非对称博弈的表示和求解①

石 黎[1]，林 仙[2]

[1](湖北经济学院 信息管理学院，武汉 430205)
[2](云南省国家税务局 信息中心，昆明 650051)

摘 要：非对称博弈是一种普遍存在的博弈现象，现实中大量的博弈都呈现出非对称的特性。但是非对称博弈的表示问题在多-Agent 影响图中是一个难以解决的问题，存在表示复杂和求解效率低的情况。针对该问题，借鉴了单-Agent 决策系统中非对称性表示的方法，提出了一种新的博弈模型，有效的解决了非对称博弈的表示问题。给出了该模型详细的求解算法，并使用一个实例来加以说明。

关键词：博弈模型；多-Agent 影响图；非对称多-Agent 影响图；策略；效用

Representation and Solution of Asymmetric Games

SHI Li[1], LIN Xian[2]

[1](School of Information Management, Hubei University of Economics, Wuhan 430205, China)
[2](Information Centrel, Internal Revenue Service, Kunming 650051, China)

Abstract: Asymmetric Games are common in reality. Many games have emerged non-symmetrical features. But in Multi-agent Influence Diagrams, representing asymmetric games may cause representation explosion and inefficient solving. In order to solve this problem, we propose a new game model to represent and solve asymmetric games compactly and efficiently. A detailed algorithm is given, and an example is used to illustrate.

Key words: game model; multi-agent influence diagrams; asymmetric multi-agent influence diagrams; strategies; utility

博弈论[1]是由 Von Neumann 和 Morgenstern 于 1947 年创立的。近年来，基于现实博弈环境中广泛存在的 Agent 策略间的效用独立关系，许多学者提出了一些更为简洁有效的博弈表示模型。P. La Mura 于 2000 年提出了博弈网，是一种更加结构化和更加压缩的图形表示模型。M.Kearns 等于 2001 年首先提出了用图形博弈模型来描述含多个 Agents 的静态完全信息博弈。Koller 和 Milch 给出的多-Agent 影响图是对贝叶斯网和影响图的扩展，它能够表示最复杂的动态不完全信息博弈。但多-Agent 影响图存在这样一个问题：当博弈是非对称结构的时候，博弈树能以很自然的方式将非对称性简洁的表示出来，而多-Agent 影响图表示非对称性却非常的困难，一个简单的非对称博弈都有可能导致表示的爆炸。针对上述问题，我们借鉴了单-Agent 决策系统中对非对称性表示的方法，将非对称影响图表示非对称决策问题的方法引入到多-Agent 影响图中，两者融合得到了能够有效表示非对称博弈的一种新的图形博弈模型—非对称多-Agent 影响图，并给出具体的算法求解。

1 相关知识介绍

博弈论[1]是关于包含相互依存情况中理性行为的研究。所谓相互依存，通常是指博弈中的任何一个局中人都受到其他局中人的行为的影响。在经典博弈论中，博弈的表示形式主要有两种：效用矩阵和博弈树。从博弈树的结构特点来对博弈分类，可将博弈分为对称博弈和非对称博弈两类。为了说明问题，我们先来看两个简单的博弈例子[1]。

例 1：智猪博弈[1]说的是猪圈里有两头猪，一大一小。猪圈的一边有个踏板，每踩一下踏板，猪圈的另

① 基金项目:湖北省教育科学“十一五”规划项目(2008B117)
收稿时间:2011-02-13;收到修改稿时间:2011-04-24

一头固有的食物槽里将会流出饲料。设食物槽内一次流出的饲料共有 6 份，若小猪踩动踏板，大猪等待在食物槽前，大猪会在小猪跑到食槽之前刚好吃光所有的 6 份食物；若大猪踩动踏板，小猪等在食物槽前，则大猪还有机会在小猪吃完落下的食物之前跑到食槽，争吃到 1 份食物；倘若它们共同踩踏板再一起跑向食物槽，则大猪能争到 4 份饲料而小猪只能争到 2 份饲料。这个博弈可以用效用矩阵表示为：

$$\begin{bmatrix} 2/4 & 0/6 \\ 5/1 & 0/0 \end{bmatrix}$$

其中 2/4 的含义为当小猪和大猪同时踩时各自获得的利益；0/6 为小猪踩大猪不踩时各自获得的利益，其他类推。我们以此例来阐述博弈的一些基本概念。任何一个博弈都包含以下三个基本的要素：

（1）局中人：以 i = 1, 2,…表示，在一场竞赛或博弈中，每一个有决策权的参与者成为一个局中人。在上例中，局中人就是大猪和小猪，有时也把局中人称作 Agent，两者是等同的概念而不加区分。

（2）策略空间：每个局中人一般有若干个策略（strategies）可供选择，它们构成了该局中人的纯策略空间。局中人 i 的纯策略空间用 Si 表示。在上例中，大猪和小猪都有两个纯策略可供选择：(踩，不踩)。

（3）每个局中人的效用函数：在博弈论中，局中人的利益用效用表示，我们记局中人 i 的效用函数为 $u_i(s)$，其中 s = (s1,…,sr)，s_j 表示局中人 j 所取策略，s 表示 r 个局中人的策略向量。效用函数 $u_i(s)$ 与 s 有密切关系，它是局中人真正关心的东西。例如，在智猪博弈中，$u_{小猪}$ (小猪踩，大猪踩)= 2，$u_{小猪}$ (小猪踩，大猪不踩)= 0。

例 2：“进入-阻抗”博弈[1]设局中人 1 为某产品市场的“在位者”，局中人 2 在考虑是否进入该产品市场。若局中人 2 做出“进入”市场的决策，那么双方形成竞争。假设在产量方面他们都有两种选择，那么该博弈的博弈树如图 1 所示：在图 1 中，一共有两个局中人：1（“在位者”）和 2（“进入者”）。首先是局中人 2 在“进入”与“不进”两个纯策略之间进行选择。如果局中人 2 选择了“不进”，那么整个博弈结束，得到两个局中人的结局 1；如果局中人 2 选择了“进入”，那么博弈进入博弈树右边的一支，开始博弈第二个阶段的决策，局中人 1 和局中人 2 将同时选择一个该产品的产量，最终到达四种结局中的任何一种。局中人 2 的两个决策点用虚线相连，表示 2 在决策的时候并不知道自己确切的在博弈树中的哪一个节点位置，也就是说，局中人 2 在选择他自己的产量的时候并不知道局中人 1 所选择的产量是多少。该博弈是一个非对称博弈，而图 2 所示的博弈是一个对称博弈。所谓对称博弈，是指当用博弈树来表示该博弈时，博弈树中的终结点数目与博弈中所有决策点的纯策略数的乘积相等。相应的，如果博弈树中的终结点数目小于博弈中所有决策点的纯策略数的乘积，则该博弈树所表示的博弈就是一个非对称博弈。

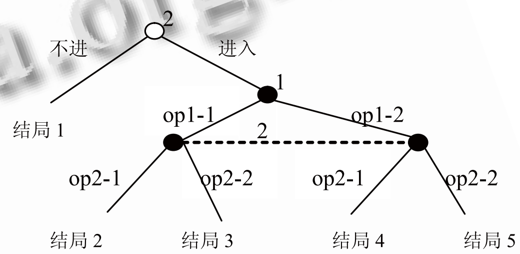

图 1　非对称博弈

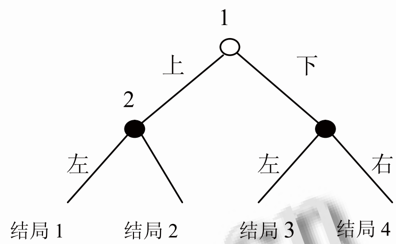

图 2　非对称博弈

2　多-Agent影响图和非对称影响图

在一个多 agents 相互作用的局部可观察环境中，博弈论为 agent 理性行为的选择提供了一个数学的框架。但是传统的博弈表示方法最初都是被设计用来帮助抽象数学公式和进行数学分析的，因此标准的博弈表示方法，包括常规的形式（效用矩阵）和扩展的形式（博弈树），他们都模糊了某些在现实世界的场景中很重要的结构——把决策情况分解成随机变量和决策变量以及这些变量之间的相互依赖关系。多-Agent 影响图能够反映这类结构并在分析和求解博弈 Nash 均衡中有诸多优势。多 agents 影响图（MAIDs）使我们能够以自然的表示形式来描述复杂的博弈并将变量级的相互作用结构清晰的表示出来。一般来说，MAID 比起博弈树来说是一种更加压缩的表示形式，但用 MAIDs 来表示一个简单的非对称博弈就有可能导致

