# 一种面向多 Agent 交互的博弈 Nash 均衡求解方法*)

李 劲[1] 岳 昆[2] 刘惟一[2]

(云南大学软件学院软件工程系 昆明 650091)[1] (云南大学信息学院计算机科学与工程系 昆明 650091)[2]

**摘 要** 现有的图型博弈 Nash 均衡求解方法基本是在离散化剖面空间中搜索求解,最终只能得到近似 Nash 均衡。针对现有求解方法存在的不足,把求解图型博弈的 Nash 均衡看作是连续策略空间中的函数优化问题,定义 Agents 在策略剖面中的效用偏离度之和为优化目标,其最优解就是博弈的 Nash 均衡。本文基于对实例的分析指出目标函数下降梯度的计算可归结为一组线性规划,进而提出一种求解图型博弈 Nash 均衡的新型梯度下降算法。算法分析及实验研究表明,对于多 Agent 交互模型中的相关问题,本文提出的方法可求解任意图结构图型博弈 Nash 均衡,对于大规模图型博弈也有较好的求解精度和求解效率。

**关键词** 多 Agent 交互模型,图型博弈,Nash 均衡,线性规划,梯度下降算法

## A Method for Solving Nash Equilibrium of the Game Oriented to Multi-agent Interactions

LI Jin[1] YUE Kun[2] LIU Wei-Yi[2]

(Department of Software Engineering, School of Software, Yunnan University, Kunming 650091)[1]

(Department of Computer Science and Engineering, School of Information Science and Engineering, Yunnan University, Kunming 650091)[2]

**Abstract** Among the existing methods for computing the Nash Equilibrium of graphical games, agents play discretized mixed Strategies. Consequently, only an approximate Nash Equilibrium can be achieved. In this paper, we induce the problem of obtaining an exact Nash Equilibrium into a generic function optimization in a space of continuous strategies. Meanwhile, the objective is defined as the summation of utility regret in strategic profiles, and its corresponding optimized solutions are achieved as Nash equilibrium. Thus, the gradient descent of objective function can be calculated through a set of linear program. Further, a gradient descent algorithm to find exact Nash Equilibrium of Graphical Game with any structure is presented. Our proposed method can be used to get the exact Nash equilibrium of graphical games with arbitrary structures. As well, our method has fairly good precision and efficiency even on the large-scaled graphical games.

**Keywords** Multi-agent interaction model, Graphical game, Nash equilibrium, Linear programming, Gradient descent algorithms

## 1 引言

在开放的、动态的多 Agent 系统中,Agent 间的交互(interactions)是最基本的方面之一。近年来,运用博弈论[4]的理论和方法来研究多 Agent 系统中 Agent 间的交互问题得到了广泛关注[10,11]。假设所有 Agent 都是理性的,即它们均追求各自利益的最大化,此时 Agent 间的协作、协调等交互问题可模型化为一个策略博弈(strategies games)[11]。具有各自利益的多个 Agent 预期其它 Agent 的行为,从而采取恰当的交互行动,使自己的利益最大化,最终所有 Agent 的决策将形成一个稳定的策略格局,没有任何 Agent 愿意偏离此格局,即博弈的 Nash 均衡(Nash Equilibrium)[4]。可见,Nash 均衡描述了多 Agent 系统交互问题的理想结局。然而,在基于博弈的 Nash 均衡解决多 Agent 交互的相关问题方面,仍有待深入地研究。经典博弈论只解释了 Nash 均衡的含义,却没有给出 Nash 均衡的计算方法。作为博弈论应用的关键,许多学者研究了 Nash 均衡的计算方法和计算复杂度,指出计算 Nash 均衡是困难的[7,5,6]。此外,博弈论应用的又一难题是表示 Agent 策略交互收益的效用矩阵,其大小与 Agent 的数目成指数关系增长[4]。因此,为了使博弈的 Nash 均衡能够有效地应用于多 Agent 交互模型中,研究博弈的简约表示方式以及高效的 Nash 均衡求解方法成为其重要的基础。

为了有效求解 Nash 均衡,及解决博弈表示复杂度高的问题,很多研究者基于实际多 Agent 系统中 Agents 策略间的效用独立性(utility independence),提出了一些更为简约的基于博弈论的 Agent 交互模型—结构化博弈(Structured Games),并研究了这些模型上 Nash 均衡求解方法[1~3,8,9]。其中,文[1]提出了一种无向图表示模型:图型博弈(Graphical Games)。模型中的图形拓扑表示很多现实环境中 Agent 交互的内在结构,例如可用一棵无向树表示具有层次关系的多 Agent 交互,树中的每一个节点对应一个 Agent,行为直接相互影响的 Agent 之间有边相连。文[1]和[8]给出了树结构图型博弈近似 Nash 均衡的求解算法。文[9]改进了以上求解算法,并给出了在任意图结构上 Nash 均衡的求解方法。然而,在 Nash 均衡的求解方面,上述工作仍存在如下不足之处:首先,这些算法将连续策略空间进行离散化,然后在离散策略空间中求解 Nash 均衡,虽然在一定程度上提高了求解的效率,但是只能保证得到近似 Nash 均衡,而近似 Nash 均

*)云南大学科研项目(No. 2005Q023C,2004Q024C),云南省自然科学基金项目(No. 2005F0009Q)。李 劲 助教,硕士,从事人工智能和多 Agent 系统的研究;岳 昆 助教,硕士,主要研究方向为智能数据分析;刘惟一 教授,博导,主要研究方向为数据与知识工程。

