

协同免疫量子粒子群算法求非合作博弈 Nash 均衡解

刘露萍 贾文生* 蔡江华

(贵州大学数学与统计学院 贵州 贵阳 550025)

摘要 考虑 n 人非合作博弈 Nash 均衡求解问题。将混合策略意义下的 Nash 均衡转化为最优化问题;把免疫记忆、自我进化、信息共享机制加入量子粒子群算法,通过概率浓度选择公式来保持种群的多样性,提出协同免疫量子粒子群算法。4 个经典的数值算例说明,该算法优于免疫粒子群算法,具有较强的寻优能力和收敛性能。

关键词 Nash 均衡 概率浓度选择 量子粒子群算法 协同免疫量子粒子群算法

中图分类号 TP3 文献标识码 A DOI:10.3969/j.issn.1000-386x.2019.08.035

COEVOLUTIONARY IMMUNE QUANTUM PARTICAL SWARM OPTIMIZATION IN SOLVING NASH EQUILIBRIUM FOR NON-COOPERATIVE GAME

Liu Luping Jia Wensheng* Cai Jianghua

(College of Mathematics and Statistics, Guizhou University, Guiyang 550025, Guizhou, China)

Abstract Considering Nash equilibrium for N -persons' non-cooperative game, Nash equilibrium problems of mixed strategy was converted to optimization problems. We introduced immune memory, self-evolution, information sharing regulation into quantum particle swarm optimization, and maintained the diversity of the population by probability density selection. Based on this, we proposed a coevolutionary immune quantum particle swarm optimization (CIQPSO). The four classical numerical examples show that CIQPSO is superior to the immune particle swarm optimization algorithm, and has stronger optimization ability and convergence performance.

Keywords Nash equilibrium Probability density selection Quantum particle swarm optimization Coevolutionary immune quantum particle swarm optimization

0 引言

群智能算法是模拟自然界的群体行为构造的一类随机优化算法。随着群智能算法应用越来越广泛,近些年来已成为人工智能、社会经济、政治及生物进化等交叉学科的研究热点和前沿之一。不同于遗传算法,群智能算法主要是模拟生物群体智能选择行为的属性,同时蕴含了生物体之间互相学习与合作的特性。传统的数学分析算法,比如 Lemke-Howso 算法^[1]、全局牛顿算法^[2]、单纯形剖分算法^[3]、同伦算法^[4]、分布式原始对偶算法^[5]在工程建设、网络通信、非线性分析、

经济管理各个领域具有明显的优势。但将其用于 Nash 均衡的求解时面临计算复杂度高和计算时间长的问题,这时探究更有效的求 Nash 均衡解的方法是必要的。因此,考虑从群智能方面来求解和解释博弈均衡 Nash 平衡点是一种新的尝试和方法。博弈论的应用广泛而深刻,特别是非合作博弈,正如 2007 年诺贝尔经济学奖获得者 Myerson 在文献[6]中指出的,要“认识到非合作博弈理论的基础与核心地位及合作博弈理论是必不可少的补充作用”。非合作博弈的核心概念便是 Nash 均衡,Roughgarden^[7]指出 Nash 均衡的求解是一个 NP-hard 问题,随着现代智能算法的不断深入研究和发 展,智能算法在解决 NP-hard 问题上体

现了较强的优越性。

1951年, n 人非合作有限博弈 Nash 均衡的存在性被 Nash 证明^[8]。其中 Nash 均衡解并不是唯一的, Nash 也未给出求解 Nash 均衡解的算法。近些年人们对智能算法研究越来越普遍, 智能算法在计算博弈 Nash 均衡问题上显露出了较强的优越性, 学者们也尝试用免疫算法^[9]、自适应小生镜算法^[10]、模拟退火算法^[11]、启发搜索算法^[12]、粒子群优化算法^[13]、烟花算法^[14]、投影梯度算法^[15]等来求解 Nash 均衡问题。随着智能算法的迅速发展, 将 Nash 均衡转化为最优化问题, 依赖智能算法寻优, 成为一种行之有效的方法。因此, 本文在量子粒子群算法中引入免疫记忆、自我进化机制并通过概率浓度选择公式来保持种群的多样性, 建立一种求解 Nash 均衡的新型协同免疫量子粒子群算法, 并通过对 4 个经典数值算例的计算和比较, 说明了本文算法的有效性, 为实际经济生活中的博弈活动提供决策参考。

1 问题描述

设一个 n 人非合作有限博弈模型, 记博弈参与人集合为 $N = \{1, 2, \dots, n\}$, 局中人 i 的纯策略为 $s^i = (s_1^i, s_2^i, \dots, s_{m_i}^i)$, 定义 s^i 上的混合策略集为 $X^i = \{x^i = (x_1^i, x_2^i, \dots, x_{m_i}^i) \mid x_j^i \geq 0, \sum_{j=1}^{m_i} x_j^i = 1\}$ 。当每个局中人 i 选择纯策略 $s_j^i \in s^i, i \in N$, 局中人 i 得到的支付为实数: $P_i(s_1^i, s_2^i, \dots, s_{m_i}^i)$, 其中: $s_{j_1}^1$ 表示局中人 1 选择纯策略; $s_{j_2}^2$ 表示局中人 2 选择纯策略, $\dots, s_{j_n}^n$ 表示局中人 n 选择纯策略时局中人 i 的收益。记作 $X = \prod_{i=1}^n X^i$, X 为所有局中人策略集的笛卡儿乘积, 它表示每个局中人选定一个混合策略后, 形成的一个混合策略组合。非合作有限博弈的一个混合局势可以记为任意 $x = (x^1, x^2, \dots, x^n) \in X$, 当局中人 i 选择混合策略 $x = (x_1^i, x_2^i, \dots, x_{m_i}^i) \in X^i$, 即局中人 i 以概率 x_1^i 选择纯策略 s_1^i, \dots , 以概率 $x_{m_i}^i$ 选择纯策略 $s_{m_i}^i$ 。对于任意的 $i \in N$, 所有局中人的选择是独立的。则在此混合局势下, 局中人 i 得到的期望支付为:

