红梅，刘妙华，赵学军. 求解 Fisher 市场均衡问题的内点算法[J]. 空军工程大学学报，2022，23(4)：77-80. BI Hongmei, LIU Miaohua, ZHAO Xuejun. An Interior-Point algorithm for Solving Fisher Market Equilibrium Problems [J]. Journal of Air Force Engineering University, 2022, 23(4): 77-80.*

出了求解线性权互补问题的全牛顿步内点算法。线性权互补问题来源于 Fisher 市场均衡模型，国内外文献重点讨论了线性权互补问题的求解方法和理论证明，算法测试大多自行设计，有些算法的设定条件并不适用于解决 Fisher 市场均衡这一实际问题，一些初步的研究结果可参考文献[9]。

1 Fisher 市场均衡问题

在 Fisher 模型中，市场中包含消费者 C 和生产者 P。消费者 $i\in C$ 有预算 $w_i>0$ 从生产者手中购买商品使得个人效用函数最大化。价格均衡即给商品定价，使得每个消费者买到最多的商品，同时市场出清，即所有的预算消费完并且所有的商品销售完。不失一般性，假定生产者 j 有 1 单位的某种商品出售。

Eisenberg 和 Gale 证明了市场出清价格为下述优化问题的最优拉格朗日乘子[10]：

$$\begin{aligned} &\max \quad \sum_{i\in C} w_i \log u_i \\ &\text{subject to} \sum_{i\in C} x_{ij} = 1, \forall j \in P \\ &u_i - \sum_{j\in P} u_{ij} x_{ij} = 0, \forall i \in C \\ &u_i, x_{ij} \geqslant 0, \forall i, j \end{aligned} \tag{1}$$

式中：$u_{ij}\geqslant 0$ 为给定的消费者 i 对生产者 j 生产商品的效应系数；x_{ij} 为消费者 i 从生产者 j 手中购买的商品数量。

更一般地，式(1)可以改写成：

$$\begin{aligned} &\max \sum_{i=1} w_i \log x_i \\ &\text{subject to } \boldsymbol{Ax} = \boldsymbol{b} \\ &\boldsymbol{x} \geqslant \boldsymbol{0} \end{aligned} \tag{2}$$

假定最优解集非空，则式(2)的 KKT 条件可以写成线性权互补问题：

$$\begin{aligned} &\boldsymbol{Ax} = \boldsymbol{b}, \boldsymbol{x} \geqslant \boldsymbol{0} \\ &-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{s} = 0, \boldsymbol{s} \geqslant \boldsymbol{0} \\ &\boldsymbol{xs} = \boldsymbol{w} \end{aligned} \tag{3}$$

式中：y、s 分别为拉格朗日乘子。

严格可行域定义为：

$$\boldsymbol{F}_{++} = \{(\boldsymbol{x}, \boldsymbol{s}) > 0 : \boldsymbol{A}x = \boldsymbol{b}, \boldsymbol{s} = \boldsymbol{A}^{\mathrm{T}}\boldsymbol{y}\}$$

给定初始严格可行点 $(x_0, y_0, s_0)\in F_{++}$，令 $\boldsymbol{c}=\boldsymbol{x}_0\boldsymbol{s}_0$，构造

$$\boldsymbol{w}(t) = (1-t)\boldsymbol{w} + tc, t \in (0,1] \tag{4}$$

文献[2]将式(4)替换(3)中的第 3 个式子得到新的中心路径：

$$\begin{aligned} &\boldsymbol{Ax} = \boldsymbol{b}, \boldsymbol{x} \geqslant \boldsymbol{0} \\ &-\boldsymbol{A}^{\mathrm{T}}\boldsymbol{y} + \boldsymbol{s} = \boldsymbol{0}, \boldsymbol{s} \geqslant \boldsymbol{0} \\ &\boldsymbol{xs} = \boldsymbol{w}(t) \end{aligned} \tag{5}$$

当 $t\to 0$ 时，$\boldsymbol{w}(t)\to \boldsymbol{w}$ 得到(3)的解。

2 原-对偶内点算法

2.1 搜索方向

令 $\gamma=\min(c)$，$\alpha=\beta\gamma$，$0<\beta<1$，定义邻域：$N(\alpha,t)=\{(\boldsymbol{x},\boldsymbol{y},\boldsymbol{s})\in F_{++} : \|\boldsymbol{xs}-\boldsymbol{w}(t)\|\leqslant \alpha t\}$。