衡并不一定能作为进一步求解精确 Nash 均衡的基础。其次，这些求解算法与博弈的图结构相关，因此只能用于具有特定图结构图型博弈 Nash 均衡的求解。

基于以上相关研究结果，本文提出了一种求解连续策略(continuous strategies)图型博弈 Nash 均衡的方法。具体而言，我们把求解图型博弈 Nash 均衡看作是一个函数优化问题，目标函数最优解就是博弈的 Nash 均衡。因此，我们首先定义了一个连续策略空间上的优化目标，指出目标函数下降梯度的计算可归结为一组线性规划，进而提出了求解目标函数最优解的梯度下降算法，从而得到博弈的 Nash 均衡。我们对算法的正确性、收敛性以及效率进行了一系列实验测试，实验结果表明该算法可以求解连续策略空间中具有任意图结构图型博弈的 Nash 均衡，从而弥补了上述已有算法的不足。同时，对于大规模图型博弈，算法的可行性和高效性也得到了验证。

接下来，本文第 2 节介绍博弈论及图型博弈模型；第 3 节介绍了我们提出的连续策略图型博弈 Nash 均衡的求解算法；第 4 节给出了实验分析及结果；最后总结全文、给出结论，并展望将来的研究工作。

## 2 背景知识

本节中，作为背景知识，我们简单介绍博弈论及图型博弈的基本概念。

### 2.1 博弈论

策略型博弈[4]是表示形式上最简单的一种博弈。$n$ 人的策略型博弈指定了 $n$ 个局中人(Agent)，每个 $\text{Agent}_i$ 有一个纯策略(行动)集 $S_i$ 和效用函数 $u_i$。

设 $S_i=\langle s_{i1},s_{i2},\cdots,s_{ik}\rangle$ 是 $\text{Agent}_i$ 的纯策略空间。$S_i$ 给出了 $\text{Agent}_i$ 在进行博弈时可选择的行动集合。$\text{Agent}_i$ 的一个混合策略为一个映射 $\delta_i: S_i\to[0,1]$，即以 $\delta_i(s_{ij})$ 的概率取 $S_i$ 中的纯策略 $s_{ij}$，其中 $\sum_{j=1}^{k}\delta_i(s_{ij})=1$。本文中，设 $\delta_i$ 是 $\text{Agent}_i$ 可用的任意一个混合策略。所有 Agents 各自采取的混合策略 $\delta_1,\delta_2,\cdots,\delta_n$ 是统计独立的，称 $n$ 个 Agents 的这个策略组合为一个策略剖面，记作 $\sigma$。定义 $(\sigma_{-i},\delta'_i)$ 为如下的策略剖面：$(\sigma_{-i},\delta'_i)$ 中除 $\text{Agent}_i$ 的混合策略为 $\delta'_i$，剖面的其余部分均与 $\sigma$ 相同。

每一个 $\text{Agent}_i$ 对应一个纯策略效用矩阵 $M_i$，矩阵的行是 $\text{Agent}_i$ 的纯策略，矩阵的列是其余 Agents 纯策略的每一种组合。$M_i$ 的值实际上给出了 $\text{Agent}_i$ 的纯策略在每一种纯策略博弈的可能结局上获得的效用。显然，对于一个含 $n$ 个 Agents，每个 Agent 有 2 个纯策略的博弈，需要给出 $n$ 个矩阵 $M_i(1\leqslant i\leqslant n)$，每一个 $M_i$ 的大小为 $2^n$。

给定一个混合策略剖面 $\sigma$，$\text{Agent}_i$ 在 $\sigma$ 下获得的期望效用 $u_i(\sigma)$ 可由下面定义得到。

**定义 1**[4] 设 $s=(s_1,s_2,\cdots,s_n)$ 为纯策略剖面，其中的 $s_i$ 表示 $\text{Agent}_i$ 可用的任意一个纯策略。设 $\delta_i$ 是 $\text{Agent}_i$ 可用的任意一个混合策略。$S=S_1\times S_2\times\cdots\times S_n$ 为纯策略空间的笛卡尔积，则

$$u_i(\sigma)=\sum_{s\in S}[\delta_1(s_1)\delta_2(s_2)\cdots\cdots\delta_n(s_n)]u_i(s)$$

其中，$u_i(s)$ 的值可直接从 $M_i$ 得到。

下面我们简单介绍博弈的 Nash 均衡。博弈的 Nash 均衡是特殊的策略剖面，在此剖面下任何一个 Agent 在其他 Agents 不改变策略的情况下，单方面改变自己的策略不能使其获得更大的 $u_i(\sigma)$ 值，所以 Nash 均衡实际上定义了多个“自利”Agents 策略间的一种稳定状态。为了定义博弈的 Nash 均衡，我们先介绍 $\text{Agent}_i$ 在给定混合策略剖面 $\sigma$ 下的“偏离度”(utility regret)的定义。

**定义 2**[4] 给定混合策略剖面 $\sigma$，$\text{Agent}_i$ 关于 $\sigma$ 的“偏离度”$Reg_i(\sigma)$ 可由下面定义给出：

$$Reg_i(\sigma)=\max_{\delta'_i}(u_i(\sigma_{-1},\delta'_i)-u_i(\sigma))$$

其中当 $\text{Agent}_i$ 在剖面 $\sigma$ 下通过改变自己的策略不能获得更大的效用时，它将保持自己在剖面 $\sigma$ 中的策略不变，则有 $Reg_i(\sigma)\geqslant 0$。

