一种新的结构化博弈模型的表示和求解

The Representation and Solution of a New Structured Game Model

石 黎　（湖北经济学院计算机科学与技术学院 武汉 430205）

摘 要：针对多-Agent 影响图表示方法存在效率低、结构模糊、表示复杂等方面的问题，提出了一种新的结构化博弈模型—非对称多-Agent 影响图。该模型借鉴了非对称影响图中的表示机制和方法，继承了多-agent 影响图在表示博弈时所具有的优点同时又具备了有效的表示非对称博弈的特点。给出了求解的算法，并使用一个实例来说明该模型的表示和求解。

关键词：多-Agent 影响图 非对称影响图 非对称多-Agent 影响图

1 问题的提出

近年来，博弈论对人工智能领域的研究产生了很大的影响。博弈论为多-Agent 系统的构建提供了数学的框架，为了得到决策问题的博弈解，各种各样的问题建模语言和求解算法相继出现。

Koller 和 Milch[1] 给出的多 agents 影响图是对贝叶斯网和影响图[4] 的扩展，它能够表示涉及多 agents 的决策问题的博弈解。实际上从结构上来说，一个多 agents 影响图可以简单的看作是一个影响图，但此时影响图中的决策节点和效用节点不再是属于一个 agent 的而是属于多个 agent 的，所以多 agents 影响图中的每个决策节点和效用节点都是和某一个 agent 相关联的。多 agents 影响图定义了非合作博弈的语义：一个多 agents 影响图可以转换成一棵等价的博弈树，而且多 agents 影响图同博弈树相比，它是一种更加压缩的表示形式。但多-Agent 影响图存在这样一个问题：当博弈是对称结构的时候，用多-Agent 影响图来表示博弈比用博弈树表示博弈在很大程度上节省空间，但当博弈是非对称结构的时候，情况就刚好相反了。此时，由于博弈树本身就具有非对称的特点，博弈树能够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，将会比博弈树表示占用更大的空间。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"。所谓"对称化"，就是为图中的某些变量添加一些"冗余"的或"人工"的状态，以使问题"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，加入的"人工"状态会导致博弈实际结构的模糊化。其次，"对称化"的过程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。

针对上述问题，需要对原有的多-Agent 影响图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，即兼有博弈树和多-Agent 影响图两者的优点。为此，我们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结构化博弈模型—非对称多-Agent 影响图。

2 非对称多-Agent 影响图

非对称多-Agent 影响图是在多-agent 影响图[1] 和非对称影响图[2] 的基础上发展而来的。非对称影响图能对非对称的单-Agent 决策问题进行有效的表示和求解，它将决策问题的非对称性在图形结构中表示出来，因此可以直接从非对称影响图中获得决策问题的非对称信息。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。

一个非对称多-agent 影响图是一个带标记的有向无环图，用 G = (A, N, E, L) 来表示。其中，A 是博弈中 Agents 的集合；结点集 N 可以划分为三个不相交的子集：决策结点集(Nd)，随机结点集(Nc)，效用结