令 $t_+=(1-\theta)t$，$\theta\in(0,1)$。搜索方向由两个方向构成，其中仿射方向$(\Delta^a\boldsymbol{x}, \Delta^a\boldsymbol{y}, \Delta^a\boldsymbol{s})$由下面的系统给出：

$$\begin{aligned} \boldsymbol{A}\Delta^a\boldsymbol{x} &= \boldsymbol{0} \\ -\boldsymbol{A}^{\mathrm{T}}\Delta^a\boldsymbol{y} + \Delta^a\boldsymbol{s} &= \boldsymbol{0} \\ \boldsymbol{s}\Delta^a\boldsymbol{x} + \boldsymbol{x}\Delta^a\boldsymbol{s} &= \boldsymbol{w} - \boldsymbol{xs} \end{aligned} \tag{6}$$

中心方向$(\Delta^c\boldsymbol{x}, \Delta^c\boldsymbol{y}, \Delta^c\boldsymbol{s})$由下述系统给出：

$$\begin{aligned} \boldsymbol{A}\Delta^c\boldsymbol{x} &= \boldsymbol{0} \\ -\boldsymbol{A}^{\mathrm{T}}\Delta^c\boldsymbol{y} + \Delta^c\boldsymbol{s} &= \boldsymbol{0} \\ \boldsymbol{s}\Delta^c\boldsymbol{x} + \boldsymbol{x}\Delta^c\boldsymbol{s} &= \boldsymbol{c} - \boldsymbol{xs} \end{aligned} \tag{7}$$

仿射方向作为预测方向，中心方向作为矫正方向。本文采用的中心方向把迭代点拉向 $\boldsymbol{c}=\boldsymbol{x}_0\boldsymbol{s}_0$ 以保持可行性，而文献[2]中的中心方向希望迭代点向 $\boldsymbol{w}(t)$ 靠近。通过二分法搜索满足条件的最大 θ 值使得新的迭代点：

$$(\boldsymbol{x}_+, \boldsymbol{y}_+, \boldsymbol{s}_+) = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s}) + (1-t_+)(\Delta^a\boldsymbol{x}, \Delta^a\boldsymbol{y}, \Delta^a\boldsymbol{s}) + t_+(\Delta^c\boldsymbol{x}, \Delta^c\boldsymbol{y}, \Delta^c\boldsymbol{s}) \tag{8}$$

属于新的邻域：$N(\alpha,t_+)=\{(\boldsymbol{x}_+,\boldsymbol{y}_+,\boldsymbol{s}_+)\in F_{++} : \|\boldsymbol{x}_+\boldsymbol{s}_+-\boldsymbol{w}(t_+)\|\leqslant \alpha t_+\}$。

2.2 算法

算法 1 线性权互补问题内点算法。

1)输入 $\varepsilon>0$，$\theta\in(0,1)$，$t_0=1$，初始可行点$(x_0, y_0, s_0)\in N(\alpha,t_0)$，令 $k=0$。

2)若 $\|\boldsymbol{x}_k\boldsymbol{s}_k-\boldsymbol{w}\|<\varepsilon$，算法停止；否则求解式(6)、(7)得搜索方向：$(\Delta^a\boldsymbol{x}_k, \Delta^a\boldsymbol{y}_k, \Delta^a\boldsymbol{s}_k)$及$(\Delta^c\boldsymbol{x}_k, \Delta^c\boldsymbol{y}_k, \Delta^c\boldsymbol{s}_k)$。

3)搜索满足邻域条件的 θ 值 θ_k，令：$t_{k+1}=(1-\theta_k)t_k$；$\boldsymbol{x}_{k+1}=\boldsymbol{x}_k+(1-t_{k+1})\Delta^a\boldsymbol{x}_k+t_{k+1}\Delta^c\boldsymbol{x}_k$；$\boldsymbol{y}_{k+1}=\boldsymbol{y}_k+(1-t_{k+1})\Delta^a\boldsymbol{y}_k+t_{k+1}\Delta^c\boldsymbol{y}_k$；$\boldsymbol{s}_{k+1}=\boldsymbol{s}_k+(1-t_{k+1})\Delta^a\boldsymbol{s}_k+t_{k+1}\Delta^c\boldsymbol{s}_k$。