下面我们给出基于“偏离度”的博弈 Nash 均衡定义。

**定义 3**[4] $\sigma$ 是一个博弈的混合策略 Nash 均衡，如果对于任意的 $\text{Agent}_i$ 都有：$Reg_i(\sigma)=0$；$\sigma$ 是一个混合策略 $\varepsilon$-Nash 均衡，如果对于任意的 $\text{Agent}_i$ 都有 $Reg_i(\sigma)\leqslant\varepsilon$。

下面的定理回答了“一个博弈是否一定存在 Nash 均衡?”问题。

**定理 1**[4] 每一个有限纯策略的策略型博弈至少存在一个混合策略 Nash 均衡解。

### 2.2 图型博弈

一般地，我们用效用矩阵来表示策略型博弈。在 2.1 节给出的背景知识中，每一个 $\text{Agent}_i$ 只有 2 个纯策略时，其效用矩阵 $M_i$ 的大小就为 $2^n$，可见博弈表示具有很高的复杂度。考虑在许多实际的博弈环境中，Agents 策略间存在效用独立关系，即 $\text{Agent}_i$ 策略的效用并非与其余 $n-1$ 个 Agents 都相关，往往只与有限几个 Agents 相关。基于这样的思想，文[1]提出了博弈的无向图表示模型。下面我们简单介绍图型博弈的基本框架。

$n$ 个 Agents 的图型博弈是一个二元组 $\langle Gr,M\rangle$，其中 $Gr$ 是 $n$ 个节点的无向图，每一个节点代表一个 Agent，连线表示节点策略间效用相关。$M$ 是效用矩阵 $M_i(1\leqslant i\leqslant n)$ 的集合，$M_i$ 对应于节点 $i$。在原有的静态博弈的表示中，我们总是假设任何一个节点其策略效用与其余 $n-1$ 个节点相关，在这样的情况下，$Gr$ 是 $n$ 个节点的完全图。在许多实际博弈环境中，节点之间的相关关系往往是稀疏的，即一个节点的策略效用往往只与其余 $n-1$ 个节点中的少数几个直接相关。设 $i$ 是 $Gr$ 中的任意节点，$N_{Gr}(i)$ 是 $Gr$ 中与 $i$ 邻接的节点集(包括节点 $i$)，一般地，总有 $|N_{Gr}(i)|=m(m<<n)$，此时 $M_i$ 的大小为 $2^m$。图 1 给出了一个含有 4 个 Agents 的图型博弈模型，每个 Agent 有两个纯策略，并分别给出了 $\text{Agent}_1$，$\text{Agent}_2$ 的效用矩阵 $M_1$，$M_2$。从 $M_1$ 可以看出 $\text{Agent}_1$ 的纯策略效用只与 $\text{Agent}_2$ 的纯策略相关，而与 $\text{Agent}_3$ 和 $\text{Agent}_4$ 的纯策略无关。

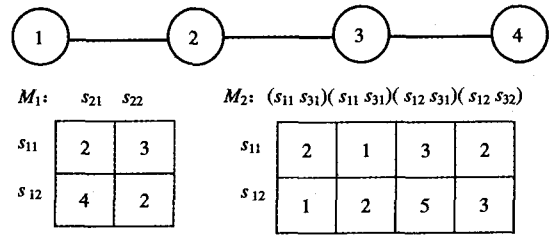



图 1 图型博弈模型示例

## 3 求解连续策略图型博弈 Nash 均衡的算法

基于上述对博弈论及图型博弈的介绍，为了有效地求解连续策略图型博弈的 Nash 均衡，本节中，我们首先定义一个

非负目标函数，然后提出一个求解该函数最优解的迭代优化算法，算法的最优解就是博弈的 Nash 均衡。

根据定义 3 可知，当博弈每个 Agents 的“偏离度”均为 0 时，所有 Agents 的混合策略构成的策略剖面就是 Nash 均衡。由此，我们给出定义 4，提出优化的目标函数。

**定义 4** 给定策略剖面 $\sigma$，一个 $\sigma$ 的非负的目标函数 $Sreg: S\to R^+$ 定义为

$$Sreg(\sigma)=\sum_{i=1}^{n}Reg_i(\sigma)$$

其中 $S$ 是 $n$ 个 Agents 的混合策略空间，$\sigma\in S$。由于“偏离度” $Reg_i(\sigma)$ 是一个关于 Agents 混合策略 $\sigma$ 的非负函数，目标函数 $Sreg(\sigma^*)=0$ 当且仅当 $\forall i, Reg_i(\sigma)=0$。根据定义 3，可知 $\sigma^*$ 是 Nash 均衡。

下面我们将给出定义 4 中目标函数的迭代优化算法。为便于理解算法，我们首先简单地介绍算法的主要步骤。

Step 1 算法从 $n$ 个 Agents 的混合策略剖面空间 $S$ 中随机地生成一个剖面 $\sigma\in S$ 作为初始解。

Step 2 对每个 Agent$_i$，固定其它 Agents 在 $\sigma$ 中的策略不变，只更新 Agent$_i$ 在 $\sigma$ 中的当前策略，求出最小目标函数值的剖面 $(\sigma_{-i},\delta_i^*)$。于是，对于全部 Agents 得到 $\sigma$ 邻域内的一组新解，记为 $NS=\{(\sigma_{-1},\delta_1^*),\cdots,(\sigma_{-n},\delta_n^*)\}$。