$$f_i(x) = \sum_{j_1=1}^{m_1} \sum_{j_2=1}^{m_2} \dots \sum_{j_n=1}^{m_n} P_i(s_{j_1}^1, s_{j_2}^2, \dots, s_{j_n}^n) \cdot \prod_{i=1}^n x_{j_i}^i$$

所有局中人都是理性的, 都希望自己能够获得最大的利益。因此, 如果任意 $i \in N$, 其中记 $\hat{i} \in N \setminus \{i\}$, $x^* = (x_1^*, x_2^*, \dots, x_n^*) \in X$, 有:

$$f_i(x_i^*, x_{\hat{i}}^*) = \max_{u_i \in X^i} f_i(u_i, x_{\hat{i}}^*)$$

则称 x^* 为此 n 人非合作有限博弈的 Nash 平衡点, 此时每个局中人都不能单独通过改变自己的策略而使自己获得更大的利益。

假设是 2 个局中人的有限非合作博弈(双矩阵博弈): 其中局中人 1 的混合策略为 $x = (x^1, x^2, \dots, x^m)$, 局中人 2 的混合策略为 $y = (y^1, y^2, \dots, y^n)$ 。局中人 1 和局中人 2 的支付矩阵分别为 $A_{m \times n}$ 、 $B_{m \times n}$ 。其期望收益分别为 $x A y^T$ 和 $x^T B y$ 。 (x^*, y^*) 是双矩阵博弈的一个 Nash 均衡的充分必要条件是:

$$\begin{cases} x^* A y^{*T} \geq x A y^{*T} & \forall x \in X \\ x^{*T} B y^* \geq x^{*T} B y & \forall y \in Y \end{cases}$$

2 协同免疫量子粒子群算法的设计

2.1 量子粒子群算法

粒子群优化算法(Particle Swarm Optimization algorithm, PSO)具有迭代搜索寻优和群体智能等特点, 但 2001 年 Bergh 证明了 PSO 算法不能保证收敛到全局最优解甚至是局部最优解^[16]。尽管众多学者分别从算法理论分析、算法的改进方法、算法在各种工程优化领域的应用做了大量的研究工作, 但实际上其优化效果仍是非常有限的。

2004 年, 孙俊等^[17]从量子力学的角度提出了一种新型粒子群优化算法。该算法建立 δ 势阱势能场模型, 提出了具有量子行为的粒子群算法(Quantum - Behaved PSO Algorithm, QPSO)。QPSO 算法在迭代进化计算过程中终究保持两个最优位置: 1) 粒子 $i (i = 1, 2, \dots, M, M$ 为种群规模) 所经历过的个体最好位置(有最好的适应度值)表示为 $P_i(t) = [P_{i,1}(t), P_{i,2}(t), \dots, P_{i,N}(t)]^T (N$ 表示问题维度), 记作 $pbest_i(t)$ 。2) 种群中所有粒子经历过的最好位置表示为 $P_g(t) = [P_{g,1}(t), P_{g,2}(t), \dots, P_{g,N}(t)]^T$, 记作 $gbest(t)$ 。其粒子更新公式为:

$$P_i(t+1) = \begin{cases} X_i(t+1) & f[X_i(t+1)] > f[P_i(t)] \\ P_i(t) & f[X_i(t+1)] \leq f[P_i(t)] \end{cases} \quad (1)$$

$$P_g(t+1) = \max_{1 \leq i \leq M} P_i(t+1) \quad (2)$$

式中: 粒子 i 的吸引子是由 $pbest_i$ 和 $gbest$ 之间的随机点 $p_i(t) = [p_{i,1}(t), p_{i,2}(t), \dots, p_{i,N}(t)]^T$ 产生, 坐标为: $p_{i,j}(t) = \varphi_{i,j}(t) \cdot P_{i,j}(t) + [1 - \varphi_{i,j}(t)] \cdot P_{g,j}(t)$ (3) 式中: $\varphi_{i,j}(t) = rand(), j = 1, 2, \dots, N, rand()$ 表示产生一个 $[0, 1]$ 之间服从均匀分布的随机数。

在 QPSO 算法中, 用波函数来描述量子粒子空间中粒子的位置。对于波函数, 通过求解粒子在 δ 势阱场中运动的定态 Schrödinger 方程的方法得到粒子在

量子空间中某一点出现的概率密度函数,再通过用 MonteCarlo 方法得到粒子位置更新公式:

$$X_{i,j}(t+1) = p_{i,j}(t) + \text{rand}(t) \cdot \alpha \cdot |C_j(t) - X_{i,j}(t)| \cdot \ln\left(\frac{1}{u_{i,j}(t)}\right) \quad (4)$$

式中: $u_{i,j}(t)$ 是 $[0, 1]$ 之间服从均匀分布的随机数。 α 为收缩-扩张系数,一般采用线性减小的方式: $\alpha = \alpha_{\min} + T \times \frac{\alpha_{\max} - \alpha_{\min}}{T - t}$ (α_{\max} 为最大收缩扩张系数, α_{\min} 为最小收缩扩张系数, T 为最大迭代次数, t 为当前迭代次数)。