4)令 $k=k+1$，转 2)。

2.3 算法收敛性分析

算法可行需要保证新的迭代点在新的邻域内，并且随着 t 的减少，x_+s_+ 与 w 的距离不断下降，最终满足精度要求。

假定当前迭代点严格可行且满足 $\|\boldsymbol{xs}-\boldsymbol{w}(t)\|\leqslant \alpha t$。

引理 1 $\boldsymbol{xs}\geqslant(\gamma-\alpha)t\boldsymbol{e}$，其中 $\boldsymbol{e}$ 为单位向量。

证明 由 $\boldsymbol{xs} \geqslant \boldsymbol{w}(t) - \alpha t\boldsymbol{e} \geqslant t\boldsymbol{c} - \alpha t\boldsymbol{e} \geqslant (\gamma-\alpha)t\boldsymbol{e}$，即得。令：

$\Delta \boldsymbol{x} = (1-t_+)\Delta^a \boldsymbol{x} + t_+ \Delta^c \boldsymbol{x}$，$\Delta \boldsymbol{s} = (1-t_+)\Delta^a \boldsymbol{s} + t_+ \Delta^c \boldsymbol{s}$。则 $\boldsymbol{x}_+ = \boldsymbol{x} + \Delta \boldsymbol{x}, \boldsymbol{s}_+ = \boldsymbol{s}+\Delta \boldsymbol{s}$，于是：

$$\boldsymbol{s}\Delta \boldsymbol{x} + \boldsymbol{x} \Delta \boldsymbol{s} = (1-t_+)(\boldsymbol{w} - \boldsymbol{xs}) + t_+(\boldsymbol{c} - \boldsymbol{xs}) = (1-t_+) \boldsymbol{w} + t_+ \boldsymbol{c} - \boldsymbol{xs} = \boldsymbol{w}(t_+) - \boldsymbol{xs} \tag{9}$$

进而：

$$\boldsymbol{x}_+ \boldsymbol{s}_+ - \boldsymbol{w}(t_+) = (\boldsymbol{x}+\Delta \boldsymbol{x})(\boldsymbol{s}+\Delta \boldsymbol{s}) - \boldsymbol{w}(t_+) = \boldsymbol{xs} + \boldsymbol{s}\Delta \boldsymbol{x} + \boldsymbol{x}\Delta \boldsymbol{s} + \Delta \boldsymbol{x} \Delta \boldsymbol{s} - \boldsymbol{w}(t_+) = \Delta \boldsymbol{x}\Delta \boldsymbol{s} \tag{10}$$

引理 2 [11] 设 $\boldsymbol{u}, \boldsymbol{v}\in R^n$，$\boldsymbol{u}^{\mathrm{T}} \boldsymbol{v} \geqslant 0$。记 $\boldsymbol{U}=\mathrm{diag}(\boldsymbol{u})$，$\boldsymbol{V} = \mathrm{diag}(\boldsymbol{v})$，则 $\|\boldsymbol{UVe}\| \leqslant 2^{-3/2}\|\boldsymbol{u}+\boldsymbol{v}\|^2$。

引理 3 $\|\boldsymbol{x}_+\boldsymbol{s}_+ - \boldsymbol{w}(t_+)\| \leqslant \frac{\sqrt{2}}{4}\frac{(\theta\|\boldsymbol{w}-\boldsymbol{c}\|+\alpha)^2 t}{(\gamma-\alpha)}$。

证明式(9)两边同乘以 $(\boldsymbol{xs})^{-1/2}$，得：$\boldsymbol{x}^{-1/2}\boldsymbol{s}^{1/2}\Delta \boldsymbol{x} + \boldsymbol{x}^{1/2}\boldsymbol{s}^{-1/2}\Delta \boldsymbol{s} = (\boldsymbol{xs})^{-1/2}(\boldsymbol{w}(t_+)-\boldsymbol{xs})$。

