一种求解单任务 Agent 联盟生成的近似算法

李 勇,蒋建国,夏 娜
LI Yong,JIANG Jian-guo,XIA Na

合肥工业大学 计算机与信息学院,合肥 230009
Department of Computer and Information Science,Hefei University of Technology,Hefei 230009,China

LI Yong,JIANG Jian-guo,XIA Na.Searching for Agent coalition for single task using approximation algorithm.Computer Engineering and Applications,2008,44(2):29-31.

Abstract: Coalition is an important cooperative method in Multi-Agent System (MAS).It is a complicated combinatorial optimization problem to search for the optimal,task-oriented Agent coalition.The problem's character is that coalitions including more Agents are better than those including few Agents.So,an approximation algorithm is proposed according to the character.The results of contrastive experiment show that this algorithm is fast and effective.
Key words: Multi-Agent System(MAS);coalition;approximation algorithm

摘 要:联盟是多 Agent 之间一种重要的合作方法,如何生成面向某个任务的最优联盟是一个复杂的组合优化问题。本问题的特点是:包含较少 Agent 的联盟要优于包含较多 Agent 的联盟。根据此特点提出一种近似算法,比较实验结果表明本算法快速、有效。
关键词:多 Agent 系统;联盟;近似算法
文章编号:1002-8331(2008)02-0029-03 **文献标识码**:A **中图分类号**:TP18

1 引言

联盟是多 Agent 系统(Multi-Agent System,MAS)的重要合作方法。Agent 间通过组成联盟,可以解决单个 Agent 无法解决的问题,获得更多的效益。自 1993 年提出联盟方法以来,联盟已成为多 Agent 系统研究的一个重要方面,并取得了相当的进展。

1.1 问题的描述

设 Agent 集 $N=\{A_1,A_2,\cdots,A_n\}$,任意 A_i 都有一个能力向量:$\boldsymbol{B}_i=\langle b_i^1,b_i^2,\cdots,b_i^r\rangle,b_i^j\geqslant 0,(1\leqslant i\leqslant n,1\leqslant j\leqslant r)$,用于定量描述 A_i 执行某种特定动作的能力大小。任务 t 具有一定的能力需求 $\boldsymbol{B}_t=\langle b_t^1,b_t^2,\cdots,b_t^r\rangle$。Agent 完成任务 t 可获得相应的利益 $P(t)$。

定义一个联盟 C 是 N 的一个非空子集。C 有一个能力向量 $\boldsymbol{B}_C=\langle b_C^1,b_C^2,\cdots,b_C^r\rangle$,$\boldsymbol{B}_C$ 是 C 中所有 Agent 能力向量的总和,即 $\boldsymbol{B}_C=\sum_{A_i\in C}B_i$。联盟 C 可以完成任务 t 的必要条件是:$\forall 1\leqslant i\leqslant r,b_t^i\leqslant b_C^i$。

联盟值的计算模型简述如下:与文献[1-4]相同,在特征函数对策中研究联盟形成。每个联盟 C 的联盟值用一个特征函数 $V(C)$给出:$V(C)=P(t)-F(C)-E(C)$。式中 $P(t)$指完成任务 t 所获得的利益,对于具体任务是一个常数;$F(C)$指联盟成员总能力折合的成本;$E(C)$指联盟协作求解 t 过程中的额外开销,主要指通信开销,即联盟中任意两个 Agent 之间的通信代价之和。如果联盟 C 不满足上述完成任务 t 的必要条件,则 $V(C)$为 0,否则 $V(C)$为正数。(2)在非超加性环境中研究联盟生成[4](超加性是指对任意 $C_1,C_2\subseteq N,C_1\cap C_2=\Phi$,有 $V(C_1\cup C_2)\geqslant V(C_1)+V(C_2)$,在这样的环境中,最大的联盟将是最有益的)。在计算 $F(C)$时,一般的情况是联盟中所有 Agent 各维能力加权求和,权系数称为成本因子,为简便起见,本文将各成本因子取为 1。在下文中,$F(C)$称为能力成本,$E(C)$称为通信成本。

