第35卷 第21期 计 算 机 工 程 2009年11月
Vol.35 No.21 Computer Engineering November 2009

·人工智能及识别技术· 文章编号：1000—3428(2009)21—0186—03 文献标识码：A 中图分类号：TP18

给定限界的势结构生成算法

李少芳[1]，胡山立[2]

(1. 莆田学院电子信息工程学系，莆田 351100；2. 福州大学计算机科学与技术系，福州 350108)

摘 要：在联盟结构生成过程中，同势的2个联盟通常具有相同值或相似值。在同势同值情况下建立不同联盟的限界时，必须搜索势结构图的最底两层。研究最优势结构生成问题，提出一种给定限界的势结构生成算法，确定需要进一步搜索的势结构。分析结果表明，搜索势结构图的最底两层和顶层后，通过搜索势结构集合，可以得到符合要求的限界。与其他势结构生成算法相比，该算法需要搜索的势结构数最少。

关键词：多Agent；势结构；联盟组合；限界

Cardinality Structure Generating Algorithm with Given Bound

LI Shao-fang[1], HU Shan-li[2]

(1. Department of Electronic Information Engineering, Putian University, Putian 351100;
2. Department of Computer Science and Technology, Fuzhou University, Fuzhou 350108)

【Abstract】During the generating process of coalition structure, two coalitions with same cardinality always have same value or similitude value. It is necessary to search the lowest two levels of the cardinality structure graph while establishing a bound of different coalition with same cardinality and same value. This paper researches the problem about best cardinality structure generating, proposes an algorithm for cardinality structure generating with bound and ascertains cardinality structures need to be further searched. Analysis results show that the required bound can be achieved by searching the cardinality structure set after searching the lowest two levels and the top level of the cardinality structure graph. The number of cardinality structures for searching of this algorithm is minimal comparing with other algorithms for cardinality structure generating.

【Key words】multi-Agent; cardinality structure; coalition combination; bound

Agent 通常需要最大化其个体或集体效益，联盟形成在合作的对策论中被长期研究[1]。文献[2-5]研究特征函数对策下的联盟结构生成问题。在很多实际问题中，不同联盟存在同势同值特征，或同势的2个联盟的值相差不大。例如，在任务分配问题中，工厂接受 m 个大小不同的任务后，需要将员工分成 m 组，而对多数员工而言，被分配到哪一组区别并不大。可见，研究最优势结构生成问题具有理论意义和应用价值。

1 基本概念

设 n 个 Agent 的集合 $A=\{a_1,a_2,\cdots,a_n\}$。A 中的一个非空子集 C 称为 Agent 的一个联盟，联盟 C 中 Agent 的个数称为联盟 C 的势，记为$|C|$。A 的一个完全的划分称为一个联盟结构 CS。设联盟结构 $CS=\{C_1,C_2,\cdots,C_p\}$，不失一般性地，设 $|C_1|\geq|C_2|\geq\cdots\geq|C_p|\geq1$。令 $n_1=|C_1|$，$n_2=|C_2|$，…，$n_p=|C_p|$，$n=n_1+n_2+\cdots+n_p$，称 $\{n_1,n_2,\cdots,n_p\}$ 为联盟结构 CS 的势结构，记为 CCS。

在特征函数对策下研究势结构生成问题，具体如下：(1)每个特征函数 V 仅与联盟 C 的势有关，联盟 C 的势$|C|$的特征值(下文简称为值)表示为 $V(|C|)$，且联盟结构的势结构 CCS 的特征值等于该联盟结构中所有联盟势的特征值的和，即 $V(CCS)=\sum_{|C|\in CCS}V(|C|)$；(2)每个联盟势的特征值有限，对任意联盟 C，假设 $V(|C|)\geq0$；(3)在非超加对策中研究联盟形成，即联盟不具有超加性，或超加性事先不知道。在某一特征函数 V 下，特征值最大的势结构就是该特征函数 V 下的最优势结构，记为 CCS^*。