令 $\boldsymbol{u} = \boldsymbol{x}^{-1/2}\boldsymbol{s}^{1/2}\Delta \boldsymbol{x}$，$\boldsymbol{v} = \boldsymbol{x}^{1/2}\boldsymbol{s}^{-1/2}\Delta \boldsymbol{s}$，由式(10)及引理 2 可得：$\|\boldsymbol{x}_+\boldsymbol{s}_+ - \boldsymbol{w}(t_+)\| \leqslant \frac{\sqrt{2}}{4}\frac{\|\boldsymbol{w}(t_+)-\boldsymbol{xs}\|^2}{\min(\boldsymbol{xs})}$。

又：$\|\boldsymbol{w}(t_+) - \boldsymbol{xs}\| \leqslant \|\boldsymbol{w}(t_+)-\boldsymbol{w}(t)+\boldsymbol{w}(t)-\boldsymbol{xs}\| \leqslant \|\boldsymbol{w}(t_+)-\boldsymbol{w}(t)\| + \|\boldsymbol{w}(t)-\boldsymbol{xs}\| \leqslant \theta t\|(\boldsymbol{w}-\boldsymbol{c})\| + t\alpha$。再由引理 1 即得。

要使 $\|\boldsymbol{x}_+\boldsymbol{s}_+ - \boldsymbol{w}(t_+)\|\leqslant \alpha t_+ = (1-\theta)t\alpha$，只需

$$\frac{\sqrt{2}}{4}\frac{(\theta\|\boldsymbol{w}-\boldsymbol{c}\|+\alpha)^2 t}{(\gamma-\alpha)} \leqslant (1-\theta)t\alpha \tag{11}$$

记 $f(\theta) = \frac{\sqrt{2}}{4}\frac{(\theta\|\boldsymbol{w}-\boldsymbol{c}\|+\alpha)^2 t}{(\gamma-\alpha)} - (1-\theta)t\alpha$，由 $\alpha = \beta\gamma$，令 $\beta = 2/3$，则：

$f(0) = \frac{\sqrt{2}\beta}{4(1-\beta)} t\alpha - t\alpha \leqslant (\sqrt{2}/2-1)t\alpha < 0$，同时 $f(1)>0$，因此必存在 $\theta\in(0,1)$ 满足式(11)。

假定 $\theta\|\boldsymbol{w}-\boldsymbol{c}\|\leqslant \alpha/8$，由式(11)可计算出 θ 的下界为

$$\theta_0 = \min\left\{0.105, \frac{\alpha}{8\|w-c\|}\right\}。$$

定理 1 给定 $\varepsilon>0$，初始点 $(\boldsymbol{x}_0,\boldsymbol{y}_0,\boldsymbol{s}_0)\in N(\alpha,t_0)$。若系统(3)存在最优解，则算法 1 至多经过 $\left\lceil \frac{1}{\theta_0}\log\frac{\alpha+\|\boldsymbol{w}-\boldsymbol{x}_0\boldsymbol{s}_0\|}{\varepsilon}\right\rceil$ 次迭代找到 ε－近似解。

证明　$\boldsymbol{x}_+\boldsymbol{s}_+$ 与 $\boldsymbol{w}$ 的距离

$\|\boldsymbol{x}_+\boldsymbol{s}_+-\boldsymbol{w}\| = \|\boldsymbol{x}_+\boldsymbol{s}_+ - \boldsymbol{w}(t_+) + \boldsymbol{w}(t_+) - \boldsymbol{w}\| \leqslant \|\boldsymbol{x}_+\boldsymbol{s}_+ - \boldsymbol{w}(t_+)\| + \|\boldsymbol{w}(t_+)-\boldsymbol{w}\| \leqslant (1-\theta)\alpha t + (1-\theta)t\|\boldsymbol{w}-\boldsymbol{c}\| = (1-\theta)t(\alpha+\|\boldsymbol{w}-\boldsymbol{c}\|) \leqslant (1-\theta_0)t(\alpha+\|\boldsymbol{w}-\boldsymbol{c}\|) = (1-\theta_0)^k(\alpha+\|\boldsymbol{w}-\boldsymbol{c}\|)$。

要使 $\|\boldsymbol{x}_+\boldsymbol{s}_+-\boldsymbol{w}\|\leqslant\varepsilon$，只需 $(1-\theta_0)^k(\alpha+\|\boldsymbol{w}-\boldsymbol{c}\|)\leqslant\varepsilon$。