单任务 Agent 联盟生成问题就是面向任务 t 寻找最优的联盟 $C(C\subseteq N)$,使联盟值 $V(C)$最大。由于在 MAS 中,可能的联盟总数同 MAS 中 Agent 的数目成指数关系[1],为了得到一个满意的结果必须考虑所有或大部分的联盟组合可能,因而该问题是一个复杂的组合优化问题。

1.2 相关的工作分析

目前在 MAS 中,联盟形成的基本理论是 N 人合作对策论理论,如 Shapley 值、核、核心等。Zoltkin & Rosenschein[2]、Ketchple[3]的工作具有代表性。N 人合作对策主要考虑如何划分联盟值,检查划分的稳定性和公平性,使 Agent 在决策时愿意形成全局最优联盟;它没有考虑算法,只考虑解的存在性,也不考虑计算资源、通信开销和计算分布等要求。N 人合作对策为 MAS 中联盟形成提供了一定的指导,近来较新的研究有文献[7,8]。

随着 MAS 的发展,研究人员从计算可实现的角度研究联

基金项目:国家自然科学基金(the National Natural Science Foundation of China under Grant No.60474035);国家教育部博士点基金(No.20060359004);安徽省自然科学基金(the Natural Science Foundation of Anhui Province of China under Grant No.070412035)。
作者简介:李勇(1970-),男,博士生,主要研究方向为分布式人工智能、进化计算等;蒋建国(1955-),男,教授,博士生导师,主要研究方向为信号与信息处理、传感与智能控制等;夏娜(1979-),男,副教授,博士,主要研究方向为分布式人工智能、智能控制和进化计算等。

盟生成,并提出了一些相关算法。Sandholm[4,9]、胡山立[10]徐晋晖[11]基于“联盟结构”求解多任务的 Agent 联盟。假设系统中由多个任务需求解,用 Agent 集合的一个划分(称为一个“联盟结构”)对应多个任务,每个分块对应一个求解某任务的联盟,即任一 Agent 至少参加且仅参加一个联盟。联盟结构的搜索就是要找出使所有联盟的联盟值之和为最大的联盟结构。因为可能的联盟结构与系统中的 Agent 数量也是成指数关系,因而如何提高求解效率成了研究的重点,实际上,联盟结构的搜索算法是当前联盟问题的研究热点,关于此方面的研究很多,在此不再列举。

但是系统中并不是任何时候都有多个任务需要执行;并且也可能有多个任务需要串行执行;或者某个任务需要优先排他执行;或者根据联盟结构原则,每个 Agent 只参加一个联盟,系统中 Agent 的能力不足以支持两个或更多任务的并行执行。因而研究单任务的 Agent 联盟的生成仍有其意义[6,12-14]。在相关的单任务联盟生成研究中,主要是基于遗传算法[13]、蚁群算法[6]、差异演化算法(Differential Evolution,DE)[14],粒子群算法[15,16]等智能算法求解该问题。DE 是一种基于群体差异的演化算法,该算法是 Storn 在 1996 年为求解切比雪夫不等式而提出的,和其他智能算法相比,DE 在求解非凸、多峰、非线性函数优化问题中表现出极强的稳健性,且在同样的精度要求下算法收敛的速度快[14],但算法的不足是对存储空间的要求大,群体规模一般在染色体编码长度的 5 到 10 倍之间, 具体算法步骤请参见文献[14]。文献[15]提出一种自适应的粒子群算法,提高了求解效率。文献[16]提出一种带有自适应扰动机制、采用基于信息正反馈孤岛模型的多种群合作搜索的粒子群算法,较好地避免了未成熟收敛的问题。