Step 3 从 $NS$ 中选取具有目标函数值最小（即具有最大下降梯度）的那个新解，记为 $\sigma^*$。算法在下列两种情况下停止迭代，输出解。第一，下降梯度小于预先指定的求解精度 $\varepsilon$ 或者达到最大迭代次数，此时输出近似解；第二，$Sreg(\sigma^*)=0$，此时输出精确均衡解。否则，以 $\sigma^*$ 作为初始解，算法转到 Step2 进入下一趟迭代。

可见，算法的关键是：在 Step 2 中，当固定其它 Agents 在 $\sigma$ 中的策略不变时，更新 Agent$_i$ 在 $\sigma$ 中的当前混合策略 $\delta_i$ 为 $\delta_i^*$，由此可获得该趟迭代中 Agent$_i$ 具有最小目标函数值的剖面 $(\sigma_{-i},\delta_i^*)$。那么，如何得到混合策略 $\delta_i^*$？定理 2 给出了求解 $\delta_i^*$ 的基本方法。

**定理 2** 给定 $n$ 个 Agents 的图型博弈 $\langle Gr,M\rangle$，其中 $Gr$ 是 $n$ 个节点的无向图，$M=\{M_1,M_2,\cdots,M_n\}$ 是博弈的效用矩阵集。设 $\forall i(i=1,2,\cdots,n)$，Agent$_i$ 的邻接节点为 $N_{Gr}(i)-i=\{i_1,i_2,\cdots,i_m\}$。不失一般性，我们假定博弈中每个 Agent 的纯策略数均为 $k$，则给定策略剖面 $\sigma$，求解 Agent$_i$ 的最小目标函数值剖面 $(\sigma_{-i},\delta_i^*)$ 可归结为求解下面的优化问题：

Minimize：

$$\min_{\delta'_i}\{\sum_{j=1}^{m}\{\max\{c_{i_j1}(\delta'_i),c_{i_j2}(\delta'_2),\cdots,c_{i_jk}(\delta'_i)\}-u_{i_j}(\sigma_{-i},\delta'_i)\}+(A-u_i(\sigma_{-i},\delta'_i))\}$$

*Subject to*：$x_{i1}+x_{i2}+\cdots+x_{ik}=1$；

$$0\leqslant x_{i1}<1,0\leqslant x_{i2}\leqslant 1,\cdots,0\leqslant x_{ik}\leqslant 1.$$

其中，$(\sigma_{-i},\delta'_i)=\langle\delta_1,\delta_2,\cdots,\delta'_i,\cdots,\delta_n\rangle$；$\delta'_i=\langle x_{i1},x_{i2},\cdots,x_{ik}\rangle$ 表示 Agent$_i$ 的混合策略，$c_{i_j}(\delta'_i)(j=1,2\cdots,k)$ 是 $x_{i1},x_{i2},x_{ik}$ 的线性表达式，表达式系数由给定策略剖面 $\sigma$ 及效用矩阵 $M_i$，$A$ 为正常数。

证明：固定其它 Agents 在 $\sigma$ 中的策略不变，更新 Agent$_i$ 在 $\sigma$ 中的当前策力为 $\delta'_i$，得到剖面 $(\sigma_{-i},\delta'_i)$。由于改变 Agent$_i$ 的策略只会影响与 Agent$_i$ 邻接的 $m$ 个 Agents 的偏离度，而其余 Agents 的偏离度保持不变。根据定义 4，我们有

$$\min_{\delta'_i}Sreg(\sigma_{-i},\delta'_i)=\min_{\delta'_i}\{\sum_{j=1}^{n}Reg_j(\sigma_{-i},\delta'_i)\}$$

$$=\min_{\delta'_i}\{Reg_i(\sigma_{-i},\delta'_i)+Reg_{i_1}(\sigma_{-i},\delta'_i)+\cdots+Reg_{i_m}(\sigma_{-i},\delta'_i)\}\quad①$$

对于①的偏离度，由定义 2，

$$Reg_j(\sigma_{-i},\delta'_i)=\max_{\delta'_j}\{u_j((\sigma_{-1},\delta'_i)_{-j},\delta'_i)-u_j((\sigma_{-i},\delta'_i))\}\quad②$$

其中 $(j=i,i_1,\cdots,i_m)$。

在②中，设 $\delta'_i=\langle x_{i1},x_{i2},\cdots,x_{ik}\rangle$，根据定义 1，我们可以得出 $u_j((\sigma_{-i},\delta'_i))=b_1x_{i1}+b_2x_{i2}+\cdots+b_kx_{ik}$，在给定剖面 $(\sigma_{-i},\delta'_i)$ 的情况下，$b_1,\cdots,b_k$ 均为常数。

对于②中的 $u_j((\sigma_{-i},\delta'_i)_{-j},\delta'_j)$，当 $j=i$ 时，由于，$x_{i1}+x_{i2}+\cdots+x_{ik}=1$ 且 $0\leqslant x_{i1}\leqslant 1,0\leqslant x_{i2}\leqslant 1,\cdots,0\leqslant x_{ik}\leqslant 1$，所以 $\max_{\delta'_i}(u_j(\sigma_{-i},\delta'_i))=\max\{b_1,b_2,b_k\}$。由于偏离度非负，我们有

$$Reg_i(\sigma_{-i},\delta'_i)=\max\{A-u_i(\sigma_{-i},\delta'_i),0\}\quad③$$

其中，$A=\max\{b_1,b_2,\cdots,b_k\}$。当 $(j=i_1,\cdots,i_m)$ 时，$u_{i_j}(\sigma_{-i},\delta'_i)=\delta_{i_j}(s_{i_j1})(a_{11}x_{i1}+a_{12}x_{i2}+\cdots+a_{1k}x_{ik})+\cdots+\delta_{i_j}(s_{i_jk})(a_{k1}x_{i1}+a_{k2}x_{i2}+\cdots+a_{kk}x_{ik})$

其中，给定了剖面 $(\sigma_{-i},\delta'_i)$ 时，系数 $a_{11}\sim a_{kk}$ 均为常数。若，设 $c_{i_j1}(\delta'_i)=(a_{l1}x_{i1}+a_{l2}x_{i1}+\cdots+a_{lk}x_{ik}(j=1,2,\cdots,k)$，则有

$$Reg_{i_j}(\sigma_{-i},\delta'_i)=\max\{\max\{c_{i_j1}(\delta'_i),c_{i_j2}(\delta'_i),\cdots,c_{i_jk}(\delta'_i)\}-u_{i_j}(\sigma_{-i},\delta'_i),0\}\quad④$$

综上①、③、④，有

$$\min_{\delta'_i}Sreg(\sigma_{-i},\delta'_i)=\min_{\delta'_i}\{\sum_{j=1}^{m}\{\max\{c_{i_j1}(\delta'_i),c_{i_j2}(\delta'_i),\cdots,c_{i_jk}(\delta'_i)\}-u_{i_j}(\sigma_{-i},\delta'_i)\}+(A-u_i(\sigma_{-i},\delta'_i))\}$$

定理 2 将求解 Agent$_i$ 的最小目标函数值剖面 $(\sigma_{-i},\delta_i^*)$ 归结为一个函数优化问题。下面我们将推导出定理中目标函数的优化可归结为 $k^m$ 组线性规划。具体推导过程如下。

定理 2 中的 $\max\{c_{i_j1}(\delta'_i),c_{i_j2}(\delta'_i),\cdots,c_{i_jk}(\delta'_i)\}-u_{i_j}(\sigma_{-i},\delta'_i)$ 可按下列 $k$ 种情形分别讨论：

$$\max\{c_{i_j1}(\delta'_i),c_{i_j2}(\delta'_i),\cdots,c_{i_jk}(\delta'_i)\}=c_{i_j1}(\delta'_i),$$

$$\Leftrightarrow c_{i_j1}(\delta'_i)\geqslant c_{i_j2}(\delta'_i),c_{i_j1}(\delta'_i)\geqslant c_{i_j3}(\delta'_i),\cdots,c_{i_j1}(\delta'_i)\geqslant c_{i_jk}(\delta'_i)(l=1,2,\cdots,k)$$

因为 $\forall l(l=1,2,\cdots,k)$，$c_{i_j1}(\delta'_i),=(a_{l1}x_{i1}+a_{l2}x_{i2}+\cdots+a_{lk}x_{ik})$，同时 $u_{i_j}(\sigma_{-i},\delta'_i)$ 也是 $\delta'_i=\langle x_{i1},x_{i2},\cdots,x_{ik}\rangle$ 的表达式，所以 $\forall l,c_{i_j1}(\delta'_i)-u_{i_j}(\sigma_{-i},\delta'_2)$，是 $\delta'_i=\langle x_{i1},x_{i2},\cdots,x_{ik}\rangle$ 的 $k$ 元线性表达式，而例如 $c_{i_jk}(\delta'_i)\geqslant c_{i_jk-1}(\delta'_i)$ 不等式也是关于 $\delta'_i=\langle x_{i1},x_{i2},\cdots,x_{ik}\rangle$ 的 $k$ 元线性约束。于是，本节中的②式等价于下面 $k$ 组线性规划问题

*Minimize*：$\{c_{i_j1}(\delta'_i)-u_{i_j}(\sigma_{-i},\delta'\}-u_i(\sigma_{-i},\delta'_i)$

*Subject to*：$c_{i_j1}(\delta'_i)\geqslant c_{i_j2}(\delta'_i)$

$$,\cdots$$

$$c_{i_j1}(\delta'_i)\geqslant c_{i_jk}(\delta'_i),x_{i1}+x_{i2}+\cdots+x_{ik}=1$$

$$0\leqslant x_{i1}\leqslant 1,0\leqslant x_{i2}1,\cdots,0\leqslant x_{ik}1$$

不难得出，每个 Agent$_i$ 有 $k$ 个纯策略、$m$ 个邻接节点时，定理 2 中目标函数可归结为 $k^m$ 组线性规划。

定理 2 中给出的方法，以及上述推导把求解 $\delta_i^*$ 归结为求解一组线性规划，解决了算法中的关键问题，算法的完整描述如下：

---

**算法**：求解连续策略图型博弈 Nash 均衡

**输入**：$\langle Gr,M\rangle$，其中 $Gr$ 是含 $n$ 个节点的无向图，$M$ 是节点的效用矩阵集。

**输出**：Nash 均衡。

Step 0 给定算法的最大迭代次数 $max\_itercout$；求解精度 $\varepsilon$。设 $Sreg(\sigma^*)=0$；

Step 1 设 $t=1$，在混合策略剖面空间中，随机生成初始策略剖面 $\sigma^{(t)}=\langle\delta_1,\cdots,\delta_i,\cdots,\delta_n\rangle$；

Step 2 求剖面 $\sigma^{(t)}$ 的目标函数值 $Sreg(\sigma^{(t)})$；

Step 3 IF $(Sreg(\sigma_{(t)})=0)$OR $(Sreg(\sigma^{(t)}-Sreg(\sigma^*)<\varepsilon)$ OR $(t=max_itercout)$THEN 输出 $\sigma^{(t)}$，算法终止；

Step 4 给定 $\sigma^{(t)}$，计算 $Sreg(\sigma^{(t)}_{-i},\delta_i^*),\cdots,Srge(\sigma^{(t)}_{-i},\delta_i^*),\cdots,Sreg(\sigma^{(t)}_{-n},\delta_n^*)$ ①；

$Sreg(\sigma^*)=\max\{Sreg(\sigma^{(t)}_{-1},\delta_1^*),\cdots,Sreg(\sigma^{(t)}_{-i},\delta_i^*),\cdots,Sreg(\sigma^{(t)}_{-n},\delta_n^*)\}$；

Step 5 将序列①与 $Sreg(\sigma^{(t)})$ 比较，将小于 $Sreg(\sigma^{(t)})$ 的目标函数排成有序序列：

$Sreg(\sigma^{(t)}_{-a_1},\delta_{a_1}^*)\leqslant\cdots\leqslant Sreg(a^{(t)}_{-a_i},\delta_{a_i}^*)\leqslant\cdots\leqslant Sreg(\sigma^{(t)}_{-a_n},\delta_{a_n}^*)\cdots\cdots$②；

Step 6 按照序列②，生成 $\sigma^{(t)}$ 的更新策略集：

$W_\sigma=\{a_{\sigma 1},a_{\sigma 2},\cdots\cdots,a_{\sigma k}\}$；

策略集满足任意 $i,j$，$Agent_{\sigma j}$ 和 $Agent_{\sigma j}$ 的邻接节点集相交为空。

Step 7 $t=t+1$，多策略更新，生成新的迭代剖面 $\sigma^{(t)}=\langle\delta_1,\cdots,\delta^*_{a_{\sigma 1}},\cdots,\delta^*_{a_{\sigma 2}},\cdots,\delta^*_{a_{\sigma k}},\cdots,\delta_n\rangle$；

Step 8 转(Step2)执行。

图 2 给出了博弈的效用矩阵，$k=3,m=1$。限于篇幅，我们只给出算法 1 的第一趟迭代的计算过程。

| | 石头 | 剪子 | 布 |
|---|---|---|---|
| 石头 | 0,0 | +1,−1 | −1,+1 |
| 剪子 | −1,+1 | 0,0 | +1,−1 |
| 布 | +1,−1 | −1,+1 | 0,0 |

图 2 石头、剪子、布博弈

给定初始策略剖面为 $\sigma^{(1)}=\langle\langle 1,0,0\rangle,\langle 1/2,0,1/2\rangle\rangle$，$Sreg(\sigma^{(1)})=Reg_1(\sigma^{(1)})+Reg_2(\sigma^{(1)})=0+3/2=3/2$。$gent_1$ 在 $\sigma^{(1)}$ 的最优混合策略 $\delta_1^*$ 的求解如下。

$$\min_{\delta_1}Sreg(\sigma^{(1)}_{-1},\delta_1^*)=$$

$$\min_{\delta_1}\{Reg_1(\sigma^{(1)}_{-1},\delta_1^*)+Reg_2(\sigma^{(1)}_{-1},\delta_1^*)=$$

$$\min_{\delta_1}\{[3/2x_{11}+1x_{12}+1/2x_{13})]$$

$$+\max\left\{\begin{matrix}(x_{11}+2x_{13})\\(2x_{11}+x_{12})\\(2x_{12}+x_{13})\end{matrix}\right\}-(1/2x_{11}+x_{12}+3/2x_{13})\}$$

上式归结为 3 个线性规划问题，求解后，最小值均为：3/2−1=0.5，对应的最优解也均为：$\langle 1/3,1/3,1/3\rangle$。基于上述计算结果，我们可得 $\min_{\delta_1}Sreg(\sigma^{(1)}_{-1},\delta_1^*)=\min_{\delta_1}\{0.5,0.5,0.5\}=0.5$，在 $Agent_1$ 取混合策略 $\delta_1^*=\langle 1/3,1/3,1/3\rangle$ 时得到。类似地，$\min_{\delta_2}Sreg(\sigma^{(1)}_{-2},\delta_2^*)=\min\{1,1,1\}=1$，在 $\delta_2^*=\langle 1/3,1/3\rangle$ 时得到。因为 $\min_{\delta_2}Sreg(\sigma^{(1)}_{-1},\delta_1^*)>\min_{\delta_2}Sreg(\sigma^{(1)}_{-2},\delta'_2)$，所以 $\sigma^{(2)}=\ll 1/3,1/3,1/3\rangle,\langle 1/2,0,1/2\gg$。以新的策略剖面 $\sigma^{(2)}$ 作为初始解，迭代执行上述过程，最终算法收敛到 $Sreg(\sigma^{(3)}=0$，最优解为 $\ll 1/3,1/3,1/3\rangle,\langle 1/3,1/3,1/3\gg$。

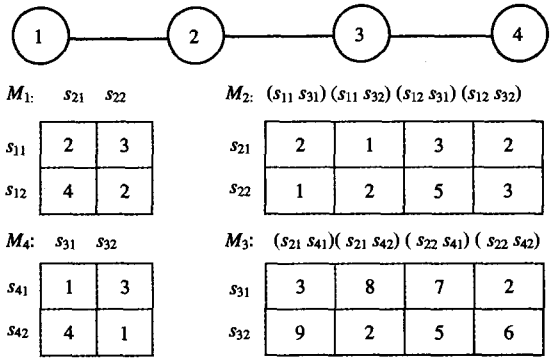
## 4 实验结果

为了验证算法的收敛性及运行效率，我们用 Matlab6.5 实现了算法，并进行了实验结果分析。实验的运行环境为：CPU P1.5G 处理器，512M 内存，WINDOWS XP 操作系统。

### 4.1 算法的收敛性

首先，我们对算法的收敛性进行了验证。所谓的收敛性是指：如果算法运行的最终结果使得目标函数值 $Sreg(\sigma)=0$，我们说算法完全收敛到 Nash 均衡；否则我们说算法近似收敛。如果算法是近似收敛的，则令 $\varepsilon$ 等于结果剖面下的最大偏离度值 $Reg^*(\sigma)$，此时收敛的结果是一个近似度为 $\varepsilon$ 的近似 Nash 均衡。

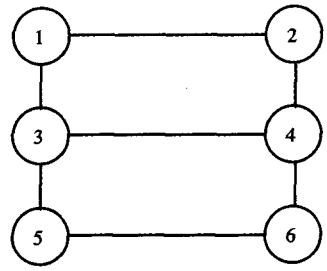
我们在不同图结构的 Agent 交互模型上对算法的收敛性进行了测试。首先，我们针对一系列预先指定图结构，在随机生成效用矩阵的交互模型上进行了实验，如下面的实例 1(图 3)。




Nash 均衡：<1.0000 0.0000>,<0.0000 1.0000>,
<0.4000 0.6000>,<0.6667 0.3333>；$Sreg(\sigma^*)=0$

图 3 实例 1：四个 Agents 的策略交互模型及其 Nash 均衡

为验证在结构复杂、Agents 数目较多的交互模型中算法的收敛性，我们在文[2]中给出的 Road 博弈基准测试上进行了一系列测试。实验采用图 3 所示矩阵作为 Agents 的效用矩阵，算法在不同规模的 Road 博弈上收敛到 $\varepsilon\leqslant 0.2$ 近似均衡。实例 2 是 6 个 Agents 的 Road 博弈及其近似 Nash 均衡(图 4)。




$\sigma^*=\{$〈0.3581 0.3333 0.30867〉,〈0.2353 0.4172 0.34767〉,
〈0.2823 0.3844 0.33337〉,〈0.3344 0.2487 0.4169〉,
〈0.3597 0.2823 0.3581〉,〈0.4329 0.3342 0.2364〉$\}$

其中 $Sreg(\sigma^*)=0.0252;\varepsilon=0.0040$。

图 4 实例 2：6 个 Agents 的 Road 交互模型及其近似 Nash 均衡

表 1 算法在不同规模的 Road 博弈收敛的情况

| Road 博弈规模 | $Sreg(\sigma^*)$ | $\varepsilon$ |
|---|---|---|
| 6 | 0.0252 | 0.0040 |
| 10 | 0.0538 | 0.0092 |
| 20 | 0.1283 | 0.0214 |
| 40 | 0.3460 | 0.0663 |
| 80 | 0.6390 | 0.1450 |
| 100 | 1.2030 | 0.2010 |

表 1 给出了在不同规模 Road 博弈上算法收敛的结果。

从实验结果可以看出，随着博弈规模的增大，算法收敛到合理 ε 值的近似解。

### 4.2 算法的运行效率

为了测试算法在较大规模 Agent 交互模型上的运行效率，我们在一组不同规模的 Road 博弈上进行了实验。实验中，Road 博弈采用图 3 所示的效用矩阵，ε 阈值为 0.3（即算法收敛到 ε≤0.3 时停止迭代），针对 Agent 数目分别为 10，20，30，40，…，100 的 Road 博弈，我们测试了本文提出的 Nash 均衡求解算法，其执行时间如图 5 所示。

从实验结果可以看出，算法的效率并不随 Agents 的增多而显著下降，并且对于大多数情况有较合理的收敛结果。因此，我们提出的算法对于求解连续图型博弈的 Nash 均衡具有一定的可行性和高效性。

**总结与展望** 博弈论在多 Agent 系统交互问题的研究中具有重要的理论价值和应用背景。图型博弈是近年来提出的一种重要的基于博弈论的多 Agent 交互模型。Nash 均衡是多个理性 Agent 交互的预期结局，求解 Nash 均衡是图型博弈的关键问题。与已有的基于 Agent 策略离散化的求解方法不同，本文把求解图型博弈的 Nash 均衡看作是一个函数优化问题，定义了目标函数，并把目标函数最优解的求解归结为一组线性规划，进而提出了一个求解连续策略空间中图型博弈 Nash 均衡的新型算法。并且，为了验证我们提出算法的可行性、收敛性，以及分析算法的运行效率，我们对在一些经典的博弈模型及复杂图型博弈上对算法进行了实验分析。实验结果表明我们提出的算法具有一定的可行性和高效性。本文研究的内容和提出的方法也引发了另外一些有价值的研究课题，例如在处理较大规模多 Agent 交互模型时，进一步提高算法的求解精度、求解非完全信息情况下多 Agent 图型博弈的 Nash 均衡，这些也是我们正在进行的研究工作。

## 参 考 文 献


1 Kearns M, Littman M L, Singh S. Graphical models for game theory [A]. In: Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence [C]. Seattle, USA, 2001. 253～260

2 Koller D, Milch B. Multi-agent influence diagram for representing and solving games [A]. In: Proceedings of the 17th International Joint Conferences on Artificial Intelligence [C]. Seattle, USA, 2001. 1027～1034

3 La Mura P. Game networks [A]. In: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence [C]. Stanford, CA, USA, 2000. 335～342

4 Fudenberg D, Tirole J. Game Theory [M]. Cambridge, MA, MIT Press, 1991. 4～42

5 Papadimitriou C H. Algorithms, Games, and the Internet [A]. In: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing [C]. Crete, Greece, 2001. 749～753

6 Conitzer V, Sandholm T. Complexity Results about Nash Equilibrium [A]. In: Proceedings of the 18th International Joint Conferences on Artificial Intelligence [C]. Acapulco, Mexico, 2003. 765～771

7 McKelvey R, McLennan A. Computation of equilibria in finite games [A]. In: Handbook of computational Economics, Elsevier Science, 1996, 1:87～142

8 Littman M L, Kearns M, Singh S. An efficient exact algorithm for singly connected graphical games [A]. In: Proceedings of the 14th Neural Information Processing Systems [C]. Vancouver, British Columbia, Canada, 2001. 817～823

9 Ortiz L E, Kearns M. Nash propagation for loopy graphical games [A]. In: Proceedings of the 15th Neural Information Processing Systems [C]. Vancouver, British Columbia, Canada, 2002. 793～800

10 Parsons S, Gmytrasiewicz P, Wooldridge M. Game Theory and Decision Theory in Agent-based Systems [M]. Kluwer Academic Publishers, 2002,5

11 Wooldridge M. An Introduction to Multiagent Systems [M]. John Wiley & Sons, 2002


（上接第 167 页）

法是可以互补的方法，它们分别适用于不同类型的数据集。

Liu 等[8]将类标志作为一个属性参与关联规则的挖掘过程，然后逐条利用得到的关联规则，特别是带有类标志的关联规则来训练分类器，将那些无助于分类，甚至对分类产生副作用的关联规则去除，最后用剩余的关联规则来构建分类器。但 Liu 对于分类器的训练方法也存在着对于训练集过度适应的问题。

**总结** 本文研究了 JEP——一种在不同数据集之间支持度从零到非零跳跃性变化的项集在数据分类中存在的问题，提出了项集独立支持度的概念。相对于传统的项集支持度来说，独立支持度能够更全面地描述数据的分布特征，为更加准确的分类提供依据。进而，在独立支持度的基础上提出了 JEP 平均长度的概念，并提出了一种以测试样本所覆盖 JEP 的平均长度作为分类特征的分类方法，该方法可以更加有效地区分类边界上的数据，能够为数据提供更为准确的分类。我们还将进一步研究 JEP 分类与概念树的结合以及 JEP 在序列分类中的应用等问题。

## 参 考 文 献


1 Han Jiawei, Kamber M. Data Mining: Concepts and Techniques. Morgan Kaufman Publishers, Inc. 2001

2 Agrawal R, Imielinski T, Srikant R. Mining association rules between sets of items in large databases. In: Proceedings of 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 1993. 207～216

3 Han J, Pei J, Yin Y. Mining Frequent Patterns without Candidate Generation. In: Proceedings of the 2000 ACM SIGMOD international conference on Management of data, Dallas, Texas, United States, May 2000. 1～12

4 Zaki M J, Hsiao C. CHARM: An Efficient Algorithm for Closed Itemset Mining: [Technical Report 99-10]. Computer Science Dept., Rensselaer Polytechnic Inst., Oct. 1999

5 Quinlan J. R C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann, 1993

6 Li J. Dong G, Ramamohanarao K. Making Use of the Most Expressive Jumping Emerging Patterns for Classification. In: Proceedings of the Fourth Pacific-Asia Conference on Knowledge Discovery and Data Mining, Kyoto, Japan, 2000. 220～232

7 Dong G, Li J. Efficient Mining of Emerging Patterns: Discovering Trends and Differences. In: Proceedings of the fifth ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, San Diego, CA, ACM Press, New York, August 1999. 43～52

8 Liu B, Hsu W, Ma Y. Integrating Classification and Association Rule Mining. In: Proceedings of the fourth Int'l Conf. on Knowledge Discovery and Data Mining, New York, AAAI Press, August 1998. 80～86

9 Lent B, Swami A, Widom J. Clustering Association Rules. In: Proc. 1997 Int. Conf. Data Engineering (ICDE'97), Birmingham, England, April 1997. 220～231

10 Wang Y, Wang K C. From Association to Classification: Inference Using Weight of Evidence. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(3): 764～767