在 QPSO 算法中,引入 $mbest(t)$ 表示平均最好位置,记为 $C(t)$,即:

$$C(t) = [C_1(t), C_2(t), \dots, C_N(t)]^T = \frac{1}{M} \sum_{i=1}^M P_i(t) = \left[\frac{1}{M} \sum_{i=1}^M P_{i,1}(t), \frac{1}{M} \sum_{i=1}^M P_{i,2}(t), \dots, \frac{1}{M} \sum_{i=1}^M P_{i,N}(t) \right]^T \quad (5)$$

2.2 协同免疫量子粒子群算法

免疫算法具有抗原识别、免疫记忆、抗体抑制和促进的特点,是抗体对抗抗原的过程。本文的协同免疫量子粒子群算法 (Cooperative Immune Quantum Particle Swarm Optimization Algorithm, CIQPSO) 将免疫记忆、自我调节机制引入到量子粒子群算法,所有粒子之间信息共享、共同进化。为了避免丢失一些适应度差但保持较好进化趋势的粒子,文中引入概率浓度选择公式来保持粒子种群的多样性。在 CIQPSO 中,目标函数和约束条件被视作抗原,问题的解被视作抗体(粒子)。在算法的搜索空间中,每个抗体都表示问题的一个解。粒子在可行解空间搜索过程中通过其自身位置最优信息和群体位置最优信息不断地调整自己的当前位置,并向全局最优解靠拢。

对于 n 人有限非合作博弈混合策略的最优化问题, Nash 均衡点的函数值最小,适应度函数值最好。在 CIQPSO 算法中, x_i 表示粒子 i 的位置, $f(x_i)$ 表示粒子 i 的适应度函数值, $i = 1, 2, \dots, M + Q$ 。集合 X 由 $M + Q$ 个抗体组成,定义粒子 $f(x_i)$ 到集合 X 的距离如下:

$$d(x_i) = \sum_{j=1}^{M+Q} |f(x_i) - f(x_j)| \quad i = 1, 2, \dots, M + Q$$

由文献[18],我们将第 i 个粒子的浓度定义如下:

$$D(x_i) = \frac{1}{\sum_{j=1}^{M+Q} |f(x_i) - f(x_j)|} \quad i = 1, 2, \dots, M + Q$$

定义基于上述粒子浓度的概率选择公式如下:

$$P(x_i) = \frac{\frac{1}{D(x_i)}}{\sum_{i=1}^{M+Q} \frac{1}{D(x_i)}} = \frac{\sum_{j=1}^{M+Q} |f(x_i) - f(x_j)|}{\sum_{i=1}^{M+Q} \sum_{j=1}^{M+Q} |f(x_i) - f(x_j)|} \quad (6)$$

对于一般的博弈模型,所有局中人都追求其自身利益的最大化,其最优解也将遵循一定的游戏规则最终达到一个动态的平衡。用 CIQPSO 算法求解 n 人非合作有限博弈的 Nash 均衡问题时,将每一个 Nash 均衡解视为一个粒子,此时所有博弈方皆采取混合策略,即完全重复地进行博弈,博弈方在其纯策略空间上服从概率分布。支付函数则为各参与人的期望,它是关于各局中人选择不同的纯策略的概率的多重线性形式。算法中的每一个粒子由所有局中人的混合策略表示,即 $x = (x^1, x^2, \dots, x^n)$ 。定义 CIQPSO 算法中的适应度函数如下:

$$f(x) = \sum_{i=1}^n \max\{f_i(x \parallel s_j^i) - f_i(x), 0\} \quad 1 \leq j \leq m_i$$

式中: $x \parallel s_j^i$ 表示仅有局中人 i 用纯策略 s_j^i 替换混合策略 x 中自己的策略,其他局中人都改变各自的策略。若为最优化问题 $\min f(x)$, $f(x)$ 为目标函数,粒子函数值越小其适应度值越好,该粒子越适应环境 e 。

因此,由 Nash 均衡的定义知: x^* 是 n 人非合作有限博弈混合策略意义下的一个 Nash 均衡解的充分必要条件是: $\exists x^*, \text{s.t. } f(x^*) = 0, \forall x \neq x^*, f(x) > 0$ 。

设为双矩阵 ($m \times n$ 维) 博弈,算法中每个粒子由两个局中人的策略混合,即 $z = (x, y)$ 。局中人 1、局中人 2 的混合策略集分别表示为:

$$X = \{x = (x^1, x^2, \dots, x^m) \in R_+^m : x^i \geq 0, \sum_{i=1}^m x^i = 1; i = 1, 2, \dots, m\}$$

$$Y = \{y = (y^1, y^2, \dots, y^n) \in R_+^n : y^j \geq 0, \sum_{j=1}^n y^j = 1; j = 1, 2, \dots, n\}$$

即局中人 1、局中人 2 分别在支付矩阵 A 、 B 上所有概率分布的集合分别为 X 、 Y 。双矩阵博弈适应度函数定义如下:

$$f(z) = \max\{\max_{1 \leq i \leq m} (A_i y^T - x A y^T), 0\} + \max\{\max_{1 \leq j \leq n} (x^T B_j - x^T B y), 0\}$$

式中: A_i 为矩阵 $A_{m \times n}$ 的第 i 行; B_j 为矩阵 $B_{m \times n}$ 的第 j 列。同理,若混合局势 z^* 是该双矩阵博弈的一个 Nash 均衡解的充分必要条件是: $\exists z^* = (x^*, y^*), \text{s.t. } f(z^*) = 0, \forall z \neq z^*, f(z) > 0$ 。