智能算法是常见的求解此类问题的算法,但是它们是一种随机化的算法,并且对参数设定敏感,但参数又很难精确事先确定,只能根据经验设定。

本文根据联盟生成问题的特点,提出一种不依赖参数的近似算法。

2 算法

2.1 联盟生成问题的特点

单任务 Agent 联盟生成问题就是在含有 n 个 Agent 的 MAS 中,寻找能够完成任务 t 且联盟值最大的 Agent 联盟。

考察任意一个满足必要条件可以完成任务的联盟 C,如果增加一个 Agent 加入 C,联盟变成 C',显然 C'也可以完成任务。但 C'的能力成本和通信成本都比 C 增大了, 因而 C'的联盟值比 C 减小了。如果减去 C 中的某个 Agent,联盟变成 C'',假定 C''仍然满足必要条件可以完成任务, 则 C''的能力成本和通信成本都比 C 减小了,因而 C''的联盟值比 C 增大了。

结论:对同一个联盟而言,在保证满足必要条件以完成任务的前提下,减少联盟中的 Agent 数量可以增大其联盟值。这是联盟生成的特点。

可以看出,能力成本的增加或减少与联盟中 Agent 的个数是线性关系,而通信成本的增加或减少与 Agent 的个数是非线性关系, 即通信成本对 Agent 的个数更敏感。由于在大规模 MAS 中,通信开销的代价远高于计算处理的代价[17],因此,一般说来,Agent 个数少的联盟的联盟值也要优于个数多的联盟。文献[18]也类似地指出:当单个 Agent 无法完成任务时,由 MAS 中的多个 Agent 通过联盟机制协作完成,并要求参与完成任务的 Agent 数量尽量少,每个 Agent 的能力浪费尽量少。

因此,本文在保证联盟可以完成任务的前提下,以尽量减少联盟内 Agent 个数为原则设计一种近似算法。

2.2 下界的定义及算法思想

将系统中所有 Agent 根据各自能力向量第一维分量 b_i^1 ($1\leqslant i\leqslant n$)的大小,从大到小进行排序,得到 Agent 序列:A_{11}, $A_{12},\cdots,A_{1n}$,且 $b_{11}^1\geqslant b_{12}^1\geqslant\cdots\geqslant b_{1n}^1$。将前 n_1 个 Agent 的第一维分量累加,累加值为 $b^1(n_1)=b_{11}^1+b_{12}^1+\cdots+b_{1n_1}^1$,求出满足 $b^1(n_1)\geqslant b_t^1$(任务 t 的第一维能力需求)的最小值 n_1,可见,任何 Agent 数量小于 n_1 的联盟,由于其第一维能力分量小于 b_t^1,将不能完成任务 t。那么,n_1 就是可以达到任务 t 的第一维能力需求,联盟所必须含有的最少 Agent 数目。同理可以得到 $n_2,n_3,\cdots,n_r$。

令 $low\_edge=n_k=\max\{n_1,n_2,\cdots,n_r\}$,可见,任何 Agent 个数小于 low_edge 的联盟,至少有一维能力不能满足任务需求,因此将不能完成任务。所以 low_edge 是联盟可以完成任务所必须包含的最少的 Agent 数量,是一种必要条件。遗憾的是,不能保证它是充分条件。low_edge 本文称为“下界”。

任务的不同维(分量)对于 Agent 的个数需求是不同的,从下界的定义看出,其对应的就是对 Agent 的个数需求最大的分量,即第 k 维分量。从减少联盟中 Agent 个数的目的出发,必须优先满足需求大的分量才可能减少 Agent 的个数。

算法的思想是,挑选能力大的 Agent 优先满足需求大的任务分量,形成一个初始联盟。然后对初始联盟在满足任务能力需求的条件下,进行联盟中 Agent 数量的“压缩”。最后,对联盟做局部优化:逐个检查联盟中的 Agent 能否被剩余的 Agent 替换,以得到更优的联盟。

2.3 算法描述

具体算法如下:

(1)初始联盟的生成:初始联盟为空,将所有 Agent 按照下界对应的第 k 维能力的大小从大到小排序,将该 Agent 序列的前下界个 Agent 加入初始联盟,此时,第 k 维任务能力需求被满足。求出任务能力需求分量与此时的联盟能力向量差值,求出最大差值的分量,将剩余 Agent 中此维分量最大的 Agent 加入联盟。继续求解任务需求向量与联盟能力向量的差值以及差值最大分量,挑选剩余 Agent 中此维分量最大的 Agent 加入联盟。不断重复上述过程,直到任务需求向量被满足。这样就得到了一个初始联盟。