表示的爆炸。将一个非对称博弈表示成一个多-Agent 影响图需要将问题“对称化”，同时还需要假定一些“退化的”概率分布和效用函数来定义相应的“人工”状态的概率或效用。这样一个“对称化”的过程存在很多的弊端。首先会导致博弈实际结构的模糊化。其次会使表示的空间膨胀，相应的也使均衡求解的效率极其低下。因此，需要对原有的 MAIDs 进行扩展使之能够同样以一种压缩的方式来表示非对称博弈，即兼有博弈树和 MAIDs 两者的优点。

Nielsen 和 Jensen 提出的非对称影响图 AIDs 建立在影响图 IDs 的基础之上，他将决策问题的非对称性定性的在图形结构中表示出来，因此我们可以从 AIDs 中直接获得决策问题的非对称信息。AIDs 的求解采用了“分而治之”的方法，将一个初始的非对称决策问题分解成多个对称的子问题，也就是将一个非对称影响图分解成多个影响图，然后再利用已有的求解影响图的方法来分别求解各个分解得到的影响图，最后通过合并每个影响图的结果最终得到决策问题的最优解。

3 非对称多-Agent影响图

3.1 非对称多-Agent 影响图的提出

非对称多-Agent 影响图是在多-agent 影响图[2]和非对称影响图[3]的基础上发展而来的。非对称影响图用一个约束弧集和一个标记集来表示决策问题的非对称性。我们将这种机制引入到多-Agent 影响图中来表示博弈的非对称性，得到一种新的结构化博弈模型：非对称多-Agent 影响图。与多-agent 影响图 MAIDs 对比，AMAIDs 除了具有 MAIDs 的结构特点外，在模型中加入了约束弧（边）和标记机制来表示博弈的非对称性；与非对称影响图 AIDs 相比，AMAIDs 除了具有 AIDs 中的用约束弧和标记来表示决策问题的非对称性的机制，还将 AIDs 中的单-agent 决策问题扩展成允许有多个 agents 的博弈情况。可见，AMAIDs 是一种既能够表示多-agent 决策问题又能表示非对称决策问题的图形表示模型，它融合了 MAIDs 和 AIDs 并同时继承了它们各自的优点。

3.2 非对称多-Agent 影响图的求解

首先采用 AIDs 中分解非对称影响图的方法将待求解的非对称多-agent 影响图分解成多个多-agent 影响图；然后利用求解多-agent 影响图的方法为每一个分解得到的多-agent 影响图求解出一个均衡；最后合并求解结果，找到初始问题的均衡解。两个算法如下所述。

非对称多-agent 影响图分解成多-agent 影响图集合的算法：Decomposition (I, CI)

输入：I, I 所处的上下文 CI;

输出：I 的一棵分解树 T.

方法：

(1) 把 I 作为树 T 的树根；

(2) 找到节点 I 的初始分裂变量 S1I, 设在 I 的所有标记中变量 S1I 有 n 个不同的取值，S1I 的状态空间 WS1I 所含的状态数| WS1I|=m, 则：

① 若n<m, 将 I 分解成 n+1 个 AMAIDs, k:=n+1；

设 n+1 个分解得到的 AMAIDs 为：M1, ,…Mi…, Mn+1，与 I 相比，在 Mi 中：如果 S1I 在 I 中有前驱的决策节点，那么分两种情况处理：

a.若 Vi 是一个单值，那么在 Mi 中就将变成一个随机节点，并且它有唯一的取值，即 Vi;

b.若 Vi 是一个多值的集合，那么在 Mi 中它仍然是一个决策节点，只不过这时它的可选行动集合变成了 Vi;

如果 S1I 在 I 中没有前驱节点，那么：我们移除节点 S1I 以及由此产生的孤点。

当 S1I=Vi 时，将 S1I=Vi 带入含有变量 S1I 的标记中进行计算，所有取值为 true 的标记被从 Mi 中移除；所有取值为 false 的标记所关联的信息边、节点（移除决策结点时，连同其后继结点一起移除）被从 Mi 中移除，随之产生的孤点也将不在 Mi 中出现；经过计算之后仍不能确定真假值的标记在移除含有变量 S1I 的项后保留，各个 Mi 中剩余的标记将不再含有变量 S1I.