两边取对数得到：$k\log(1-\theta_0)+\log(\alpha+\|\boldsymbol{w}-\boldsymbol{c}\|)\leqslant \log\varepsilon$。

由 $\log(1+x)\leqslant x, \forall x\in(-1,+\infty)$ 可得：$-k\theta_0+\log(\alpha+\|\boldsymbol{w}-\boldsymbol{c}\|)\leqslant\log\varepsilon$。于是 $k\geqslant\frac{1}{\theta_0}\log\frac{\alpha+\|\boldsymbol{w}-\boldsymbol{c}\|}{\varepsilon}$，证毕。

3 数值实验

本节用算法 1 求解 Fisher 市场均衡问题。假定市场上有 $n_c\geqslant2$ 个消费者，$n_p\geqslant2$ 个生产者，为简单起见，不妨设 $n_c=n_p=p$。记 $\boldsymbol{e}=\mathrm{ones}(1,p)$，$\overline{\boldsymbol{u}_1}=(-u_{11},\cdots,-u_{1p}),\cdots,\overline{\boldsymbol{u}_p}=(-u_{p1},\cdots,-u_{pp})$。

对比式(1)、(2)可知

$\boldsymbol{x}=(x_{11},\cdots,x_{1p},\cdots,x_{p1},\cdots,x_{pp},u_1,\cdots,u_p)^{\mathrm{T}}$，

$\boldsymbol{b}=(\boldsymbol{e},0_{1\times p})^{\mathrm{T}}$，其中 $0_{1\times p}$ 为 1 行 p 列 0 向量，

$\boldsymbol{A}$ 为 $m\times n=2p\times(p^2+p)$ 矩阵

$$\begin{bmatrix} \boldsymbol{e} & & 0 & & \\ & \ddots & & \ddots & \\ & & \boldsymbol{e} & & 0 \\ \overline{\boldsymbol{u}_1} & & 1 & & \\ & \ddots & & \ddots & \\ & & \overline{\boldsymbol{u}_p} & & 1 \end{bmatrix}。$$

$\boldsymbol{w}=(0_{1\times p^2},w_1,\cdots,w_p)$，$w_i$，$u_{ij}$，$i,j=1,2,\cdots,p$ 在区间(0,1)中随机生成，初始点 $(\boldsymbol{x}_0,\boldsymbol{y}_0,\boldsymbol{s}_0)$ 的取法与参考文献[1]一致。

用 MATLAB R2010b 编程实现算法，用 Intel Xeon2 CPU(3.3 GHz)，8 GiB ram 的微机测试。当满足条件 $\|\boldsymbol{xs}-\boldsymbol{w}\|\leqslant10^{-5}$ 时，算法终止。

为说明 Fisher 市场均衡问题求解过程，以最简单的 $p=2$ 时情形为例，即市场上仅有 2 位消费者和 2 位生产者。系数矩阵 $\boldsymbol{A}$ 为 $m\times n=4\times6$ 矩阵

$$\boldsymbol{A}=\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ -0.800\,3 & -0.141\,9 & 0 & 0 & 1 & 0 \\ 0 & 0 & -0.421\,8 & -0.915\,7 & 0 & 1 \end{bmatrix},$$

$\boldsymbol{b}=[1,1,0,0]$，权向量即预算 $\boldsymbol{w}=[0,0,0,0,0.957\,2,0.485\,4]$，

即 2 位消费者预算分别为 0.957 2 和 0.485 4，初始点为 $\boldsymbol{x}_0=[0.5,0.5,0.5,0.5,0.471\,1,0.668\,7]$，$\boldsymbol{y}_0=[2.871\,5,2.871\,5,1.523\,9,1.073\,5]$，$\boldsymbol{s}_0=[1.652\,0,2.655\,3,2.418\,8,1.888\,5,1.523\,9,1.073\,5]$。

用算法 1 求解上述问题，迭代 8 次，耗时 0.001 3 s 后得到最优解 $\boldsymbol{x}^*=[1,0,0,1,0.800\,3,0.915\,7]$，

$\boldsymbol{y}^{*}=[0.9572, 0.4853, 1.1960, 0.5300]$，$\boldsymbol{s}^{*}=[0, 0.7875, 0.2618, 0, 1.1960, 0.5300]$。

结果表明消费者 1 购买了生产者 1 生产的 1 个单位的商品，消费者 2 购买了生产者 2 生产的 1 个单位的商品，相应的效应系数分别为 0.800 3，0.915 7，若商品价格定为 1.196，0.53，对应相乘得到购买商品消费金额分别为 0.957 2，0.485 4，与预算一致，商品售完，市场出清。

下面对每种规模产生 10 个问题，计算 10 次结果的平均迭代次数和平均时间。表 1 和表 2 分别给出了算法 1 和 Ye[1]、Potra[2] 算法的迭代步数和迭代时间。

表 1 Fisher 均衡问题 Ye、Potra 及算法 1 迭代步数比较

m	n	Ye 算法平均迭代步数	Potra 算法平均迭代步数	算法 1 平均迭代步数
10	30	244.2	15.9	16.4
20	110	475.7	25.6	26.2
30	240	703.7	33.2	33.0
40	420	934.0	41.0	41.2
50	650	1 163.6	47.9	46.3

表 2 Fisher 均衡问题 Ye、Potra 及算法 1 迭代时间比较

m	n	Ye 算法平均时间/s	Potra 算法平均时间/s	算法 1 平均时间/s
10	30	0.020 5	0.006 5	0.006 6
20	110	0.317 9	0.079 4	0.073 3
30	240	2.894 2	0.590 1	0.582 1
40	420	18.138 8	3.146 4	3.176 0
50	650	76.731 0	12.739 3	12.439 0

由表 1 和表 2 可见，算法 1 与 Potra 算法在迭代次数及时间上都优于 Ye 算法；当问题规模达到 $m\times n=50\times 650$ 时，算法 1 略优于 Potra 算法。

4 结语

本文从求解 Fisher 市场均衡这一实际问题出发，设计线性权互补问题的内点算法，就中小规模问题给出了数值结果。对于大规模问题，迭代步数和计算时间会有所增加，因此如何调节仿射方向和中心方向参数以提高算法效率，或设计更为高效的宽邻域内点算法是未来的研究重点。

参考文献

[1] YE Y Y. A Path to the Arrow - Debreu Competitive Market Equilibrium[J]. Mathematical Programming, 2008,111(1):315-348.

[2] POTRA F A. Weighted Complementarity Problems - a New Paradigm for Computing Equilibria [J]. SIAM Journal on Optimization, 2012, 22(4):1634-1654.

[3] JIAN Z. A smoothing Newton Algorithm for Weighted Linear Complementarity Problem [J]. Optimization Letters, 2016, 10(3):499-509.

[4] TANG J, ZHANG H. A Nonmonotone Smoothing Newton Algorithm for Weighted Complementarity Problem [J]. Journal of Optimization Theory and Applications, 2021, 189: 679-715.

[5] 韩平，刘长河，尚有林. 单调加权互补问题的路径跟踪算法[J]. 河南师范大学学报(自然科学版)，2018，46(4)：120-124.

[6] 张睿婕，迟晓妮，刘文丽. 线性权互补问题基于核函数的全牛顿步可行内点算法[J]. 桂林电子科技大学学报，2020，40(6)：533-538.

[7] ASADI S, DARVAY Z, LESAJA G, et al. A Full-Newton Step Interior-Point Method for Monotone Weighted Linear Complementarity Problems [J]. Journal of Optimization Theory and Applications, 2020, 186:864-878.

[8] 迟晓妮，张睿婕，刘三阳. 线性权互补问题的新全牛顿步可行内点算法[J]. 应用数学，2021，34(2)：304-311.

[9] 霍东升，徐大川. 竞争市场均衡问题的内点算法[J]. 应用数学学报，2007，30(5)：872-884.

[10] EISENBERG E, GALE D. Consensus of Subjective Probabilities: The Pari-Mutuel Method [J]. Annals of Mathematical Statistics, 1959, 30:165-168.

[11] WRIGHT S J. Primal-Dual Interior-Point Methods [M]. Philadelphia: SIAM, 1997.

（编辑：姚树峰）