2.3 协同免疫量子粒子群算法实现

协同免疫量子粒子群算法实现步骤如下:

Step1 参数初始化。确定最大迭代次数 T_{\max} 、精度 ε 、群体规模 M ;

Step2 用式(5)计算粒子平均最好位置 $mbest$;

Step3 用式(1)和式(2)更新个体最好位置和全

体最好位置;

Step4 根据适应度函数计算每个粒子适应度值,找到个体极值 p_{best} 和全体最好极值 g_{best} ,并将 g_{best} 对应的粒子位置存入记忆库;

Step5 Q 个粒子随机生成;

Step6 根据粒子的概率浓度选择式(6),从 $M + Q$ 个粒子中选取 M 个粒子;

Step7 用记忆库中的粒子代替粒子群中适应度较差的粒子,生成新一代粒子群 p_1 的同时再进行下一次迭代;

Step8 用式(3)计算粒子群 p_1 得到一个随机位置;

Step9 用式(4)计算粒子新位置;

Step10 判断最大迭代次数或精度是否达到要求?是则停止迭代,否则返回 Step3。

算法实现流程如图 1 所示。

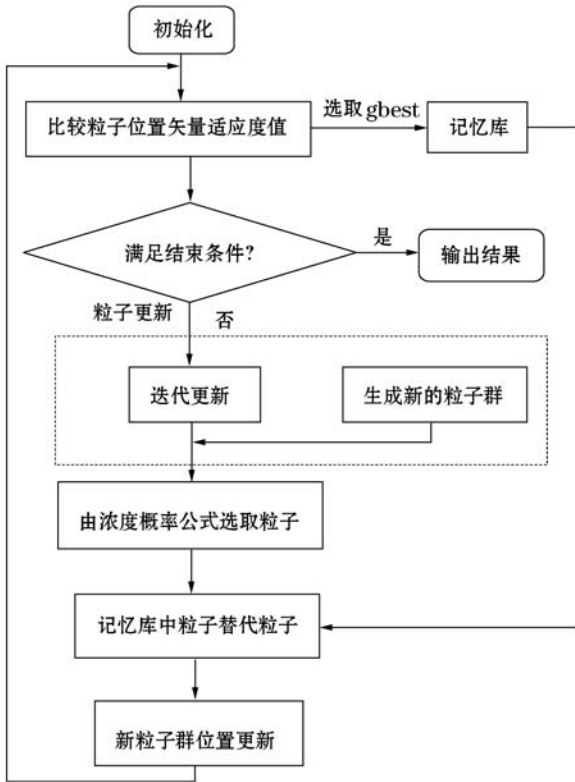


图 1 CIQPSO 算法流程图

2.4 算法性能的评价

协同免疫量子粒子群算法是一种生物演化的群体智能算法,它与遗传算法效仿生物界“物竞天择、适者生存”的演化法则有许多相似之处。所以,可以借鉴 Dejong 在文献[19]中提出的定量分析方法,用离线性能来测试算法的收敛性。

定义 1 $s^* : X \rightarrow R, s_e^*(x)$ 为在环境 e 下策略 x 的离线性能,即 $s_e^*(x) = \frac{1}{T} \sum_{i=1}^T f_e^*(x)$, $f_e^*(x)$ 表示环境 e

的最佳适应度值, $f_e^*(x) = \min \{f_e(1), f_e(2), \dots, f_e(N)\}$ 。在算法运行过程中,离线性能是各进化代数最佳适应度值的累和平均。

2.5 协同免疫量子粒子群位置收敛性证明

定理 1 在 N 维搜索空间中,按照式(4)进化的粒子 i 的位置依概率收敛到其吸引子 $p_i(t) = [p_{i,1}(t), p_{i,2}(t), \dots, p_{i,N}(t)]^T$ 的充分必要条件是:每一维坐标 $X_{i,j}(t)$ 都依概率收敛于 $p_{i,j}$ 。

证明 必要性: $X_i(t) \xrightarrow{P} p_i(p_{i,1}, p_{i,2}, \dots, p_{i,N})$ 对任意 $\varepsilon > 0$, 有。

$$\lim_{t \rightarrow \infty} P\{|X_i(t) - p_i| \geq \varepsilon\} = 0$$

① 对于任意的 $j \in \{1, 2, \dots, N\}$, 恒有

$$|X_{i,j}(t) - p_{i,j}| < \varepsilon$$

成立,则 $\lim_{t \rightarrow \infty} P\{|X_{i,j}(t) - p_{i,j}| \geq \varepsilon\} = 0$ 显然成立。

② 当 $|X_i(t) - p_i| \geq \varepsilon$ 时,存在 $j \in \{1, 2, \dots, N\}$ 使得 $|X_{i,j}(t) - p_{i,j}| \geq \varepsilon$, 此时有以下事件包含关系成立:

$$\{|X_i(t) - p_i| \geq \varepsilon\} \supset \{|X_{i,j}(t) - p_{i,j}| \geq \varepsilon\}$$

从而:

$$P\{|X_i(t) - p_i| \geq \varepsilon\} \supset P\{|X_{i,j}(t) - p_{i,j}| \geq \varepsilon\}$$

