ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.18, No.3, March 2007, pp.574−581
DOI: 10.1360/jos180574

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

多 Agent 联盟结构动态生成算法*

张新良+, 石纯一

(清华大学 计算机科学与技术系,北京 100084)

A Dynamic Formation Algorithm of Multi-Agent Coalition Structure

ZHANG Xin-Liang+, SHI Chun-Yi

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

+ Corresponding author: Phn: +86-10-62783505, E-mail: xinliangzhang@alcatel-lucent.com

Zhang XL, Shi CY. A dynamic formation algorithm of multi-Agent coalition structure. *Journal of Software*, 2007,18(3):574−581. http://www.jos.org.cn/1000-9825/18/574.htm

Abstract: To solve the number of coalition structure increase rapidly, algorithm SCS (search of coalition structure), fast dynamic formation of Agent coalition, is given. It can prune the graph of Agent coalition structure, decrease the searching space., and proved that after pruning, the number of coalition structure is $\frac{n}{(k-1)^{n-k}}$ of that before pruning. Finally, an experiment is given. This work can be seen as an improvement of Jennings and Sandholm's related work.

Key words: multi-Agent system; coalition; coalition structure; SCS algorithm

摘 要: 针对多 Agent 联盟数量是 Agent 个数指数倍的问题,基于 Agent 合作收益独立性,给出了 Agent 联盟快速动态生成算法——SCS(search of coalition structure)算法;依 Agent 联盟之间的同构关系,将 Agent 联盟结构图剪枝,然后进行 Agent 联盟结构搜索,可降低搜索空间大小,并证明了是剪枝前搜索量的 $\frac{n}{(k-1)^{n-k}}$.最后,以机器人足球赛 RoboCup 为背景给出了实验分析,表明了 SCS 算法的效率.SCS 算法是对 Jennings 和 Sandholm 等人相关工作的改进.

关键词: 多 Agent 系统;联盟;联盟结构;SCS(search of coalition structure)算法
中图法分类号: TP18 文献标识码: A

多 Agent 合作求解是多 Agent 系统(multi-Agent system,简称 MAS)理论与技术研究的重点,Agent 联盟可看作是 Agent 合作求解的一种形式.在 MAS 中,联盟内的 Agent 通常采取合作策略,而不同联盟内的 Agent,则采取一般的求解策略.Agent 如何联盟以使整体收益最大是多 Agent 合作求解的关键.近年来,越来越多的研究者考虑了 Agent 动态联盟形成算法[1−7].Jennings[1]采用对联盟结构中最大联盟规模不小于$\lceil n(q-1)/q \rceil$的逐步搜索的办法,其中,$q=\lfloor (n+1)/4 \rfloor$到 n,可得到收益不低于最优收益的 $1/(2q-1)$的联盟结构.Anderson[2]分析了 RoboCup 中足球机器人之间的联盟动态形成问题.文献[3]中,Klusch 等人提出了 Agent 联盟 sate core 状态,用来表示 Agent 联

* Supported by the National Natural Science Foundation of China under Grant Nos.60573076, 60496323 (国家自然科学基金)
Received 2005-01-27; Accepted 2006-02-23

盟的均衡状态.Gerber[4]和 Konishi[5]则重点考虑了 Agent 理性在多 Agent 联盟动态形成过程中的影响问题.Sanholm 等人[6]证明了对联盟结构图搜索完第 $L1,L2,\ldots,Ln$ 层后,可得到收益不低于最优收益的 $1/\lceil n/2\rceil$的 Agent 联盟.Dant[8]等人给出用动态规划的方法搜索 Agent 联盟,可以求出次优的 Agent 联盟结构.

但是,对于 n 个 Agent,可能的联盟结构个数是 $Sum(n)=\sum_{i=1}^{n}S(n,i)$,其中:$S(n,i)$是第 2 类 Striling 数;$Sum(n)$呈 n 的指数倍增长[1].因此,Sandholm 和 Jennings 等人的算法虽然可以保证搜索到最优解,但其搜索量非常大.而 Dant 等人[8]的算法虽然可以降低搜索量,但是只能保证搜索到次优的 Agent 联盟结构.本文给出了一种对 Agent 联盟结构图自上而下的搜索算法.在基于 Agent 合作收益独立性假设的基础上,该算法可以对联盟结构图进行化简,对化简后的联盟结构图进行搜索,其搜索量是化简前的 $\frac{n}{(k-1)^{n-k}}$.与 Jennings 和 Sandholm 等人的结果相比,可以提高联盟生成效率,并且能够保证所得 Agent 联盟结构是最优的.