② 若 n=m, 将 I 分解成 m 个 AMAIDs， k:=m;

设 m 个分解得到的 AMAIDs 为：M1, ,…Mi…, Mm, Mi 是使 S1I 取值 Vi 分解得到的一个子图。Mi 的结构如上所述，不再复述。

(3) 如果图 I 中有约束弧（约束弧是用约束函数来定量定义的），那么根据约束函数把弧头节点当作一个分裂变量来处理，设有约束弧 r(A,B),对应它的约束函数的值域用 range(r)来表示，即对于节点 A 的不同取值，节点 B 一共有 range (r)个不同的行动集。当 A 作为初始分裂变量对图形 I’进行分解的时候，我们把图 I’分解成|range(r)|个子图，在每个子图 Ii 中，变量 B 的行动空间 WB=r(A=ai)，其中 ai 是使 B 具有不同行

动集的 A 的第 i 个取值，它可以是单个的值，也可以是多个取值的集合，那么 WB 改变后，我们对图中的标记用新的 WB 进行计算，可能使子图中的某些标记成为永假，相应的在子图 Ii 中我们删除与这些永假标记相关联的节点、信息边（及被删除决策结点的后继节点）；WB 改变后，也可能使另一些标记成为永真，相应的子图 Ii 中我们把这些永真的标记删除，表示先前与之相关联的节点在该上下文中将无条件出现；经过计算之后仍不能确定真假值的标记在移除含有变量 A 的项后保留。对节点 A 的处理：

a. 若 A 在原图 I’中无前驱，则从子图 Ii 中删除 A 及由此产生的孤点；

b. 如果 A 在 I 中有前驱的决策节点，那么分两种情况处理：

若 ai 是一个单值，那么在 Ii 中 A 就将变成一个随机节点，并且它有唯一的取值，即 ai;

若 ai 是一个多值的集合，那么在 Ii 中 A 仍然是一个决策节点，只不过这时它的可选行动集合变成了 ai;

(4) 对新分解生成的 k 个 AMAIDs(M1,…Mi,…Mk)：i=1,…,k

① 如果 Mi 不是 MAID, 则执行以下的操作: Ti:= Decomposition(Mi, CMi);把 Ti 加入到 T 中作为树根的第 i 棵子树;

② 如果 Mi 是一个 MAID 的话，就直接将 Mi 作为树根 I 的第 i 个儿子节点加入到树 T 中.

(5) return(T)；

求解非对称多-agent 影响图的算法：Compute(I)
//* 该算法对一个非对称多-agent 影响图 I 进行求解*//

输入: I

输出: I 的一个纳什均衡解

方法:

（1）C:= Φ;

（2）T:= Decomposition (I,C);

（3）T 中的每一个叶子节点 Li 都是一个 MAID，调用文献[2]中给出的 Compute-equilibrium(Li)算法求解出 Li 的一个均衡策略及所对应的效用向量。当所有这样的子树处理完后，得到一棵新的以 MAID 为叶子节点，以 AMAID 为内部节点的树 T’；

（4）对 T’继续执行第 3 步，直到 T’成为只有单个节点的树；

（5）对 T’中唯一的节点（它是一个 MAID）求解，最终得到初始博弈的纳什均衡解和所有局中人的效用向量。

4 应用举例

以文献[4]所举的“公司和工会的工资讨价还价”博弈为例。该博弈树所表示的博弈是一个非对称博弈。使用算法 Decomposition，我们用一棵树来表示分解的结果，如下图 3 所示，在求解之前，我们先假定: Wl<Pl<Wh<Ph。

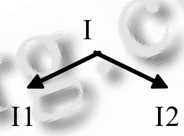

图 3 简单讨价还价博弈求解的分解树

首先，我们来求解 I1: CI1: D=A。其中随机节点 B1 是一个工会和公司都共知的公司类型的一个先验概率分布，我们把它假设为如下的 CPD(效用函数表)：

表 1 B1 的效用函数表

B1	Ph	Pl
P(B1)	0.5	0.5

在 CI1 下，工会在决策点“工会 1”的可选行动集为{Wh,Wl}，下面分别计算“工会 1”选择 Wh 和 Wl 时的最大期望效用值：

“工会 1”=Wh 时：工会在效用节点 U1 的期望效用 EU1(Wh)=0.5*Wh+0.5*Wh=Wh;