本文用特征函数 $V(CCS)$给出势结构 CCS 的值或权(即收益)，寻求最优势结构的过程可以视为在对应节点带权的势结构图上的搜索过程。设所有势结构的集合为 M，在特征函数 V 下，设 CCS^*是 M 中值最大的势结构(即最优势结构)，CCS_N^* 是搜索 M 的一个子集 $N\subseteq M$ 得到的局部最优解(即局部最优势结构)。若对任意特征函数 V_α，存在与 V_α无关的 k(一般与 n 和 N 有关)满足：

$$\frac{V_\alpha(CCS^*)}{V_\alpha(CCS_N^*)}\leq k \tag{1}$$

则称 k 是搜索到的解的限界，可以用来衡量最坏情况下搜索得到的局部最优解 CCS_N^* 的质量，它表示最坏情况下已搜索的局部最优解和全局最优解的距离。如果 k 是限界，且存在一个特征函数 V_α，使式(1)中的等号成立，则称限界 k 是紧的。

对于有 n 个 Agent 的系统，所有可能的联盟结构的势结构及其关系可以用势结构图表示，图中节点代表势结构，由底向上共 n 层，第 i 层(L_i)中的节点代表含有 i 个势的势结构，节点间的弧线代表势结构之间的关系，沿弧线向下代表上面的势结构中2个势合并产生下面的势结构，沿弧线向上代表下面的势结构中一个势分裂为 2 个势产生上面的势结构，6个 Agent(1,2,3,4,5,6)的势结构如图1所示。整数 n 的每种划分对应一种势结构，例如，整数 6 的一种划分表示为 6=2+2+1+1，对应势结构{2,2,1,1}，整数 n 的划分数即势结构数，

基金项目：国家自然科学基金资助项目(60573076)

作者简介：李少芳(1971－)，女，副教授、硕士，主研方向：计算机算法设计与分析，人工智能，多 Agent 系统；胡山立，教授

收稿日期：2009-05-18 **E-mail：**LSF9808@163.com

其值可以通过递归方法计算。文献[2]已证明，要建立最坏情况下的限界 k，搜索联盟结构图的最底两层是必要且充分的，此时限界 $k=n$。搜索联盟结构图的最底两层 L_1,L_2 和顶层 L_n，限界 $k=\lceil n/2\rceil$ 且是紧的。

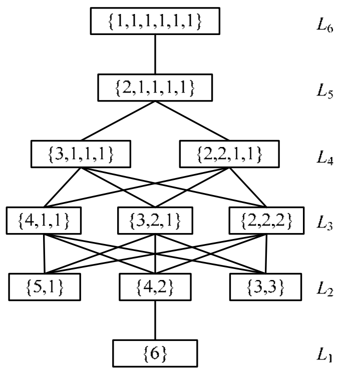

图 1　6 个 Agent 的势结构

定理 1 要建立最坏情况下的限界 k，搜索势结构图的最底两层是必要且充分的，此时限界 $k=n$。

定理 2 搜索势结构图的最底两层 L_1, L_2 和顶层 L_n，限界 $k=\lceil n/2\rceil$ 是紧的。

当实际问题提出比上述限界更优的具体限界要求时，针对如何进行进一步搜索，以最小搜索量满足该要求的问题，文献[3]给出一种以层为单位的最优搜索算法，文献[4]给出一个联盟结构生成算法，文献[5]提出一个更优的联盟结构生成算法。上述算法可以用于势结构生成。

2　势结构生成算法

2.1　相关定义

定义 1 已知 n, k，令 $k^*\leftarrow\lfloor k\rfloor$，取 $h=\lfloor n/k^*\rfloor$，余数 $m=n \bmod k^*$，则 $m<k^*$。若 $m\geq h$，令联盟组合势结构集合 $CCCS(n, k^*)=\phi$，否则 $CCCS(n, k^*)=\{n_1,n_2,\cdots,n_i\}$ 为满足以下条件的集合(其中，$n_1\geq n_2\geq\cdots\geq n_i\geq 1$ 且 $h-1\geq n_1\geq 2$)：