两边求极限得:

$$0 = \lim_{t \rightarrow \infty} P\{|X_i(t) - p_i| \geq \varepsilon\} \supset \lim_{t \rightarrow \infty} P\{|X_{i,j}(t) - p_{i,j}| \geq \varepsilon\}$$

$$\lim_{t \rightarrow \infty} P\{|X_{i,j}(t) - p_{i,j}| \geq \varepsilon\} = 0$$

即 $X_{i,j}(t) \xrightarrow{P} p_{i,j}$ 。

充分性: 设 $X_{i,j}(t) \xrightarrow{P} p_{i,j}$ 即是对于每一个 $j \in \{1, 2, \dots, N\}$, 以及任意 $\varepsilon > 0$, 都有:

$$\lim_{t \rightarrow \infty} P\{|X_{i,j}(t) - p_{i,j}| \geq \frac{\varepsilon}{\sqrt{N}}\} = 0$$

如果 $|X_{i,j}(t) - p_{i,j}| \geq \frac{\varepsilon}{\sqrt{N}}$, 必有:

$$|X_i(t) - p_i| = \left(\sum_{j=1}^N |X_{i,j}(t) - p_{i,j}|^2 \right)^{\frac{1}{2}} \geq \left[N \cdot \left(\frac{\varepsilon}{\sqrt{N}} \right)^2 \right]^{\frac{1}{2}} = \varepsilon$$

从而 $\{|X_{i,j}(t) - p_{i,j}| \geq \frac{\varepsilon}{\sqrt{N}}\} \supset \{|X_i(t) - p_i| \geq \varepsilon\}$

因此 $P\{|X_{i,j}(t) - p_{i,j}| \geq \frac{\varepsilon}{\sqrt{N}}\} \supset P\{|X_i(t) - p_i| \geq \varepsilon\}$

两边求极限可得到:

$$0 = \lim_{t \rightarrow \infty} P\{|X_{i,j}(t) - p_{i,j}| \geq \frac{\varepsilon}{\sqrt{N}}\} \supset \lim_{t \rightarrow \infty} P\{|X_i(t) - p_i| \geq \varepsilon\}$$

ε 故必有:

$$\lim_{t \rightarrow \infty} P\{|X_i(t) - p_i| \geq \varepsilon\} = 0$$

即 $X_i(t) \xrightarrow{p} p_i$ 。

这表明在 N 维空间中,满足式(4)的单个粒子位置的收敛性可以归结为一维空间的粒子收敛性,粒子 i 的位置收敛,也就意味着算法解的收敛性。这也说明了粒子 i 的位置依概率收敛到吸引子充要条件是 $\lim_{t \rightarrow \infty} [\text{rand}(t) \cdot \alpha \cdot |C_j(t) - X_{i,j}|] = 0$ 。

3 数值算例

分别考虑 4 个不同的算例,例 1 是文献[12]和文献[20]共同给出的一个博弈算例。例 2 是文献[21],此问题至少有 6 个解。例 3 是文献[22]中非合作双矩阵博弈模型,例 3 的对策问题只有唯一解。例 4 表示例 1 推广到 10×10 阶高维矩阵。分别用协同免疫量子粒子群算法求解 4 个算例,算法中参数值设置为: $M=20, Q=10$,最大迭代次数的参数设置为 150,例 1 和例 4 适应度函数精度为 $\varepsilon=10^{-4}$,例 2 和例 3 适应度函数精度为 $\varepsilon=10^{-2}$ 。

例 1 考虑 3×3 非合作矩阵博弈 $\Gamma(X_1, Y_1, A_1, B_1)$ 对策 Nash 均衡点,支付矩阵如下:

$$A_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad B_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

例 2 考虑 4×4 非合作矩阵博弈 $\Gamma(X_2, Y_2, A_2, B_2)$ 对策的 Nash 均衡点,支付矩阵如下:

$$A_2 = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 1 & 7 & 8 \\ 8 & 5 & 4 & 2 \\ 7 & 2 & 9 & 6 \end{pmatrix} \quad B_2 = \begin{pmatrix} 2 & 3 & 4 & 5 \\ 1 & 2 & 5 & 8 \\ 5 & 9 & 4 & 7 \\ 6 & 3 & 10 & 3 \end{pmatrix}$$

例 3 考虑 3×4 非合作矩阵博弈 $\Gamma(X_3, Y_3, A_3, B_3)$ 对策的 Nash 均衡点,支付矩阵如下:

$$A_3 = \begin{pmatrix} 2 & -3 & 0 & 6 \\ 4 & 9 & -4 & 7 \\ 3 & -1 & 1 & 6 \end{pmatrix} \quad B_3 = \begin{pmatrix} 4 & 7 & 5 & -10 \\ 2 & -3 & 4 & -1 \\ 3 & -1 & 4 & -2 \end{pmatrix}$$

例 4 考虑 10×10 双矩阵博弈 $\Gamma(X_4, Y_4, A_4, B_4)$ 对策的 Nash 均衡点,支付矩阵如下:

$$A_4 = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} \quad B_4 = \begin{pmatrix} 0 & 0 & \dots & 1 \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & 0 \end{pmatrix}$$

将上面 4 个例子的数据代入协同免疫量子粒子群算法中得到的运行结果如表 1 - 表 4 所示。

表 1 $\Gamma(X_1, Y_1, A_1, B_1)$ 运行结果

计算次数	迭代次数	局中人 1 混合策略	局中人 2 混合策略	适应度函数值
1	109	(0.333 3, 0.333 4, 0.333 3)	(0.333 4, 0.333 2, 0.333 4)	0.999 89e-004
2	186	(0.333 3, 0.333 3, 0.333 4)	(0.333 4, 0.333 3, 0.333 3)	0.514 73e-004
3	166	(0.333 4, 0.333 3, 0.333 3)	(0.333 3, 0.333 3, 0.333 4)	0.956 06e-004
4	105	(0.333 3, 0.333 4, 0.333 3)	(0.333 4, 0.333 3, 0.333 3)	0.532 77e-004
5	95	(0.333 4, 0.333 4, 0.333 2)	(0.333 3, 0.333 4, 0.333 3)	0.936 05e-004
6	65	(0.333 3, 0.333 4, 0.333 3)	(0.333 3, 0.333 3, 0.333 4)	0.88725e-005