1 Agent 联盟优化问题

设 MAS 中有 n 个 Agent,$MAS=\{1,2,\ldots,n\}$.MAS 中的一个非空子集 C 称为一个 Agent 联盟,而 MAS 的一个完全的划分称为一个联盟结构 CS_k,即

定义 **1**. 设 MAS 中有 n 个 Agent,记 $MAS=\{1,2,\ldots,n\}$,MAS 中的一个非空子集 C 称为一个 Agent 联盟.而联盟结构 $CS_k=\{C_1,C_2,\ldots,C_k\}$,其中:$C_i\subseteq MAS;\bigcup_{i=k}^{k}C_i=MAS$;$C_i\cap C_j=\varnothing,i,j=1,\ldots,k,i\neq j$,并称联盟结构 CS_k 的大小为 k.

对 Agent 联盟结构 $CS_k=\{C_1,C_2,\ldots,C_k\}$来说,称联盟结构 $CS_{k-1}=\{C_1,\ldots C_{i-1},C_{i+1},\ldots C_{j-1},C_j\ldots,C_k,C_i\cup C_j\}$是由 CS_k 生成的,记作 $CS_k\rightarrow CS_{k-1}$.显然,每个 CS_k 可以生成 $k(k-1)/2$ 个 CS_{k-1}.例如,$\{(1),(2),(3)\}\rightarrow\{(1,2),(3)\},\{(1),(2),(3)\}\rightarrow\{(1),(2,3)\}$.如果 $CS_k\rightarrow CS_{k-1},CS_{k-1}\rightarrow CS_{k-2}$,则 $CS_k\rightarrow CS_{k-2}$,即生成关系具有传递性.$G(CS_k)=\{CS_i|CS_k\rightarrow CS_i\}$是 CS_k 生成的所有 Agent 联盟结构所组成的集合.

例如:对于 $MAS=\{1,2,3\}$,有 7 个可能的 Agent 联盟:(1),(2),(3),(1,2),(1,3),(2,3),(1,2,3),有 5 个可能的 Agent 联盟结构$\{(1),(2),(3)\},\{(1,2),(3)\},\{(1,3),(2)\},\{(1),(2,3)\},\{(1,2,3)\}$.所有大小为 k 的联盟结构记作 $L_k=\{CS_k\}$,所有的联盟结构记作 $L=\bigcup_{k=1}^{n}L_k$.

可以将所有的 Agent 联盟结构画为一个 n 层 Agent 联盟结构图,其中第 k 层是 L_k 中所有的 Agent 联盟结构.在相邻的两层 k,$k-1$ 之间,如果 $CS_k\rightarrow CS_{k-1}$,则 CS_k 和 CS_{k-1} 之间画一条线,如图 1 所示.

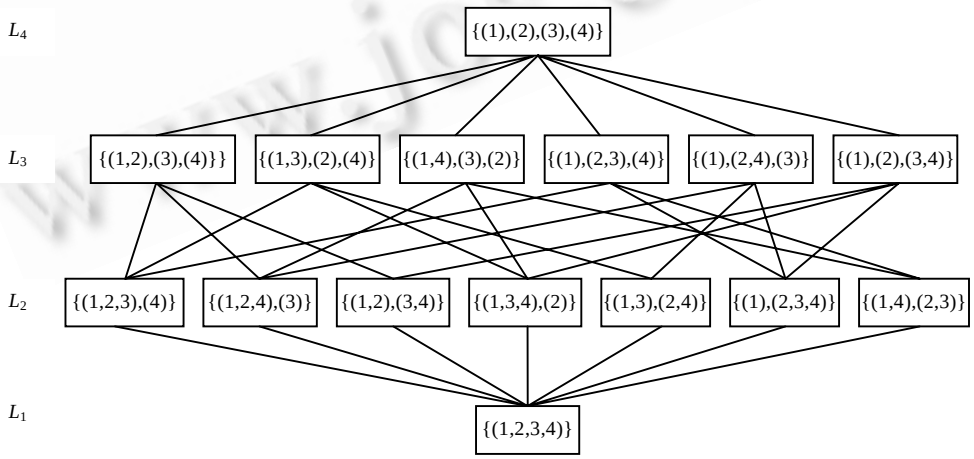

Fig.1 The graph of Agent coalition of $MAS=\{1,2,3\}$

图 1 $MAS=\{1,2,3,4\}$的联盟结构图

每个 Agent 联盟 C_i 都有一个收益 $u(C_i)$,一个 Agent 联盟结构 CS_k 的收益 $V(CS_k)=\sum_{i=1}^{k}u(C_i)$.给定所有的 Agent 联盟 C_i 的收益,Agent 联盟结构优化问题就是找出收益最大的 Agent 联盟结构 CS^*,$V(CS^*)=\text{argmax}_{cs\in L}V(CS)$.