(1)若 $m=0$，则满足 $n_1+n_2+\cdots+n_i\geq h$ 且 $n_1+n_2+\cdots+n_{i-1}<h$；

(2)若 $1\leq m<h$，则满足 $n_1+n_2+\cdots+n_i\geq h+1$ 且 $n_1+n_2+\cdots+n_{i-1}<h+1$，且包括{$h$, 1},{$h$, 2},⋯,{$h$, m}。

例如，当 n=13, k^*=3 时，h=4, m=1,$CCCS$(13,3)={{3,3}, {3,2}, {3,1,1}, {2,2,2}, {2,2,1}, {2,1,1,1}, {4,1}}。

定义 2 设 $SCCS(n, k^*)$为所有势结构的集合，根据 n, k^*，按定义 1 求得 $CCCS(n, k^*)$，记 $CCCS(n,k^*)=\{CC_1,CC_2,\cdots,CC_t\}$，构造 $SCCS(n, k^*)$所有势结构的方法描述如下：

```
Step1  令 sub=1;
Step2 B= CC_sub ，此时 s=|B| ;
Step3 for (i=sub+1;i<=t;i++)
if ( s+|CC_i-B| ≤ n )
{ p=|CC_i-B| ;
sub=i;
B=B∪(CC_i-B) ;/*按降序插入*/
s=s+p ;
continue;    }
else
{ B=B∪{(n-s)个1} ;
  break;    }
输出 SCCS(n, k*)中的一个势结构 B
Step4  令 sub=sub+1,
if (sub>t) return;
else  重复 Step2, Step3, Step4;
```

Step4 中每执行一次 Step3，可以输出 $SCCS(n, k^*)$中的一个势结构 B。因此，可以构造出 $SCCS(n, k^*)$中的所有势结构。而 Step3 中的关键是求 CC_i-B，编写一个自定义函数 *compare*，其功能是在 B 中顺序查找 CC_i 中的元素，找到此类元素后，先记录被找到的 CC_i 中的元素下标，然后在 B 中未搜索过的元素中继续顺序查找 CC_i 中的下一个元素。若在 B 中无法找到 CC_i 中的元素，则返回 0。若在 B 中找到 CC_i 中的全部元素，则返回 $|CC_i|$。可以通过在主调函数中加入语句 p=*compare*(B, CC_i)来调用 *compare* 实现在 B 中添加 CC_i 中从 p+1 到 $|CC_i|$ 的元素。为方便叙述，要求在 Step3 中按降序插入新添加的元素，所求的 $SCCS(n, k^*)$中的元素是有序的，实际应用中不一定要求按序构造。

例如，当 n=12, k^*=3, h=4, m=0 时，$CCCS$(12, 3)={{3,3}, {3,2}, {3,1}, {2,2}, {2,1,1}}; $SCCS$(12,3)={3,3,2,2,1,1}。

当 n=14, k^*=3, h=4, m=2 时，$CCCS$(14, 3)={{3,3}, {3,2}, {3,1,1}, {2,2,2}, {2,2,1}, {2,1,1,1}, {4,1}, {4,2}, $SCCS$(14,3)={{3,3,2,2,2,1,1}, {4,2,1,1,⋯,1}}。

2.2　算法描述与基本性质

势结构生成算法的具体步骤如下：

步骤 1 获得所要求的限界 k，搜索势结构图的最底下 2 层 L_1, L_2 和顶层 L_n。若 $k\geq\lceil n/2\rceil$，则转步骤 3，若 $1\leq k<2$，则搜索整个势结构图后转步骤 3。

步骤 2 令 $k^*\leftarrow\lfloor k\rfloor$，搜索势结构集合 $SCCS(n, k^*)$(若 $SCCS(n, k^*)$为 ϕ，则直接跳到下一个 k^*)，一直搜索到时间不允许或搜索完整个联盟结构图为止。