(2)联盟的“压缩”:在初始联盟的形成中,最后被满足的任务需求向量的某维分量的实现是“低效率的”,因此,可进行“压缩”。

设最后满足的是任务需求向量的第 j 维,将初始联盟中的 Agent 根据其第 j 维能力分量从小到大排序, 剩余 Agent 也根据其第 j 维能力分量从大到小排序。按照上述排序遍历初始联盟中所有两个 Agent 的组合,对每个组合,按剩余 Agent 排好的顺序检查每个剩余 Agent 能否替换该组合,即剩余 Agent 代替该组合后,初始联盟仍能满足任务能力需求。如果能替换,则将替换后形成的新联盟和更新后的剩余 Agent 仍按上述方法排序,再检查新联盟中的组合能否被替换,直到所有组合都无法替换。将 Agent 排序是为了尽可能早地发现可以替换的 Agent 组合。

这样经过替换后,联盟中 Agent 个数就会相应减少,即实现了"压缩"。

理论上,可以再对初始联盟中倒数第 2,3……等次序被满足的分量执行同样的"压缩"操作,但笔者进行的大量实验中发现其作用很有限,因此就不再对这些分量做"压缩"操作。

(3)联盟的局部优化:仍然在满足任务需求的条件下,对联盟中的每一个 Agent, 检查所有剩余 Agent 中能否有 Agent 可以替换该 Agent,即替换后的联盟仍然满足任务的能力需求,并且新的联盟值大于原联盟值。显然,该局部优化不会减少联盟中的 Agent 数量。完成该操作后,可以实现联盟值的进一步优化。

上述描述已经十分清晰, 并且算法的伪代码描述过于繁琐,因此就不再给出。

2.4　算法复杂度分析

算法中计算量最大的是"压缩"部分。设初始联盟中 Agent 的个数是 $m(low\_edge \leqslant m \leqslant n)$,在最恶劣的情况,从联盟中每"压缩"一个 Agent 的最大代价是 $O(m\cdot(m-1)\cdot n\log n)$,最多减少 $m-low\_edge$ 个 Agent,一般远小于系统中 Agent 的数量 n,而 m 的数值可能接近 n,因此"压缩"部分和整个算法的复杂度上界是 $O(n^3\log n)$。

3　实验

实验取三组数据。第一组数据:系统中有 30 个 Agent,6 维能力向量,Agent 任一能力向量分量是在 0 与 100 之间随机生成的一个整数; 任意两个 Agent 之间的通信成本是在 0 与 200 之间随机生成的整数。第二组是 60 个 Agent,8 维能力向量,第三组是 100 个 Agent,8 维能力向量。Agent 数据均类似第一组随机生成。每组均随机生成 10 个任务。

在大量的实验中发现在相同的搜索资源下(相同种群个体数、迭代次数)文献[16]的搜索结果要相对优于其他智能算法(遗传算法[13]、蚁群算法[6]、DE[14]以及文献[15]的粒子群算法),因此,仅列出文献[16]与本文近似算法的比较。

文献[16]的参数如下:四个子种群,每个子种群有 20 个粒子(联盟),迭代 10 000 次。每个任务重复运行 5 次,取 5 次运行的最优值。下表中 $V(c1)$表示近似算法的初始联盟值,$K1$ 表示联盟中 Agent 数量,$V(c2)$表示近似算法最后获得的联盟值,$K2$ 表示其中的 Agent 数量,$V(c3)$是文献[16]算法的得到的联盟值,$K3$ 是 Agent 数量。第二组的实验结果对比如表 1。

表 1　对比实验结果

task	$V(c1)$	$K1$	$V(c2)$	$K2$	$V(c3)$	$K3$
t1	43 483	47	52 490	44	50 739	45
t2	20 185	55	24 226	54	24 226	54
t3	73 794	35	78 783	33	78 234	33
t4	66 744	38	66 744	38	66 357	39
t5	61 040	40	68 404	37	68 877	38
t6	92 706	25	97 944	23	97 626	24
t7	27 884	53	30 946	52	30 846	52
t8	33 214	51	36 293	50	37 082	50
t9	32 566	51	43 370	47	39 591	49
t10	31 104	52	31 104	52	31 104	52