Agent 联盟结构图中第 k 层 G_k 的数量是 $S(n,k)$,第 2 类 Stirling 数.因此,所有的 Agent 联盟结构个数是 $\sum_{k=1}^{n}S(n,k)$.Sandholm 证明了它的计算复杂度是 $O(n^n)$,Agent 联盟结构优化问题是 NP 的.

如果 $u(C)$是任意的,则为了找到 CS^*,需要搜索所有可能的 Agent 联盟结构.为了提高 Agent 联盟结构图的搜索效率,需要对 Agent 联盟收益函数 $u(C)$做一些必要的限制.下面给出算法 SCS(search of coalition structure),在 Agent 联盟收益函数满足 Agent 合作收益性独立的前提下,可以对 Agent 联盟结构图进行剪枝化简,化简后的 Agent 联盟结构收益图,L_k 层的搜索量是化简前的 $\frac{n}{(k-1)^{n-k}}$.

2 Agent 联盟结构优化算法

2.1 Agent合作收益独立性

在实际问题中,有这么一类问题,Agent 合作收益具有独立性.例如:对于 Agent 1,2,3,如果 $u(\{1,2\})+u(\{3\})>u(\{1,3\})+u(\{2\})$,即 1 和 2 合作比 1 和 3 合作要好,则新来一个 Agent 4,那么,{1,4}与 2 合作仍然比{1,4}与 3 合作收益要好,即新加入的 Agent 不影响加入前其他 Agent 联盟之间收益的大小关系.这个性质可以描述为:

性质 **1**. 如果 $C=C_1\cup C_2=C_1'\cup C_2'$,且 $u(C_1)+u(C_2)\geq u(C_1')+u(C_2')$,$C_3\cap C=\varnothing$,则

$$u(C_1\cup C_3)+u(C_2)\geq u(C_1'\cup C_3)+u(C_2'),$$
$$u(C_1\cup C_3)+u(C_2)\geq u(C_1')+u(C_2'\cup C_3),$$

称作 Agent 合作收益独立性.

2.2 Agent联盟结构图化简

在满足 Agent 合作收益独立性的条件下,可以对 Agent 联盟结构图进行化简.

定义 **2**. 如果 $CS_k=\{C_1,C_2,\ldots,C_{k-1},C_k\}$, $CS_k'=\{C_1',C_2',\ldots,C_{k-1}',C_k'\}$, $C_i=C_i'$, $i=1,\ldots,k-2$,

$$|C_1|\leq|C_2|\leq\ldots\leq|C_{k-1}|\leq|C_k|,|C_1'|\leq|C_2'|\leq\ldots\leq|C_{k-1}'|\leq|C_k'|,\text{且 } 2\leq k\leq n-2,$$

则称 CS_k 和 CS_k' 同构,记作 $CS_k\cong CS_k'$,其中,$|C_i|$表示 Agent 联盟 C_i 内 Agent 的个数.即同构的两个 Agent 联盟结构,只有其中最大的两个联盟不同.

性质 **2**. 如果 $CS_{k1}\cong CS_{k2}$, $CS_{k2}\cong CS_{k3}$,则 $CS_{k1}\cong CS_{k3}$.

定理 **1**. 如果 CS_k 和 CS_k' 同构,且 $V(CS_k)\geq V(CS_k')$,则 $\max_{cs_{k-1}\in G(CS_k)}V(CS_{k-1})\geq\max_{cs_{k-1}'\in G(CS_k')}V(CS_{k-1}')$.

证明:由于 $CS_k=\{C_1,C_2,\ldots,C_{k-2},C_{k-1},C_k\}$, $CS_k'=\{C_1',C_2',\ldots,C_{k-2}',C_{k-1}',C_k'\}$,且 $V(CS_k)\geq V(CS_k')$,

所以,$u(C_1)+u(C_2)+\ldots+u(C_{k-2})+u(C_{k-1})+u(C_k)\geq u(C_1')+u(C_2')+\ldots+u(C_{k-2}')+u(C_{k-1}')+u(C_k')$.