表 2 $\Gamma(X_2, Y_2, A_2, B_2)$ 运行结果

计算次数	迭代次数	局中人 1 混合策略	局中人 2 混合策略	适应度函数值	运行时间
1	8	(0, 1.000 0, 0, 0)	(0, 0, 0, 1.000 0)	5.353 5e-004	0.0584 9
2	9	(0, 0, 0, 1.000 0)	(0, 0, 1.000 0, 0)	2.243 6e-004	0.064 33
3	9	(0, 0, 0, 1.000 0)	(0, 0, 1.000 0, 0)	9.884 5e-004	0.055 71
4	13	(0, 0.750 0, 0.250 0, 0)	(0, 0.250 0, 0, 0.750 0)	3.547 8e-004	0.082 43
5	16	(0, 0.950 0, 0, 0.050 0)	(0.250 0, 0, 0.750 0, 0)	7.736 9e-004	0.115 30
6	17	(0, 0.500 0, 0.000 0, 0)	(0, 0, 0.000 0, 0.500 0)	6.774 2e-004	0.107 45

表 3 $\Gamma(X_3, Y_3, A_3, B_3)$ 运行结果

计算次数	迭代次数	局中人 1 混合策略	局中人 2 混合策略	适应度函数值	运行时间
1	3	(0, 0.714 2, 0.285 8)	(0.833 5, 0, 0.166 5, 0)	0	0.0584 9

表 4 $\Gamma(X_4, Y_4, A_4, B_4)$ 运行结果

计算次数	迭代次数	局中人 1 混合策略	局中人 2 混合策略	适应度函数值
1	1	(0.120 1, 0.149 0, 0.052 0, 0.145 4, 0.053 3, 0.129 6, 0.057 6, 0.081 3, 0.062 7, 0.149 0)		
2	2	(0.094 1, 0.098 7, 0.121 1, 0.080 3, 0.109 8, 0.068 1, 0.093 8, 0.134 7, 0.130 3, 0.069 1)		
3	1	(0.105 5, 0.059 9, 0.091 7, 0.091 3, 0.106 3, 0.082 4, 0.103 1, 0.148 4, 0.089 1, 0.122 3)		
4	6	(0.098 8, 0.106 5, 0.053 4, 0.086 3, 0.117 9, 0.136 7, 0.079 7, 0.085 2, 0.118 0, 0.117 5)		
5	3	(0.078 8, 0.104 1, 0.119 1, 0.103 6, 0.058 2, 0.127 0, 0.1388, 0.076 5, 0.067 0, 0.126 9)		
6	3	(0.089 8, 0.102 0, 0.078 2, 0.100 9, 0.127 4, 0.127 2, 0.078 7, 0.066 3, 0.091 4, 0.138 1)		
		局中人 2 混合策略		适应度函数值
		(0.129 8, 0.130 5, 0.052 4, 0.078 4, 0.129 3, 0.128 8, 0.084 5, 0.040 8, 0.087 1, 0.138 4)		0.078 8-004
		(0.089 9, 0.085 6, 0.111 0, 0.062 6, 0.106 8, 0.141 7, 0.090 7, 0.114 4, 0.092 1, 0.105 2)		0.076 5-004
		(0.079 1, 0.064 1, 0.139 9, 0.066 8, 0.090 3, 0.077 4, 0.130 5, 0.087 5, 0.148 1, 0.116 3)		0.095 9-004
		(0.121 6, 0.115 9, 0.122 5, 0.105 5, 0.055 5, 0.115 2, 0.129 0, 0.115 4, 0.052 2, 0.067 2)		0.067 2-004
		(0.040 5, 0.051 0, 0.143 8, 0.124 6, 0.150 5, 0.095 4, 0.147 6, 0.062 7, 0.088 2, 0.095 7)		0.085 8-004
		(0.087 2, 0.044 5, 0.115 2, 0.089 1, 0.087 2, 0.077 9, 0.149 1, 0.063 3, 0.157 2, 0.129 3)		0.961 0-004

例 1,由 6 次实验可知,用 CIQPSO 算法求得该博弈的近似解为(0.333 3, 0.333 3, 0.333 3;0.333 3, 0.333 3, 0.333 3),平均只需进化到 121 代。优于免疫粒子群算法计算的 288 代结果^[9],也优于基本粒子群算法计算的 376 代结果^[23],更优于遗传算法计算的

400 代结果^[20]。所以,本文的协同免疫量子粒子群算法的收敛速度确实得到了很大程度的改进。在相同的计算机上,与免疫粒子群算法比较,协同免疫量子粒子群算法收敛速度更快,迭代次数更少。其离线性能如图 2 所示。