“工会 1”=Wl 时：工会在效用节点 U1 的期望效用 EUI(Wl)=0.5*Wl+0.5*Wl=Wl;

由于 Wh>Wl，所以求解 I1 得到的解是：“工会 1”=Wh，对应的效用向量为（Wh, F-Wh），这里的 F 表示公司的盈利（或类型），F= Ph 或 Pl 。

接下来求解 I2: CI2: D=R。这是一个含有多个 agents 决策点的 MAID，于是我们利用 MAIDS 的均衡求解算法来计算“公司 2”的策略。得到“公司 2”的策略之后，便可求解“工会 2”的策略：求解“工会 2”的策略时需要考虑随机变量 B2 的取值，我们将 B2 的 CPD 定义为如下的形式：

表 2 B2 的效用函数表

（工会 1,D2） B2	(Wh,R)	(Wl,R)
Ph	0.3	0.1
Pl	0.7	0.9

当“工会 1”= Wh 时：

“工会 2”取 Wh 的期望效用：

EU2(Wh)= (0.3*Wh+0.7*0)* δ = 0.3 δ Wh;

“工会 2“取 Wl 的期望效用：

EU2(Wl)= (0.3*Wl+0.7*Wl)*δ= δWl;

当“工会 1”= Wl 时：

“工会 2”取 Wh 的期望效用：

EU2(Wh)= (0.1*Wh+0.9*0)*δ= 0.1δWh;

“工会 2“取 Wl 的期望效用：

EU2(Wl)= (0.1*Wl+0.9*Wl)*δ= δWl;

假设 Wh=5Wl， 则可以得到“工会 2”的策略：

“工会 1= Wh，D= R”：　“工会 2”= Wh;

“工会 1= Wl，D= R”：　“工会 2”= Wl.

最后求解“工会 1”的策略：

“工会 1”= Wh 时，“工会 2”= Wh, 工会最后得到的效用 Uh= 0.3 δ Wh;

“工会 1”= Wl 时，“工会 2”= Wl, 工会最后得到的效用 Ul= δ Wl;

因为 Uh= 0.3 δ Wh = Uh= 1.5 δ Wl > Ul= δ Wl，所以“工会 1”的策略是：“工会 1”= Wh.

至此，我们得到了 I2 的策略剖面：(工会 1，工会 2，公司 2) = (Wh，Wh，(A，R))。其中（A,R）对应公司在两种类型（Ph,Pl）下分别采取的策略。

I1 和 I2 求解出来之后，我们可以对 I 进行求解了：首先将 I 转换成 I’，I’ 及对应的策略相关图如下图 4 所示：

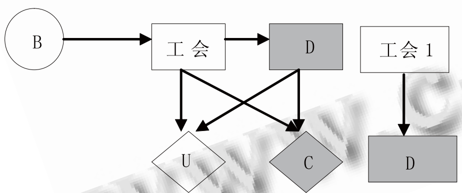

图 4　I 转换后得到的 I’以及 I’的策略相关图

其中效用节点 U 和 C 的 CPD 如下表所示：

表 3　工会的效用函数表

工会 1 / D	Wh	Wl
A	V(U)= Wh	Wl
R	0.3δWh	δWl

表 4　公司的效用函数表

工会 1 / D	Wh	Wl
A	V(C)= F-Wh	F-Wl
R	(δ(Ph-Wh), 0)	δ(F-Wl)

使用 Compute 算法，我们得到了初始博弈的均衡解：(工会 1， D，工会 2， 公司 2) =（Wh, (A,R), Wh, (A,R)）

3　结语

博弈问题属于多-agent 决策问题的范畴。在现实中，非对称博弈是普遍存在的，在今后的研究中，我们将着力于寻找更有效的求解非对称多-Agent 影响图的方法以及博弈的多-均衡解处理问题等相关问题。

参考文献

1 施锡铨.博弈论.上海:上海财经大学出版社,2000.

2 Koller D, Milch B. Multi-agent influence diagrams for representing and solving games. IJCAI, 2001. 1027−1034.

3 Nielsen TD, Jensen FV. Representing and solving asymmetric decision problems. Uncertainty in Artificial Intelligence: Proc. of Sixteenth Conference, Morgan Kaufmann, San Francisco, CA, 2000.416−425.

4 石黎,林仙.AMAIDs 的研究与应用.计算机系统应用,2006, 15(9):57−59.