由于 $C_i=C_i'$, $i=1,\ldots,k-2$,因此,

$$u(C_{k-1})+u(C_k)\geq u(C_{k-1}')+u(C_k') \tag{1}$$

所以,对于任意的 $CS_i'=\{D_1',\ldots,D_i'\}\in G(CS_k')$(其中: $D_i'=C_k'\cup C_{i1}'\cup\ldots\cup C_{ik}'$, $D_{i-1}'=C_k'\cup C_{(i-1)1}'\cup\ldots\cup C_{(i-1)(k-1)}'$; $D_j'=C_{j1}'\cup\ldots\cup C_{jnj}'$, $j=1,\ldots,i-2$), 存在 $CS_i=\{D_1,\ldots,D_i\}\in G(CS_k)$, 其中 :$D_i=C_k\cup C_{i1}\cup\ldots\cup C_{ik}$;$D_{i-1}=C_k\cup C_{(i-1)1}\cup\ldots\cup C_{(i-1)(k-1)}$; $D_j'=C_{j1}\cup\ldots\cup C_{jnj}$, $j=1,\ldots,i-2$;$D_j=D_j'$, $j=1,\ldots,i-2$.

根据 Agent 联盟收益独立性和公式(1),有 $u(D_i)+u(D_{i-1})\geq u(D_{i-1}')+u(D_i')$.

由于 $D_j=D_j'$, $j=1,\ldots,i-2$,

$$\begin{aligned}u(D_1)+\ldots+u(D_{i-2})+u(D_{i-1})+U(D_i)&\geq u(D_1)+\ldots+u(D_{i-2})+u(D_{i-1})+U(D_i)\\&=u(D_1)+\ldots+u(D_{i-2})+u(D_{i-1}')+U(D_i')\\&=u(D_1')+\ldots+u(D_{i-2}')+u(D_{i-1})+U(D_i'),\end{aligned}$$

所以,$V(CS_i)\geq V(CS'_i)$.

因此,对任意的 $CS'_i\in G(CS'_k)$,均存在 $CS_i\in G(CS_k)$,使得 $V(CS_i)\geq V(CS'_i)$,

所以, $\max_{cs_{k-1}\in G(CS_k)}V(CS_{k-1})\geq\max_{cs'_{k-1}\in G(CS'_k)}V(CS'_{k-1})$. □

定理 1 说明:如果在 $CS_{k1},CS_{k2},\ldots,CS_{kt}$ 之间互相同构,且 $V(CS_{k1})\geq V(CS_{k2})\geq\ldots\geq V(CS_{kt})$,则在它们所生成的 k–1 层的 Agent 联盟结构中,收益最大的 Agent 联盟必然属于 $G(CS_k)$.因此,从上往下搜索 Agent 联盟结构时,执行完第 k 层搜索,则可将这一层的 Agent 联盟按同构关系分组.当搜索第 k–1 层时,只需搜索 k 层每一组同构的 Agent 联盟结构中收益最大的 Agent 联盟结构所生成的 k–1 层的联盟结构即可.

2.3 SCS算法

将上述过程可以描述成算法 SCS.

算法. SCS.

$i=n$

$CS^*=CS_n$

while (i>=1)

{

列出所有 i 层所剩的所有 Agent 联盟结构 CS_i

If $CS_i\cong CS'_i$,且 $V(CS_i)>=V(CS'_i)$

则将 $G(CS'_i)$删除 //删除同构的 Agent 联盟中所有的收益不是最大的

If 存在 $CS_i,V(CS_i)>V(CS^*)$

$CS^*=CS_i$

$i=i-1$

}

Return CS^*

End_SCS

3 SCS 算法分析

SCS 是一种剪枝算法.在搜索到第 k 层时,对第 k 层的 Agent 联盟结构按同构关系分类.对每一类同构关系的 Agent 联盟结构,将不是最大的 Agent 联盟结构及其生成的联盟结构删除.因此,可以显著地化简 Agent 联盟结构图.下面证明采用这种算法,与文献[1,2]的完全搜索算法相比,第 k 层的搜索量可以降低为 $\frac{n}{(k-1)^{n-k}}$ 倍.

定理 **2**. 对于 $C_k=\{C_1,C_2,\ldots,C_k\}$,如果$|C_1|\leq|C_2|\leq\ldots\leq|C_k|$,则 $n-(k-2)\leq|C_{k-1}|+|C_k|\leq n-(k-2)\lceil n/k\rceil$.

证明:由于 $1\leq|Ci|\leq\lceil n/k\rceil,i=1,\ldots,k-2$,

$k-2\leq|C_1\cup C_2\cup\ldots\cup C_k|=|C_1|+|C_2|+\ldots+|C_k|\leq(k-2)\lceil n/k\rceil$,

$|C_{k-1}|+|C_k|=n-|C_1|+|C_2|+\ldots+|C_k|$,

所以,$n-(k-2)\leq|C_{k-1}|+|C_k|\leq n-(k-2)\lceil n/k\rceil$. □

令 $n-(k-2)=h_1,n-(k-2)\lceil n/k\rceil=h_2$,因此,第 k 层 Agent 联盟可以按$|C_{k-1}|+|C_k|$从 h_1 到 h_2 分为 h_2-h_1 组.令 $G_k(t)$表示第 k 层 Agent 联盟中$|C_{k-1}|+|C_k|=t$ 的联盟的集合,即 $G_k(t)=\{CS_k=\{C_1,C_2,\ldots,C_k\},|C_{k-1}|+|C_k|=t\}$,则 $L_k=\bigcup_{t=h_1}^{h_2}G_k(t)$.