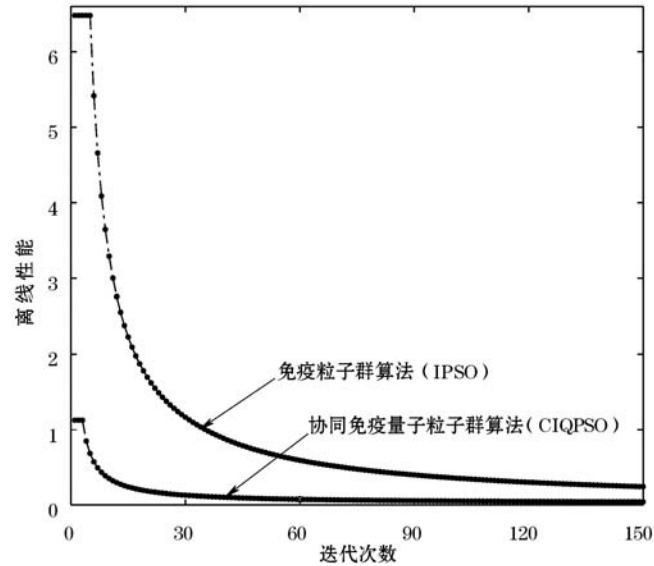


图 2 协同免疫量子粒子算法与免疫粒子群算法求解 $\Gamma(X_1, Y_1, A_1, B_1)$ 的离线性能比较

由图 2 可知,两条曲线分别表示 CIQPSO 算法和 IPSO 算法求解例 1 博弈均衡解的离线性能曲线。CIQPSO 算法收敛比较快,明显优于 IPSO 算法,又因文献[8]给出的 IPSO 算法求解博弈的离线性能优于文献[12]给出的 PSO 算法。所以,本文提出的 CIQPSO 算法优于文献[23]和文献[9]提出的算法。另外,当迭代次数到达 121 代左右时,离线性能基本趋近于 0,说明 CIQPSO 算法求出的近似解基本接近精确解,具有较好的收敛性能。

例 2,运行 6 次实验结果输出的 6 个解分别属于 6 个不同的精确解,且 6 个精确解皆是不同的 Nash 均衡解。协同免疫量子粒子群算法求解例 2 平均进化到 12 代就得到该博弈的 6 个不同的 Nash 均衡解。6 次运行结果的平均时间为 0.200 947 秒,可知该算法的计算时间精度优于文献[10]。而文献[13]给出的算法,需要运行 30 或 40 次才能求出 5 个不同的精确解,事实上,文献[24]中反复运行算法求出的多个 Nash 均衡解具有很大的随机性,并不能保证每次运行能得到不同的 Nash 均衡解,很有可能运行 30 次以后只得到同一个 Nash 均衡解。文献[9]虽只需运行 1 次主算法,却不能得到所有 Nash 均衡解,在其计算过程中,还需多次调用粒子群优化算法来调整小生镜半径,计算

的空间复杂度和时间复杂度都比协同免疫量子粒子群算法要求更高。其离线性能如图 3 所示。

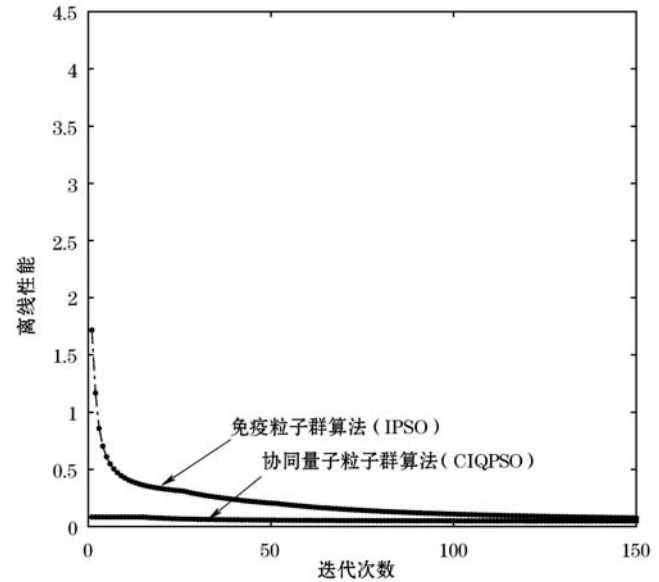


图 3 协同免疫量子粒子算法与免疫粒子群算法求解 $\Gamma(X_2, Y_2, A_2, B_2)$ 的离线性能比较

由图 3 知,两条曲线分别表示 CIQPSO 算法和 IPSO 算法求解例 2 博弈均衡解的离线性能曲线。CIQPSO 算法收敛比较快,明显要优于 IPSO 算法。所以,本文提出的 CIQPSO 算法优于 IPSO 算法。

例 3,只需 1 次实验运行结果就得到例 3 博弈的精确解为 $(0, 0.714 2, 0.285 8; 0.833 5, 0, 0.166 5, 0)$,运行时间只需 0.058 59 秒。这表明该算法的计算精确度高,收敛速度快。

例 4,针对策略为 10×10 阶的高维双矩阵博弈,本文提出的 CIQPSO 算法平均进化到 2.6 代后得到该博弈的近似解为 $(0.097 9, 0.103 4, 0.085 9, 0.101 3, 0.095 5, 0.111 8, 0.091 9, 0.098 7, 0.093 1, 0.120 5; 0.091 4, 0.081 9, 0.114 1, 0.087 8, 0.103 2, 0.106 1, 0.121 9, 0.080 7, 0.104 2, 0.108 7)$ 。进一步探讨了该算法可以进一步推广到求解高维的双矩阵博弈。

由例 1 - 例 4 数值算例的计算和比较可知,用 CIQPSO 算法在求解 Nash 均衡方面具有较好的性能,在算法精度、迭代次数、迭代时间都比 IPSO 算法有了进一步提高,算法的空间复杂度和时间复杂度都得到了一定改善。另外,粒子在随机生成的过程中不依赖于初始点的选取,并通过概率浓度选择公式来保持种群多样性,避免丢失可能成为最优解的潜在解,因此更有可能找寻到全局最优解,避免陷入局部最优解的早熟现象。

4 结 语

本文提出的 CIQPSO 算法将免疫记忆、自我进化引入 QPSO 算法中,并通过概率浓度选择公式来保持种群的多样性。将该算法应用到求解 n 人有限非合作博弈中,通过实验可以看出改进后的算法大大节约了收敛的时间,提高了算法效率,较好地克服了量子粒子群算法的早熟现象。另外,CIQPSO 算法仍是一种群体智能迭代算法,在算法迭代搜索过程中,每一个粒子记录自身的最优位置,并向其他粒子学习。通过粒子的个性化学习和彼此间的协作,促使群体不断向问题最优解逼近,同时所有局中人都会向个体极值和群体极值学习,最终趋向博弈的均衡点。下一步研究中,可将该算法应用于更加复杂的博弈求解问题。

参 考 文 献

- [1] Nabatova D S. On the convergence of the Lemke-Howson algorithm for Bi-Matrix games [J]. *Journal of Mathematical Sciences*, 2016, 216(5): 702 - 715.
- [2] Chen C M, Hu H L. Global existence of real roots and random Newton flow algorithm for nonlinear system of equations [J]. *Science China Mathematics*, 2017, 60 (7): 1341 - 1352.
- [3] Karami F, Ziad L, Sadik K. A splitting algorithm for a novel regularization of Perona-Malik and application to image restoration [J]. *EURASIP Journal on Advances in Signal Processing*, 2017, 2017(1): 46 - 55.
- [4] Shah K, Singh T. The Modified Homotopy Algorithm for Dispersion Phenomena [J]. *International Journal of Applied & Computational Mathematics*, 2017, 3(S1): 785 - 799.
- [5] Peng Y, Lacra P. A distributed primal-dual algorithm for computation of generalized Nash equilibria via operator splitting methods [C] // *IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017.
- [6] Myerson R B. *Game Theory: Analysis of Conflict* [M]. Beijing: China Economic Press, 2001.
- [7] Roughgarden T. Computing equilibria: a computation complexity perspective [J]. *Economic Theory*, 2010, 42 (1): 193 - 236.
- [8] Nash J. Non-cooperative games [J]. *Annals of Mathematics*, 1951, 54 (2): 286 - 295.
- [9] 贾文生,向淑文,杨剑锋,等. 基于免疫粒子群算法的非合作博弈 Nash 均衡问题求解 [J]. *计算机应用研究*, 2012, 29(1): 28 - 31.
- [10] 贾文生,向淑文,杨剑锋. 基于自适应小生镜粒子群算法的多重 Nash 均衡 [J]. *计算机应用与软件*, 2015, 32 (1): 247 - 250.
- [11] Oliveira H, Petraglia A. Establishing Nash equilibria of strategic games: a multistart Fuzzy Adaptive Simulated Annealing approach [J]. *Applied Soft Computing*, 2014, 19(6): 188 - 197.
- [12] 隗立涛,修乃华. 基于启发搜索算法的纳什均衡计算 [J]. *北京交通大学学报*, 2007, 36(3): 58 - 62.
- [13] 王昱,章卫国,傅莉,等. 基于精英改选机制的粒子群算法的空战纳什均衡策略逼近 [J]. *控制理论与应用*, 2015, 32(7): 857 - 865.
- [14] 杨彦龙,向淑文,夏顺友,等. 基于烟花算法的非合作博弈 Nash 均衡问题求解 [J]. *计算机应用与软件*, 2018, 35 (3): 215 - 218.
- [15] Belgioioso G, Grammatico S. Projected-gradient algorithms for generalized equilibrium seeking in aggregative games are preconditioned Forward-Backward methods [EB]. arXiv: 1803.10441, 2018.
- [16] Bergh F V D. An analysis of particle swarm optimizers [D]. Pretoria: University of Pretoria, 2001: 78 - 126.
- [17] Sun J, Feng B, Xu W B. Particle swarm optimization with particles having quantum behavior [C] // *Proceedings of the 2004 Congress on Evolutionary Computation*. Piscataway, NJ: IEEE, 2004: 325 - 331.
- [18] Liu J, Sun J, Xu W B, et al. Quantum-behaved Particle Swarm Optimization Based on Immune Memory and Vaccination [C] // *Proceedings Computing USA, IEEE*, 2006: 453 - 456.
- [19] Dejong K A. Analysis of the behavior of a class of genetic adaptive system [D]. Ann Arbor: University Michigan, 1975.
- [20] 陈士俊,孙永广,吴宗鑫. 一种求解 Nash 均衡解的遗传算法 [J]. *系统工程*, 2001, 19(5): 52 - 55.
- [21] 胡山鹰,陈丙珍,何小荣. 连续变量全局优化的模拟退火算法 [J]. *系统工程理论与实践*, 1995, 15(9): 73 - 80.
- [22] 谢政. 博弈论 [M]. 北京: 科学出版社, 2010: 168 - 189.
- [23] 余谦,王先甲. 基于粒子群优化求解纳什均衡的演化算法 [J]. *武汉大学学报*, 2006, 52(1): 25 - 29.
- [24] 瞿勇,张建军,宋新业. 多重纳什均衡解的粒子群优化算法 [J]. *运筹与管理*, 2010, 19(2): 52 - 55.