步骤 3 返回得到的最优势结构。

设最优势结构 CCS^*包含 t 个联盟 $C_1,C_2,\cdots,C_t$ 的势，且 $|C_1|\geq|C_2|\geq\cdots\geq|C_t|\geq 1$。令 $n_1=|C_1|,n_2=|C_2|,\cdots,n_t=|C_t|$，且 $n_1+n_2+\cdots+n_t=n$，$CCS^*=\{n_1,n_2,\cdots,n_t\}$。

将 CCS^*按以下规则分为 a 组：

当 $m<h$ 时，将所有 n_i ($n_i\geq h$)各单独分成一组。对所有 $n_{i+1},n_{i+2},\cdots,n_j$，若满足 $n_{i+1}+n_{i+2}+\cdots+n_j\geq h$ 且 $n_{i+1}+n_{i+2}+\cdots+n_{j-1}<h$ ($m=0$ 时)或 $n_{i+1}+n_{i+2}+\cdots+n_j\geq h+1$ 且 $n_{i+1}+n_{i+2}+\cdots+n_{j-1}<h+1(1\leq m<h)$，则将它们各分为一组{ $n_{i+1},n_{i+2},\cdots,n_j$ }。若有元素剩余，则将剩余元素分为一组。

可知 $1\leq a\leq\lceil n/h\rceil$，当 m=0 时 $\lceil n/h\rceil=k^*$，否则 $\lceil n/h\rceil=k^*+1$。

例如，n=13, k^*=3，若 CCS^*={4,3,3,2,1}，则可以分为{{4}, {3,3}, {2,1}} 3 组，若 CCS^*={4,2,2,2,2,1}，则可分为{{4}, {2,2,2}, {2,1}} 3 组。

引理 1 搜索完势结构集合 $SCCS(n,k^*)$ 后，CCS^*中各分组都已被搜索过。

证明：

(1)若组里只有一个元素，则该组{ n_i }在 L_2 上的势结构{ $n_i, n-n_i$ }中已被搜索过。

(2)若组里有 2 个或 2 个以上元素，则设为组{ $n_{i+1},n_{i+2},\cdots,n_j$ }，若 $n_{i+1}=1$，则该组在 L_n 上的势结构{1, 1,⋯,1}中已被搜索过，否则各组中的 $n_{i+1},n_{i+2},\cdots,n_j$ 一定满足 $n_{i+1}+n_{i+2}+\cdots+n_j\geq h$ 且 $n_{i+1}+n_{i+2}+\cdots+n_{j-1}<h$ ($m=0$ 时) 或 $n_{i+1}+n_{i+2}+\cdots+n_j\geq h+1$ 且 $n_{i+1}+n_{i+2}+\cdots+n_{j-1}<h+1(1\leq m<h)$。上述组在势结构集合 $SCCS(n, k^*)$内已被搜索过。

(3)对于剩余元素构成的组，组中所有元素的和小于 h ($m=0$时)或小于 h+1($1\le m<h$时)。可以按 $CCCS(n,k^*)$的规则进行补充，直到该新组的元素和 s 满足

$$\begin{cases} s\ge h & m=0 \\ s\ge h+1 & m\ne 0 \end{cases}$$

该新组在势结构集合 $SCCS(n, k^*)$内已被搜索过，因此，剩余元素构成的组在势结构集合 $SCCS(n, k^*)$内被搜索过。

引理 2 若 CCS^*中的元素可以分成 a 组，且 a 组中每组的元素都同时出现在已被搜索过的同一个势结构中，则限界 $K(n)\le a$。

证明：设 CCS^*可以分为 $g_1,g_2,\cdots,g_a$ 共 a 组，其中，第 i 组 $(1\le i\le a)$ $g_i=\{n_{ip},n_{ip+1},\cdots,n_{iq}\}$的值为$V(g_i)=V(n_{ip})+V(n_{ip+1})+\cdots+V(n_{iq})$，设第 d 组($1\le d\le a$)的值最大为$V(g_d)=\max\limits_{1\le i\le a}\{V(g_i)\}=V\max$，由于 a 组中的每一组已被搜索过，因此第 d 组肯定已被搜索过，可知 $V(CCS^*)\ge V\max$，$V(CCS^*)=V(g_1)+V(g_2)+\cdots+V(g_a)\le a\cdot V\max$，因此，$\frac{V(CCS^*)}{V(CCS_N^*)}\le \frac{aV\max}{V\max}=a$，即限界 $K(n)\le a$。

定理 3 对给出的限界 $k\,(2\le k\le\lceil n/2\rceil-1)$，令$k^*\leftarrow\lfloor k\rfloor$，若算法刚搜索完势结构集合 $SCCS(n, k^*)$，则限界 $K(n)\le k^*$，且是紧的。

证明：

(1)当 m=0 时，$n=h\cdot k^*$，按上述分组规则，把 CCS^*中的元素分成 a 组，由于每组元素和大于等于 h，因此 $a\le k^*$，即最多可以分为 k^*组，而每组中的元素均同时被搜索过，由引理 1 和引理 2 可知，此时限界 $K(n)\le k^*$。

下文将证明当 m=0 时，限界 $K(n)\le k^*$是紧的。

设 n 个 Agent 的集合 $A=\{a_1,a_2,\cdots,a_n\}$，$n=hk^*$，设联盟 $C_i=\{a_{(i-1)h+1},a_{(i-1)h+2},\cdots a_{ih}\}$，则有$|C_i|=h$，$i=1,2,\cdots,k^*$。设 V 是任意一个给定的特征函数，$CCS^*=\{n_1,n_2,\cdots,n_t\}$是最优势结构，令$V(h)=1$，任意其他势不等于 h 的特征值为 0，即 $V(w)=0,w\ne h$。取 $CCS^*=\{|C_1|,|C_2|,\cdots,|C_{k^*}|\}=\{h,h,\cdots,h\}$，则 $V(CCS^*)=k^*$。根据定义 5，任何 2 个大小为 h 的联盟都不可能同时出现在已搜索过的势结构中，即$V(CCS_N^*)=1$，因此

$$\frac{V(CCS^*)}{V(CCS_N^*)}=\frac{k^*}{1}=k^*$$

即限界 $K(n)=k^*$是紧的。

(2)当$1\le m<h$时，$n=hk^*+m$，按上述分组规则把 CCS^*中的元素分成 a 组，除最后剩余元素的组外，其他组的元素和大于等于 h，且至多 m 个组的和大于 h，可知$1\le a\le\lceil n/h\rceil=k^*+1$。当至多含一个组$\{h\}$时，$a\le k^*$，即 CCS^*最多可以分为 k^*组，且每一组中的元素均同时被搜索过，由引理 1 和引理 2 可知，此时限界$K(n)\le k^*$。否则当 $a=k^*+1$时，设 CCS^*中组$\{h\}$的个数为 $p\,(p\ge 2)$，且第 k^*+1 组中元素和为 q，则 $p\cdot h+(k^*-p)(h+1)+q\le n$，即 $k^*\cdot(h+1)-p+q\le n$，又因为 $n=k^*\cdot h+m<k^*\cdot(h+1),(\because m<h)$，所以 $q<p$。

上述分析表明第 k^*+1 组中元素的个数少于 CCS^*中组$\{h\}$的个数，可以将第 k^*+1 组中的各元素分配到 p 组$\{h\}$中的不同组中，使 CCS^*最多可以分为 k^*组。由于$\{h,1\},\{h,2\},\cdots,\{h,m\}$的组合在 $CCCS(n, k^*)$中已出现过，因此这些可能的新组和其他各组在 $SCCS(n, k^*)$上已被搜索过，由引理 1 和引理 2 可知，此时限界 $K(n)\le k^*$。

下文将证明当$1\le m<h$时，限界$K(n)\le k^*$是紧的。

设 n 个 Agent 的集合 $A=\{a1,a2,\cdots,an\}$, $n=hk^*+m$，设联盟 $C_i=\{a_{(i-1)h+1},a_{(i-1)h+2},\cdots a_{ih}\}$，则有$|C_i|=h$，$i=1,2,\cdots,k^*$。设 V 是任意一个给定的特征函数，$CCS^*=\{n_1,n_2,\cdots,n_t\}$是最优势结构，令$V(h)=1$，其他势不等于 h 的特征值为 0，即$V(w)=0,w\ne h$，取 $CCS^*=\{|C_1|,|C_2|,\cdots,|C_{k^*}|,|\{a_{n-m+1},a_{n-m+2},\cdots,a_n\}|\}=\{h,h,\cdots,h,m\}$，则 $V(CCS^*)=k^*$。根据定义 5，任何 2 个大小为 h 的联盟都不可能同时出现在已搜索过的势结构中，即 $V(CCS_N^*)=1$，因此$\frac{V(CCS^*)}{V(CCS_N^*)}=\frac{k^*}{1}=k^*$，即限界 $K(n)=k^*$是紧的。

3 相关算法的比较

文献[3]的联盟结构生成算法讨论了以最少搜索层数达到给定限界要求，并给出一种以层为单位的最优搜索算法，具体描述如下：在搜索联盟结构图的最底两层后，当给定限界 $k\ge 3$时，只搜索一层就得到满足限界的解；当$2\le k<3$时，最多需要搜索 2 层就能得到满足限界的解。文献[4]给出的联盟结构生成算法对于奇数限界 $k\ge 3$的情况，在搜索最底两层和顶层后，进一步搜索最大联盟的势大于等于$\lceil n(k-1)/(k+1)\rceil$的所有联盟结构。文献[5]基于势结构的联盟结构生成算法在搜索最底两层和顶层后，进一步搜索势结构集合为 $CCS(n,k)$ 的所有联盟结构。

以 n=12, k=3 为例，文献[3]算法需要搜索第 8 层的 5 个势结构，文献[4]算法需搜索最大联盟的势大于等于 6 的 23 个势结构 {10,1,1},{9,2,1},{9,1,1,1},{8,3,1},{8,2,2},{8,2,1,1},{8,1,1,1,1},{7,4,1},⋯,{7,1,⋯,1},{6,5,1},{6,4,2},⋯,{6,1, ⋯,1}，文献[5]算法需要搜索的 3 个势结构为{3,3,1,⋯,1},{3,2,1,⋯,1},{2,2,1,⋯,1}，本文势结构生成算法必须搜索的势结构仅一个，即{3,3,2,2,1,1}。以 n=50, k=8 为例，文献[3]算法需要搜索第 42 层的 22 个势结构，文献[4]算法需要搜索最大联盟的势大于等于 39 的 183 个势结构{48,1,⋯,1}, {47,1,⋯,1}, {47,2,1,⋯,1},⋯, {39,6,5}，文献[5]算法需要搜索的 12 个势结构为{5,5,1,⋯,1},⋯,{5,2,1,⋯,1},{4,4,1,⋯,1},⋯,{4,2,2,1,⋯,1},{3,3,3,1,⋯,1},⋯,{3,2,2,1,⋯,1},{2,2,2,2,1,⋯,1},{6,2,1,⋯,1}，本文势结构生成算法必须搜索的势结构仅一个，即{6,5,5,4,4,3,3,3,2,2,2,2,1,⋯,1}。当 n=50 时，各算法在搜索完层 L_1, L_2和 L_n后，需要进一步搜索的势结构数如表 1 所示。

表 1 n=50 时各算法搜索的势结构数

k	文献[3]算法	文献[4]算法	文献[5]算法	本文势结构生成算法
2	5 507	-	1 377	218
3	1 002	9 270	256	40
4	385	-	95	19
5	135	898	45	5
6	77	-	27	2
7	42	259	18	2
8	22	-	12	1
9	22	128	12	1
10	11	-	6	1
11	11	58	6	1
12	5	-	4	1
13	5	37	4	1
14	5	-	4	1
15	5	23	4	1
16	2	-	2	1
17	2	13	2	1
18	2	-	2	1
19	2	13	2	1
20	2	-	2	1
21	2	7	2	1
22	2	-	2	1
23	2	7	2	1
24	2	-	2	1

（下转第 191 页）