令 $G_k(t,D)=\{CS_k=\{C_1,C_2,\ldots,C_k\},|C_{k-1}|+|C_k|=t,(C_{k-1}\cup C_k)=D\}$,即 $G_k(t,D)$中的 Agent 联盟结构,其最后两个联盟 C_{k-1},C_k 的并集都等于 D,并且$|D|=t$.一共有 C_t^n 个 $G_k(t,D)$.

定理 **3**. 采用 SCS 算法,如果 Agent 联盟收益满足性质 1,搜索完第 k 层后,至多剩下其中$\lceil n/t\rceil$个 $G_k(t,D)$.

证明:假设从 n 到 k–1 层,所有的 Agent 联盟已按上述方法排序,则可证明对于 k 层的 $G_k(t,D_1)$和 $G_k(t,D_2)$,如

果 $D_1\cap D_2\neq\varnothing$,SCS 算法执行完第 k 层以后,$G_k(t,D_1)$和 $G_k(t,D_2)$只能保留一个.

令 $D_1\cap D_2=D'$,任取 $CS_1\in G_k(t,D_1)$,$CS_2\in G_k(t,D_2)$,

$$CS_1=\{C_1,C_2,\ldots,C_{k-2},C_{k-1},C_k\},$$
$$CS_2=\{C'_1,C'_2,\ldots,C'_{k-2},C'_{k-1},C'_k\}.$$

其中:$(C_{k-1}\cup C_k)\cap(C'_{k-1}\cup C'_k)=D$;

$$(C_{k-1}\cup C_k)\backslash D'\subseteq\cup C'_i;$$
$$(C'_{k-1}\cup C'_k)\backslash D'\subseteq\cup C'_j,$$

则从 k 层往上看,根据假设,必存在 $p>k$,

$$CS_1=\{C_1,\ldots,C'_i,\ldots,D'\},$$
$$CS_2=\{C_1,\ldots,C'_j,\ldots,D'\}.$$

CS_1 和 CS_2 都属于 $Gp(t',D')$.根据假设,由于 $Gp(t',D')$中只保留一个,因此,CS_1 和 CS_2 至少有一个不保留.因此,CS_1 和 CS_2 至少有一个不保留.这说明:如果 $D_i\cap D_j=\varnothing$,则算法在第 k 层结束后,只能保留其中一个.因为$|D_i|=t$,则这样的 D_i 不超过$\lceil n/t\rceil$个. □

定理 **4**. 对于任意固定的 D,其中 $G_k(t,D)$中 Agent 联盟结构按同构关系分组,最多可以分为 $S(n-t,k-2)$组.

证明:每组中的 Agent 联盟结构 $CS=\{C_1,C_2,\ldots,C_{k-2},C_{k-1},C_k\}$,$CS'=\{C'_1,C'_2,\ldots,C'_{k-2},C'_{k-1},C'_k\}$,其中,$C_i=C'_i$,$i=1,\ldots,k-2$.因此,$\{C_1,C_2,\ldots,C_{k-2}\}=\{C'_1,C'_2,\ldots,C'_{k-2}\}$.而不同组中的 CS,CS',必然存在 $C_i\neq C'_i$,$1\le i\le k-2$.因此,$\{C_1,C_2,\ldots,C_{k-2}\}\neq\{C'_1,C'_2,\ldots,C'_{k-2}\}$.

这相当于对除了 D 以外的 $n-t$ 个 Agent 分为 $k-2$ 个联盟结构,数目是 $S(n-t,k-2)$. □

定理 1 说明可以按$|C_{k-1}\cup C_k|$分为 $t=h_1-h_2$ 组 $G_k(t)$.对于每一组 $G_k(t)$,算法执行到第 k 层时,根据定理 3,可只保留$\lceil n/t\rceil$个 $G_k(t,D)$,对于每一组 $G_k(t,D)$,根据定理 4,又存在 $S(n-t,k-2)$个同构的联盟,每个同构的联盟只保留其中一个,而每个可以生成 $k(k-1)/2$ 个 $k-1$ 层 Agent 联盟结构.因此,算法执行到第 k 层时,第 $k-1$ 层保留的联盟结构数为

$$NS(k-1)=\sum_{t=h_1}^{h_2}\lceil n/t\rceil S(n-t,k-2)k(k-1)/2.$$

由于对于 $t_1<t_2$,$\lceil n/t_1\rceil>[n/t_2]$, $S(n-t_1,k-2)>S(n-t_2,k-2)$,

$$\sum_{t=h_1}^{h_2}\lceil n/t\rceil S(n-t,k-2)k(k-1)/2\le\lceil n/h_1\rceil(h_2-h_1)S(n-h,k-2),$$

而 $k-1$ 层联盟结构总数为 $S(n,k-1)$,所以,

$$NS(k-1)/S(n,k-1)\le\frac{\lceil h/h_1\rceil(h_2-h_1)S(n-h_1,k-2)k(k-1)/2}{S(n,k-1)},$$

$$\begin{aligned}S(n,k-1)&=(k-1)S(n-1,k-1)+S(n-1,k-2)\ge(k-1)S(n-1,k-1)\\&\ge(k-1)^2S(n-2,k-1)\\&\ge\\&\cdots\\&\ge(k-1)^{h_1-1}S(n-h_1+1,k-1)\\&\ge(k-1)^{h_1-1}S(n-h_1,k-2).\end{aligned}$$

因此,

$$\begin{aligned}NS(k-1)/S(n,k-1)&\le\frac{\lceil h/h_1\rceil(h_2-h_1)S(n-h_1,k-2)k(k-1)/2}{(k-1)^{h_1-1}S(n-h_1,k-2)}\\&=\frac{\lceil n/h_1\rceil(h_2-h_1)k(k-1)}{2(k-1)^{h_1-1}},\end{aligned}$$

$$h_2-h_1=(k-2)\lceil n/k\rceil,\ \lceil n/h_1\rceil=\lceil n/n-k+2\rceil<1.$$

因此,$NS(k-1)/S(n,k-1)<\frac{(k-2)\lceil n/k\rceil k}{2(k-1)^{n-k+1}}\le\frac{k\lceil n/k\rceil}{2(k-1)^{n-k}}\le\frac{n+k}{2(k-1)^{n-k}}\le\frac{n}{(k-1)^{n-k}}$.

故与完全搜索算法相比,SCS 算法在第 $k-1$ 层的搜索空间大小是其 $\frac{n}{(k-1)^{n-k}}$.

4 与已有工作的比较和实验分析

4.1 与已有工作的比较

对于 Agent 联盟结构搜索算法,可以从是否任意时间算法、搜索方式以及对第 L_k 层的搜索量等方面加以比较.任意时间算法是指算法可以在任意时间终止,返回一个结果.任意时间算法相对于非任意时间算法,具有更好的实际应用.搜索方式可以有自上而下的宽度优先方式,以及按其他规律的搜索方式,如文献[1].在 Agent 联盟搜索算法中,自上而下搜索方式的最大好处是,如果算法是任意时间算法,可以在算法终止时,返回本层及其上层的所有联盟结构中最优的联盟结构.对第 L_k 层的搜索量反映了算法的效率.

Sandholm 等人的算法可以看作是任意时间算法.Sandholm 等人采用对联盟结构图首先搜索 L_1,L_2,L_n 层,然后再执行自上往下的搜索方法.Sandholm 等人的结论表明:如果搜索完 L_1,L_2 和 L_n 层,可得到收益不低于最优收益 $1/[n/2]$的联盟结构;但是,如果要得到最优的联盟结构,仍需要搜索完所有可能的 Agent 联盟结构.Jennings 等人则采用对 $SL(n,[n(q-1)/q])$(所有最大联盟规模不大于$[n(q-1)/q]$的联盟结构)逐步搜索的方法(其中,q 从$[(n+1)/4]$到 2),每执行一次对 $SL(n,[n(q-1)/q])$的搜索,可得到收益不小于最大收益 $1/(2q-1)$的联盟结构.但是,如果想找到最优收益,Jennings 等人的方法也是需要搜索完全部可能的 Agent 联盟结构.因此,Sandholm 和 Jennings 等人的算法可以针对任意的 Agent 联盟收益函数,但是,为了搜索到最优联盟结构,需要对所有的联盟结构遍历一遍.

本文的 SCS 算法主要针对 Agent 收益函数满足合作收益独立性的前提下,可以对 Agent 联盟结构图进行剪枝.在 Agent 联盟结构图中,当 L_k 处于中间时,具有同构关系的 Agent 联盟结构数目比较大.如果对这些同构的 Agent 联盟结构按收益大小排序,可以将收益不是最大的剪枝去掉,因此减少了搜索量.与 Sandholm 和 Jennings 等人的工作相比,SCS 算法可以看作是任意时间算法,但其 Agent 联盟结构图中每层的数量降为原来的 $\frac{n}{(k-1)^{n-k}}$ 倍,可以显著减少搜索量.几种算法的比较见表 1.

Table 1 Comparison of search algorithms of Agent coalition

表 1 几种 Agent 联盟搜索算法的比较

	Anytime algorithm	Search method	The search steps of L_k
Sandholm's algorithm	Yes	Top-Bottom	$S(n,k)$
Jennings's algorithm	No	Horizontal	$S(n,k)$
SCS algorithm	Yes	Bottom-Top	$\frac{n}{(k-1)^{n-k}}S(n,k)$

4.2 实验分析

实验以机器人足球赛 RoboCup 为背景,相邻位置的球员可以互相合作,如传球、合作过人等.球员在球场上可以分为多个联盟,如后卫之间合作、后卫与中场之间的合作.这种合作可以看作是一种联盟.因此,采用 RoboCup 为实验背景,可以很好地说明 Agent 联盟在 Agent 合作中的作用.

两边球队分为红队、兰队,每只球队由 10 名队员组成(无守门员).红队=$\{a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8,a_9,a_{10}\}$,兰队=$\{b_1,b_2,b_3,b_4,b_5,b_6,b_7,b_8,b_9,b_{10}\}$.$a_i,b_i$ 表示第 i 号队员.队员的属性用 4 元组$\langle a_i,pos,role,ability\rangle$表示,其中:$a_i$ 表示队员 id;pos 表示队员在场上的位置;$role$ 表示队员的身份,$role$={前锋=1,中场=2,后卫=3};$ability$ 表示队员的能力,包括跑动能力、截球能力、传球能力、射门能力.每个球队中的 Agent 可以组成几个联盟,在比赛过程中,随着 Agent 位置的不同,Agent 的联盟可以动态变化.联盟内的 Agent 采用预先定义的合作策略进行动作.不同联盟内的 Agent 不合作.所有的 Agent 划分成几个联盟可以提高合作效率,因为同一个联盟内的 Agent 如果数量太

大,Agent 合作策略必然复杂.

实验中,Agent 联盟收益函数定义为

$$U(C)=\frac{k_1}{\sum_{i\neq j}|\,pos(i)-pos(j)\,|}+\sum_{i\neq j}k_2\,f(role(i),role(j)),\ i,j\in C,$$

其中:k_1,k_2 是两个常数;$f(role(i),role(j))$ 是角色匹配函数,如果 $role(i)=role(j)$,则 $f(role(i),role(j))=2$;如果 $role(i)-role(j)=1$,则 $f(role(i),role(j))=1$;否则,$f(role(i),role(j))=0$; $U(C)$表示考虑了球员在场上的位置以及他们所持有的身份:位置越近,则收益越大,同一类球员之间合作收益越大.实验中,取 $k_1=20,k_2=5$.对于联盟 C_1,C_2 当 C_3,在 C_3 中的球员位置和 C_1,C_2 中的球员位置较远时,即

$$\frac{k_1}{\sum_{i\neq j}|\,pos(i)-pos(j)\,|}\approx\frac{k_1}{\sum_{i\neq k}|\,pos(i)-pos(p)\,|}\quad i\in C_3,\ j\in C_1,\ p\in C_2,$$

则 C_3 对 C_1 和 C_2 的影响可以认为是相当的.因此在比赛中,随着球员位置的变化,收益函数在某些时刻是满足合作收益独立性的.

4.3 实验结果

(a) 实验中分别采用 Sandholm 和 SCS 算法,计算需要搜索的 Agent 联盟结构数,如图 2 所示(x 轴是 L_k,对应每层 Agent 联盟结构图,y 轴是需要搜索的联盟结构数).

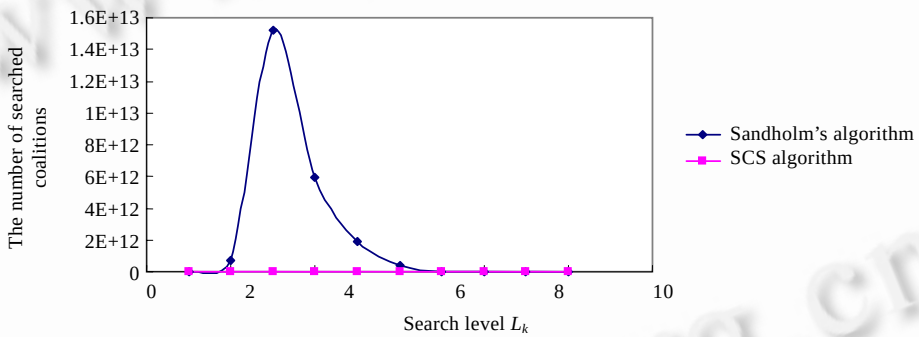

Fig.2 Comparison of SCS and Sandholm's algorithms

图 2 SCS 和 Sandholm 等人算法比较

L_k 在从 20 到 1 的变化过程中,开始时,Agent 联盟结构中具有同构关系的 Agent 联盟结构较少,因此,SCS 算法与 Sandholm 等人算法相差不大.但是在 L_k 变化到 3~7 时,每一层的 Agent 联盟结构中,具有同构关系的 Agent 联盟结构非常多,因此,经过 SCS 算法剪枝以后,可以显著减少需要搜索的 Agent 联盟结构个数.

(b) 实验中,同一联盟内的球员采用合作策略,不同联盟内的球员采用一般的非合作的策略.联盟越复杂,则合作策略越复杂.在实验中,红队采用 SCS 算法,兰队采用一般的自上而下的搜索算法.每隔 $T=20$ 个时间间隔,采用任意时间 Agent 联盟动态调整一次,调整时间设定为 $T=80$ 个时间间隔.两队比赛 10 次,比分数据见表 2.

Table 2 The score of red team and blue team in RoboCup

表 2 RoboCup 中红队和兰队的比分

Match	1	2	3	4	5	6	7	8	9	10
Red team	8	7	9	8	7	9	8	6	7	7
Blue team	2	4	3	3	5	2	3	1	2	4

红队采用 SCS 算法,其搜索效率较高,因此在同样的时间内,其搜索的联盟收益大于兰队的联盟收益,亦即,其联盟内的合作效率高于兰队联盟内的合作收益.从实验结果也可以看出:在收益函数满足合作收益独立性的前提下,采用 SCS 算法的红队成绩优于采用一般自上而下的兰队.

5 结 语

在多 Agent 合作求解过程中,Agent 联盟是一种重要的合作求解形式.但 Agent 联盟结构的个数,随着 Agent 数目的增加而增长很快.如果不对 Agent 联盟结构图进行化简而直接搜索,则效率比较低.本文给出了在 Agent 联盟收益满足合作收益独立性的性质下,对 Agent 联盟结构图进行剪枝,可以显著提高搜索效率,其每层搜索量是剪枝前的 $\frac{n}{(k-1)^{n-k}}$.

在今后的工作中,可以考虑如何对不满足或者部分满足合作收益独立性的 MAS 进行 Agent 联盟结构图化简;也可以结合 Agent 理性,考虑 Agent 个体收益与群体收益对 Agent 联盟形成的影响等问题.

References:

[1] Jennings NR, Dang VD. Generation coalition structures with finite bound from optimal guarantees. In: Proc. of the AAMAS 2004. vol.2. 2004. 572−579. http://ieeexplore.ieee.org/iel5/9442/29991/01373523.pdf

[2] Anderson J, Tanner B, Baltes J. Dynamic coalition formation in robotic soccer. In: Proc. of the AAAI 2004. 2004. 1−11. http://brian.tannerpages.com/Publications/44E1CC68-DAA1-440F-910F-8D138192E7BF.html

[3] Klusch M, Blankenburg B. On safe kernel stable coalition formation among Agents. In: Proc. of the AAMAS 2004. vol.2. 2004. 580−587. http://ieeexplore.ieee.org/iel5/9442/29991/01373525.pdf

[4] Klush M, Gerber A. Dynamic coalition formation among rational Agents. IEEE Intelligent Systems, 2002,17(3):42−47.

[5] Konishi H. Coalition formation as a dynamic process. Journal of Economic Theory, 2003,110:1−41.

[6] Tsvetovat M, Sycara K. Customer coalitions in electronic marketplace. In: Proc. of the 4th Int'l Conf. on Autonomous Agents. 2000. 263−264. http:// www.cs.cmu.edu/~softagents/papers/agents2k.ps.gz

[7] Sandholm T, Larson K, Andersson M, *et al*. Coalition structure generation with worst case guarantees. Artificial Intelligence, 1999, 111(1-2):209−238.

[8] Dang TT, Frankovie B, Budinnska I. Create Agent's coalition based on a dynamic programming approach. In: Proc. of the 15th European Conf. on Artificial Intelligence ECAI 2002, Workshop "Agent Technologies and Logistics". 2002. 16−24. http://www.ui.savba.sk/mas/publications/ECAI2002-workshop-last.pdf

张新良(1978－),男,山东高密人,博士,主要研究领域为分布式人工智能,多 Agent 系统.

石纯一(1935－),男,教授,博士生导师,CCF 高级会员,主要研究领域为人工智能应用基础.