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

点集(Nu)。它们分别表示了博弈中的决策变量、随机变量和效用变量。随机结点表示自然的决策,在图中用椭圆来表示。每一个决策变量与一个 Agent 相关

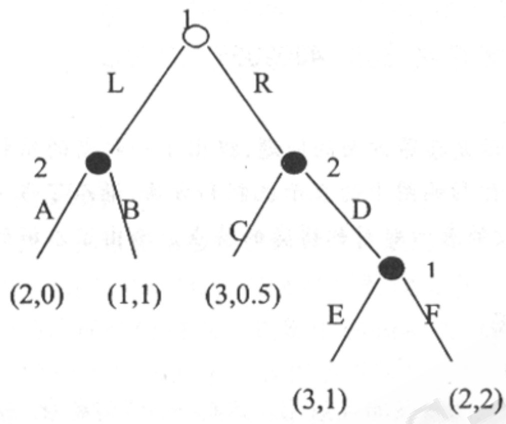

图 1

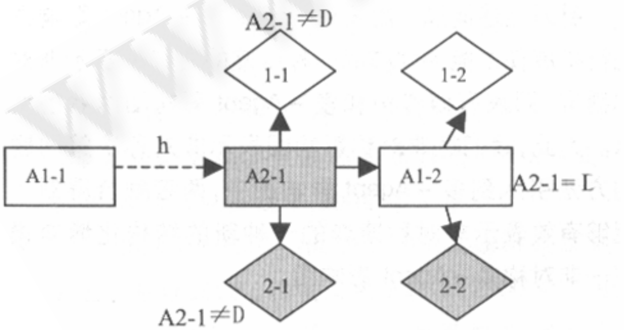

图 2 图 1 所示博弈的非对称多 – agent 影响图 I

联,并由该 Agent 来决定它的取值,决策变量在图中用矩形来表示。每个效用结点也与一个 Agent 相关联,表示该 Agent 的效用,在图中用菱形来表示;边集 E 也可以划分为三个不同的子集:指向随机结点的边,指向效用结点的边和指向决策结点的边(信息边)。与非对称影响图类似,博弈的非对称性在 AMAIDs 中用一个限制边集和一个标记集 L 来表示。限制边在图中用虚线来表示,它是信息边的一个子集,一条信息边(X,D)表示决策点 D 的可选行动集随着变量 X 的取值的不同而不同。一个标记 l 与一个结点或一条信息边相关联,它定义了相关联的结点或信息边在博弈决策场景中出现的条件。与多 – agent 影响图对比,非对称多 – agent 影响图除了具有多 – agent 影响图的结构特点外,在模型中加入了约束弧和标记机制来表示博弈的非对称性;与非对称影响图相比,非对称多 – agent 影响图除了具有非对称影响图中的用约束弧和标记来表示决策问题的非对称性的机制,还将非对称影响图中的单 – agent 决策问题扩展成允许有多个 agents 的博弈情况。

3 非对称多 – Agent 影响图的求解

首先采用非对称影响图中分解非对称影响图的算法 Decomposition(I,C)(详见参考文献[2])将待求解的非对称多 – agent 影响图分解成多个多 – agent 影响图;然后利用多 – agent 影响图中求解多 – 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。

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[1]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。

当前在所有以叶子节点为儿子节点的子树 Ti 中,假设树根为 STi,此时,STi 的所有子节点都已经求解得到均衡策略,设 Mi 是 STi 处的 AMAID,S1Mi 是 Mi 唯一的分裂变量,也是 Mi 的初始分裂变量,在 Mi 中移除 S1Mi 所有的后继节点及与它们相关联的弧,记录下被移除的决策节点的均衡策略及效用向量;在 S1Mi 指向的效用节点(如果 Mi 中没有的话,在其中为每个 Agent 加入一个效用结点)中增加相应的取值入口,指示当 S1Mi 取不同的值时各个 Agents 所能获得的效用。移除 Mi 中的所有标记,将 Mi 转换成一个 MAID Mi'。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。