在实验中发现,在迭代 10 000 次后,粒子群算法基本已失去进一步的搜索能力。从表 1 看出,近似算法的解的质量令人满意,而实验中发现计算时间远远小于粒子群算法。第一组的实验结果与第二组十分相似,而第三组的实验中,近似算法的解的质量有所下降,不能令人满意,这说明,对于较大规模的问题,近似算法不是很适合,但可以将近似算法与各种局部搜索算法结合起来,或者把近似算法的解及其变形作为智能算法的初始种群,以提高其搜索的效率,具体就不再赘述。

4　结论

本文根据单任务 Agent 联盟生成问题的自身特点,针对性地提出了一种近似算法。从实验中发现总体求解结果在问题规模不大时求解质量令人满意,且本算法所用时间要远远优于粒子群算法(约为后者的几十分之一),因而更适合对实时性要求高的场合。实验结果证明,本文的算法简单、有效。

(收稿日期:2007 年 9 月)

参考文献:

[1] Shehory O,Kraus S.Task allocation via coalition formation among autonomous agents[C]//Proc of IJCAI-95,August 1995:655-661.

[2] Zoltkin G,Rosenschein J S.Coalition,cryptography,and stability:mechanisms for coalition formation in task oriented domains[C]//Proc of AAAI-94,Seattle,US,1994:432-437.

[3] Ketchple S.Forming coalitions in the face of uncertain rewards[C]//Proc of AAAI-94,Seattle,US,1994:414-419.

[4] Sandholm T W,Lesser V R.Coalition among computationally bounded agents[J].Artificial Intelligence,1997,94(1):99-137.

[5] 兰少华,叶东海,吴慧中.一种 Agent 任务求解联盟形成策略[J].小型微型计算机系统,2004(5).

[6] 夏娜,蒋建国,魏星,等.改进型蚁群算法求解单任务 Agent 联盟[J].计算机研究与发展,2005,42(5):734-739.

[7] Blankenburg,B,Dash R K,Ramchurn S D,et al.Trusted kernel-based coalition formation[C]//Proceedings of the 4th International Conference on Autonomous Agents and Multi-agent Systems,AAMAS 05,2005:989-996.

[8] Cao Yuan-da,Li Jian.Algorithm to form coalition in multi-agent cooperation[J].Journal of Beijing Institute of Technology:English Edition,2005,14(2):117-120.

[9] Sandholm T W,Larson K,Andersson M R,et al.Anytime coalition structure generation with worst case guarantees[C]//Proc of the National Conference on Artificial Intelligence,Madison,WI,July 1998:46-53.

[10] 胡山立,石纯一.一种任意时间联盟结构生成算法[J].软件学报,2001,12(5):729-734.

[11] 徐晋晖,张伟,石纯一,等.面向结构的 Agent 组织形成和演化机制[J].计算机研究与发展,2001,38(8):897-903.

[12] 骆正虎.移动 Agent 系统若干关键技术问题研究[D].合肥:合肥工业大学研究生院,2002:81-98.

[13] 郑金华,陈振洲,蔡自兴.用遗传算法实现多智能体联盟的形成[J].计算机工程与科学,2004,26(6).

[14] 武志峰,黄厚宽.用差异演化算法求解单任务 Agent 联盟[J].计算机研究与发展,2006,43(Suppl.):186-189.

[15] 蒋建国,吴琼,夏娜.自适应粒子群算法求解 Agent 联盟[J].智能系统学报,2007(4).

[16] 张国富,蒋建国,夏娜,等.基于离散粒子群算法求解复杂联盟生成问题[J].电子学报,2007(2).

[17] 陈刚,陆汝钤.关系网模型-基于社会合作机制的多 Agent 协作组织方法[J].计算机研究与发展,2003(1).

[18] 叶斌,马忠贵,曾广平,等.多 agent 系统的联盟框架及形成机制[J].北京理工大学学报,2005(10).