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

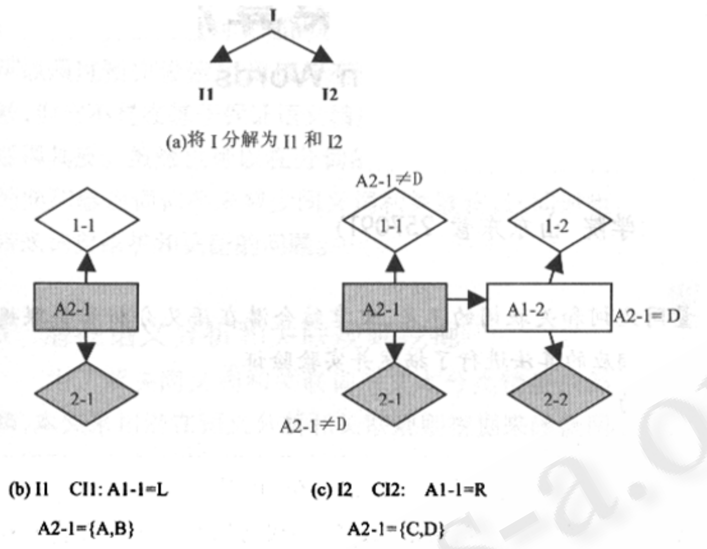

图 3 图 1 所示博弈的非对称多 – agent 影响图 I 的第一次分解

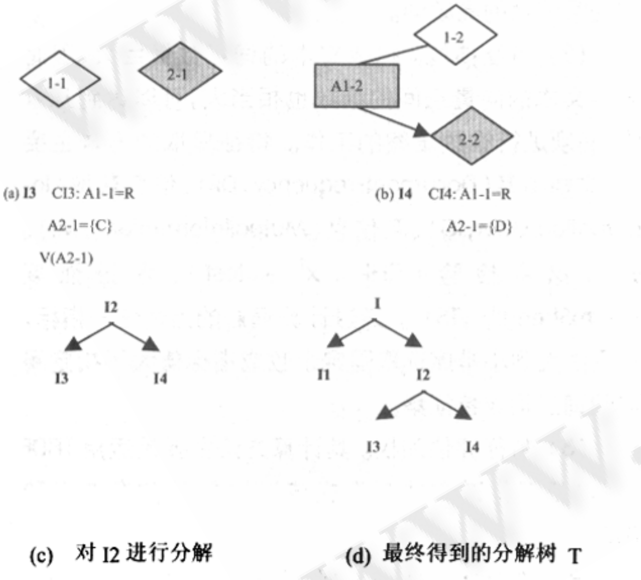

图 4 图 1 子博弈非对称多 – agent 影响图 I2 的分解

4 应用举例

下面通过一个简单的例子说明用非对称多 – agent 影响图来表示和求解博弈的过程。

在该博弈树所表示的博弈中，一共有四个决策场景，在博弈过程中一共有三个决策点，我们设为 A1 – 1, A2 – 1, A1 – 2，它们的可选行动集 A1 – 1 = {L, R}, A2 – 1 = {A, B, C, D}, A1 – 2 = {E, F}, |A1 – 1| * |A2 – 1| * |A1 – 2| = 2 * 4 * 2 = 16 > 5，所以该博弈树所表示的博弈是一个非对称博弈。

下面使用非对称多 – agent 影响图对该博弈进行表示(图 2)。

用算法 Decomposition 进行分解：I：(分解的时候将约束弧当做分裂变量来处理) 初始分裂变量：A1 – 1

在 I1 中已经没有标记和约束弧，它已经是一个多 – Agent 影响图，不需要再进行分解了；在 I2 中仍然有 3 个标记存在，所以它是一个非对称多 – Agent 影响图(图 3)，我们还需要对它进行进一步的分解。见图 4。

得到分解树 T 之后，我们可以用算法 Compute 来对该博弈进行求解，最终结果如图 5。最终 agent 1 获得效用 3，agent 2 获得效用 1。最终得到博弈的均衡解 σ = {(R, E), (B, D)}。

A1 – 1	L	R
(1,2)	(1,1)	(3,1)

图 5 非对称多 – agent 影响图的最终求解

5 结束语

本文提出了一种有效的表示和求解非对称博弈的结构化博弈模型：非对称多 – Agent 影响图，并通过一个实例说明了它的应用。

参考文献

1 D. Koller and B. Milch. Multi – agent influence diagrams for representing and solving games. In IJCAI, pages 1027 – 1034, 2001.

2 Nielsen, T. D. and F. V. Jensen (1999). Representing and solving asymmetric Bayesian decision problems. Technical report, Department of Computer Science, Fredrik Bajers 7C, 9220 Aalborg, Denmark. R – 99 – 5010.

3 施锡铨、博弈论[M]，上海：上海财经大学出版社，2000 年.

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn