

自适应粒子群算法求解 Agent 联盟

蒋建国, 吴琼, 夏娜

(合肥工业大学 计算机与信息学院, 安徽 合肥 230009)

摘要: 联盟生成是多 Agent 系统的一个关键问题, 主要研究如何在多 Agent 系统中动态生成面向任务的最优 Agent 联盟. 引入粒子群算法来解决这一问题, 受到惯性权重 c_0 在进化过程中所起作用的启发, 引入自适应惯性权重 c_{adapt} 对粒子群算法进行改进, 使其不再易于陷入局部极小. 对比实验结果表明, 该算法在解的性能和收敛速度上均优于相关算法.

关键词: 多 Agent 系统; 粒子群优化算法; 自适应粒子群算法; 联盟

中图分类号: TP18 **文献标识码:** A **文章编号:** 1673-4785(2007)02-0069-05

Solving Agent coalition using adaptive particle swarm optimization algorithm

JIANG Jian-guo, WU Qiong, XIA Na

(College of Computer and Information, Hefei University of Technology, Hefei 230009, China)

Abstract: Coalition Generation is a key issue in a Multi-Agent System which primarily focuses on generation of an optimal task-oriented coalition in a dynamic manner. A particle swarm optimization (PSO) algorithm is adopted to solve the problem. And a novel "adaptive inertia weight" is proposed to improve PSO by the illumination of function of inertia weight, so as to avoid falling into local minimum. The results of comparison experiments show that this algorithm is superior to other related methods in both performance of solution and convergence rate.

Key words: MAS; PSO algorithm; adaptive particle swarm optimization; coalition

在大规模复杂的分布式智能控制系统中, 多个子系统之间的协调、合作尤为重要. 基于智能 Agent 的 MAS 系统将是解决这一难题的重要途径. 在由自治 Agent 构成的多 Agent 系统 (MAS) 中, Agent 之间经常需要协作来完成任务求解. 假设系统中有一个 Agent 集合和一组待求解任务, 由于单个 Agent 无法独立完成某一任务或者通过多个 Agent 协作能提高求解效率, Agent 之间可以通过协商形成 Agent 组——称之为联盟 (coalition) 来共同承担该任务. 自 Shehory^[1] 和 Sandholm^[2-3] 提出联盟方法以来, 已取得了一定的研究进展. 通过联盟可以提高 Agent 求解问题的能力, 获得更多的报酬, 因而联盟是多 Agent 系统 (MAS) 中的重要合作方法, 已经有很多学者在 Agent 联盟这一领域进行了大量

研究, 包括联盟生成^[4-5]、成员间协商^[6] 以及联盟内收益分配^[7-8] 等方面, 其中联盟生成问题是本论文的研究重点.

1 联盟生成问题

1.1 问题的描述

联盟生成问题可形式化描述如下:

设 Agent 集 $N = \{A_1, A_2, \dots, A_n\}$, 任意 A_i 都有一个能力向量: $A_i = \langle a_i^1, a_i^2, \dots, a_i^r \rangle$, $a_i^k \geq 0$, ($1 \leq i \leq n, 1 \leq k \leq r$) 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 $N(t) = \langle n^1, n^2, \dots, n^r \rangle$, $n^k \geq 0$, ($1 \leq k \leq r$), 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. 联盟 C 有一个能力向量 $C = \langle c^1, c^2, \dots, c^r \rangle$, C 是联盟中所有 Agent 能力向量的总和, 即 $C = \sum_{A_i \in C} A_i$. 联盟 C

可以完成任务 t 的必要条件是:

$$\forall i, k, r, n^k, c^k.$$

作以下假设:(1)和惯例一样^[2-3,9],在特征函数对策(character function games, CFGs)中研究联盟形成.每个联盟 C 的值用一个特征函数 $V(C)$ 给出.假定 $V(C) \geq 0, V(C) = P(t) - F(C) - C(C)$. 式中 $P(t)$ 指完成任务所获得的利益; $F(C)$ 指联盟成员总能力折合的成本; $C(C)$ 指联盟协作求解 t 过程中的额外开销,主要是通信开销.如果联盟 C 不满足上述必要条件,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形成^[3](超加性是指 $\forall C_1, C_2 \subseteq N, C_1 \cap C_2 = \emptyset$, 有 $V(C_1 \cup C_2) \geq V(C_1) + V(C_2)$),在这样的环境中,最大的联盟将是最有益的).

Agent 联盟生成问题就是对于任务 t 寻找最优的联盟 $C(C \subseteq N)$,使得联盟值 $V(C)$ 尽可能大.由于在 MAS 中,不同 Agent 之间几乎都存在彼此合作形成联盟的可能,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形式,因此联盟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,如 Shapley 值、核、核心等. 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.

随着 MAS 的发展,研究人员从计算可实现的角度研究联盟生成,并提出了一些相关算法. Sandholm^[2-3]、胡山立^[10]、徐晋晖^[11] 基于“联盟机构”求解多任务的 Agent 联盟,联盟的生成过程就是在联盟结构图中进行搜索.这些算法大多通过约束条件来裁减搜索树,减少搜索空间,在降低了算法复杂度的同时也牺牲了对最优解的求解,往往得到一个与最优解相距在一个界限内的次优解. Sen^[12]、骆正虎^[13] 基于遗传算法求解 Agent 联盟,在可接受的时间内求解的质量有所提高.夏娜^[5] 等引入改进后的蚁群算法来求解单任务的 Agent 联盟,加快了收敛速度;还有很多学者在 MAS 联盟的应用方面做了大量工作, MAS 联盟已经被成功应用于项目管理^[14]、虚拟企业创建^[15]、智能诊断^[16] 和智能控制^[17] 等领域.

本文在参考这些成果的基础上,引入粒子群算法来求解单任务的 Agent 联盟.

2 算 法

2.1 粒子群算法

粒子群算法 (PSO) 是由 Kennedy 和 Eberhart^[18-19] 等于 1995 年提出的一种进化计算算法,其基本思想来源于对鸟群觅食过程的研究及行为模拟.群体中的鸟被抽象为理想“粒子”,这些“粒子”的运动受到自身速度、自身和群体的历史最佳位置信息的影响,能够在复杂的解空间寻找最优解. PSO 算法概念简明、实现方便,因此自从提出以来便在短期内迅速得到了国际进化计算研究领域的认可,并在算法实现模式及应用领域得到了很大的发展.

PSO 初始化为一群随机粒子(随机解),然后通过迭代找到最优解.在每次迭代中,每个粒子根据以下公式更新自己的速度和新的位置:

$$v_{k+1} = \omega v_k + c_1 (pbest_k - x_k) + c_2 (gbest_k - x_k). \tag{1}$$

$$x_{k+1} = x_k + v_{k+1}. \tag{2}$$

式中: $pbest_k$ 表示粒子本身找到的最优解的位置; $gbest_k$ 表示整个种群目前找到的最优解的位置; v_k 是粒子的速度向量; x_k 是当前粒子的位置; ω, c_1 和 c_2 是权重系数. v_{k+1} 是 $v_k, pbest_k - x_k$ 和 $gbest_k - x_k$ 矢量的加权和,其示意图如图 1^[20] 所示:

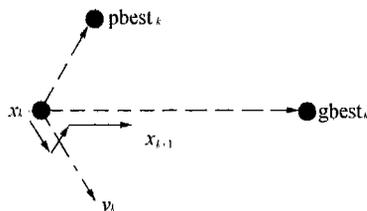


图 1 粒子运动矢量示意图

Fig. 1 Sketch map of particle's movement vector

2.2 基于基本 PSO 算法求解 Agent 联盟

Agent 联盟问题是在含有 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,属于组合优化问题.在 PSO 算法中每一个位置对应一个联盟,在单任务联盟中,由于 Agent 只能选择参加或者不参加联盟,因此解可以表示为一个 n 维矢量 $L_n = \langle l_0, l_1, \dots, l_{n-1} \rangle, l_i \in \{0, 1\} (1 \leq i \leq n)$, $l_i = 1$ 代表 $Agent_i$ 在 L 所对应的联盟里. PSO 算法的本质是利用个体极值和全局极值 2 个信息,来指导粒子下一步的迭代位置^[13].如按照基本 PSO 算法,其速度难以表达,故借鉴遗传算法的思想:式 (1) 中: ωv_k 项可以看作变异操作, $c_1 (pbest_k - x_k) + c_2 (gbest_k - x_k)$ 可以看作交叉操作,将当前解与个体

极值和全局极值分别作交叉操作.

算法描述如下:

1) 设定粒子数 m , 规定迭代次数 NC , 随机产生 m 个初始解 $L_n^m = \{L_n^0, L_n^1, \dots, L_n^{m-1}\}$;

2) 根据当前位置计算出每个粒子的适应值 V_i (即粒子对应联盟的联盟值 $V(C)$), 设置当前适应值为个体极值 V_{ipbest} , 当前位置为个体极值位置 L_n^{ipbest} , 找出其中最大的作为全局极值 V_{gbest} , 对应位置为 L_n^{gbest} ;

While(迭代次数 $< NC$) do

for ($i = 0; i < m; i++$)

3) 第 i 个粒子位置 L_n^i 与 L_n^{gbest} 交叉得到 L_n^{i1} ;

4) L_n^{i1} 与 L_n^{ipbest} 交叉得到 L_n^{i2} ;

5) 对 L_n^{i2} 进行变异操作得到 L_n^i ;

6) 对 L_n^i 计算适应值 V_i ;

7) 如果 $V_i > V_{ipbest}$, 则 $L_n^{ipbest} = L_n^i, V_{ipbest} = V_i$;

End

8) 根据个体极值 V_{ipbest} , 找出全局极值 V_{gbest} 及其位置 L_n^{gbest} ;

End

9) 输出全局极值 V_{gbest} 及其位置 L_n^{gbest}

此算法中交叉和变异操作花费时间最多, 里面循环 $3m$ 次, 外面循环 NC 次, 因此算法复杂度大约为 $O(3mNC)$.

2.3 算法的改进

PSO 算法有 3 个权重因子^[13]: 惯性权重 ω_0 、加速常数 c_1 和 c_2 . 惯性权重 ω_0 使粒子保持运动惯性, 使其有扩展搜索空间的趋势, 有能力探索新的区域, 具有全局搜索能力. 加速常数 c_1 使粒子具有认知能力, 在粒子的相互左右下, 有能力到达新的搜索空间. c_2 保证了粒子间的交互, 使得社会信息得到共享. 基本 PSO 算法在求解 Agent 联盟生成问题时没有考虑 ω_0 、 c_1 和 c_2 的关系, 研究发现较小的惯性权重 ω_0 有利于算法收敛, 而较大的 ω_0 有利于跳出局部极值. 为了提高算法的全局搜索能力和搜索速度, 引入自适应惯性权重 c_{adp} , 在出现局部极小情况时通过 c_{adp} 的作用使解跳出局部极小, 从而向最优解继续进化, 称这种能够自适应调整惯性权重 c_{adp} 的 PSO 算法为自适应粒子群算法 (APSO).

定义 1 $c_{adp} = \begin{cases} \omega_0, & \text{算法求解的最优解仍在进化,} \\ c_{adp} + 1, & \text{最优解在 } N \text{ 次循环内没,} \\ & \text{有明显改进, 且 } c_{adp} + 1 < c_{max}. \end{cases}$

可见, 在 APSO 系统运行初期, 算法求得的最优解仍在进化时, 自适应惯性权重 $c_{adp} = \omega_0$, 仍然运行基本 PSO 算法; 当算法求得的最优解在 N 次循

环内没有明显进化, 即可能陷入局部极小, c_{adp} 开始逐步增大, 使解更容易跳出局部极小; 另外, 为了保证算法的收敛速度, 引入 c_{max} 对 c_{adp} 进行强度控制. 一旦解跳出局部极小, 算法求得的最优解又开始进化时, $c_{adp} = \omega_0$, 返回成基本 PSO 算法.

此外, 区别于基本 PSO 算法, APSO 算法中分别随机选择 c_2 和 c_1 个位置进行相应的赋值, 而不是随机选择长度为 c_2 和 c_1 的段进行交叉, 保证了 3) 和 4) 操作的有效性; 随机选择 c_{adp} 个位置赋值 0 或者 1, 而不是变异, 这样不仅简单易于操作, 而且提高了了解的多样性. 改进后 3) ~ 5) 就成为简单的赋值, 花费时间远小于原本的交叉和变异. 算法复杂度约为 $O(mNC)$, 降低为原来的 $1/3$.

2.4 算法描述

应用 APSO 算法求解 Agent 联盟问题, 具体算法伪代码如下:

1) 设定粒子数 m , 规定迭代次数 NC , 随机产生 m 个初始解 $L_n^m = \{L_n^0, L_n^1, \dots, L_n^{m-1}\}$;

2) 根据当前位置计算出每个粒子的适应值 V_i (即粒子对应联盟的联盟值 $V(C)$), 设置当前适应值为个体极值 V_{ipbest} , 当前位置为个体极值位置 L_n^{ipbest} , 找出其中最大的作为全局极值 V_{gbest} , 对应位置为 L_n^{gbest} ;

While(迭代次数 $< NC$) do

for ($i = 0; i < m; i++$)

3) 对于粒子 i 随机从 L_n^{ipbest} 和 L_n^{gbest} 选择 c_1 和 c_2 个位置赋值给 L_n^i , 保证这些位置不重叠; 再从 L_n^i 中随机选择 c_{adp} 个位置随机赋值为 0 或者 1;

4) 对 L_n^i 计算适应值 V_i ;

5) 如果 $V_i > V_{ipbest}$, 则 $L_n^{ipbest} = L_n^i, V_{ipbest} = V_i$;

End

6) 根据个体极值 V_{ipbest} , 找出全局极值 V_{gbest} 及其位置 L_n^{gbest} ;

If (N 次循环 V_{gbest} 都没有改进)

then if ($c_{adp}++ < c_{max}$) then $c_{adp}++$

else $c_{adp} = c_{max}$

else $c_{adp} = \omega_0$

End

7) 输出全局极值 V_{gbest} 及其位置 L_n^{gbest}

3 实验验证

为了检验 APSO 算法性能进行了对比实验. 首先产生实验所需的参数: Agent 数 $n = 30$ 、每个 Agent 的能力向量 $A_i = \langle a_i^1, a_i^2, \dots, a_i^t \rangle$, Agent 之间的通信开销 d_{ij} 、任务 t 的能力需求 $N(t)$; 然后分别

应用文献[10]的改进型蚁群算法、基本 PSO 算法和本文提出的 APSO 算法求解该 Agent 联盟问题,并对所得结果进行了比较。下图 2 为最优解进化曲线,说明了 3 种算法求解的进化过程比较。

从图 2 可以看出,和前 2 种算法相比,APSO 算法不仅在解的质量上有所提高,而且收敛速度较快。

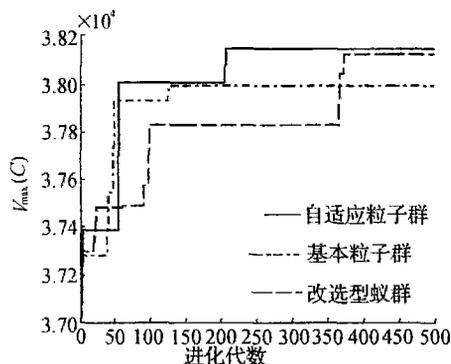


图 2 3 种算法求解过程最优解进化曲线

Fig. 2 The solution evolving curves of 3 algorithms

4 结束语

本文引入 APSO 算法来求解单任务 Agent 联盟问题,保持了 PSO 算法简单、并行的优点;并且在求解过程中 APSO 算法对惯性权重 ω 进行了自适应调整,使 V_{gbest} 更容易跳出局部极小,具有更强的鲁棒性。实验证明 APSO 算法有利于求解此类 Agent 联盟形成问题。下一步工作是研究多任务并行 Agent 联盟生成算法的构造。

参考文献:

[1] SHEHORY O, KRAUS S. Task allocation via coalition formation among autonomous agents [A]. In O Shehory ed. Proc of IJCAI95 [C]. Los Angeles, CA, USA, Morgan Kaufmann Publishers, 1995.

[2] SANDHOLM T, LARSON K, ANDERSSON M, et al. Anytime coalition structure generation with worst case guarantees [A]. In T Sandholm ed. Proc of the National Conference on Artificial Intelligence [C]. Madison, WI, 1998.

[3] SANDHOLM T, LESSER V. Coalition among computationally bounded agents [J]. Artificial Intelligence, 1997, 94(1): 99 - 137.

[4] 蒋建国, 夏娜, 齐美彬, 等. 一种基于蚁群算法的多任务联盟串行生成算法[J]. 电子学报, 2005, 33(12): 2178 - 2182.

JIAN G Jianguo, XIA Na, QI Meibin, et al. An ant colony algorithm based multi-task coalition serial generation algorithm[J]. Acta Electronica Sinica, 2005, 33(12): 2178 - 2182.

[5] 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.

XIA Na, JIANG Jianguo, WEI Xing, et al. Searching for Agent coalition for single task using improved ant colony algorithm[J]. Journal of Computer Research and Development, 2005, 42(5): 734 - 739.

[6] 陶海军, 王亚东, 郭茂祖, 等. 基于熟人联盟及扩充合同网协议的多智能体协商模型[J]. 计算机研究与发展, 2006, 43(7): 1155 - 1160.

TAO Haijun, WANG Yadong, Guo Maozu, et al. A multi-Agent negotiation model based on acquaintance coalition and extended contract net protocol[J]. Journal of Computer Research and Development, 2006, 43(7): 1155 - 1160.

[7] 蒋建国, 夏娜, 于春华. 基于能力向量发挥率和拍卖的联盟形成策略[J]. 电子学报, 2004, 32(12A): 215 - 217.

JIAN G Jianguo, XIA Na, YU Chunhua. The coalition formation strategy based on capability vector contribution-rate and auction [J]. Acta Electronica Sinica, 2004, 32(12A): 215 - 217.

[8] 李亚东, 李从东, 张炎亮. 动态联盟收益分配问题的博弈研究[J]. 工业工程, 2006, 9(3): 15 - 18.

LI Yadong, LI Congdong, ZHANG Yanliang. Research on the profit-allotting problem of dynamic-alliances through the game theory [J]. Industrial Engineering Journal, 2006, 9(3): 15 - 18.

[9] SEN S, P DUTTA S. Searching for optimal coalition structures [A]. In S Sen ed. Proc. of the 4th ICMAS [C]. Boston, USA, 2000.

[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.

HU Shanli, SHI Chunyi. An anytime coalition structure generation algorithm [J]. Journal of Software, 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.

XU Jinhui, ZHANG Wei, SHI Chunyi, et al. A structure-oriented mechanism of agent organization formation

and evolution[J]. Journal of Computer Research and Development, 2001, 38(8): 897 - 903.

[12] SEN S, DUTTA P S. Searching for optimal coalition structures[A]. In: Proc the 4th ICMA[C]. Boston, MA, USA, 2000.

[13] 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, 2002.

LUO Zhenghu. Research on several key problems in mobile agent systems[D]. Hefei: Hefei University of Technology, 2002.

[14] 罗东坤,丁治国. 石油工程项目管理机制[J]. 石油勘探与开发, 2006, 33(2): 242 - 245.

LUO Dongkun, DING Zhiguo. Petroleum engineering project management [J]. Petroleum Exploration and Development, 2006, 33(2): 242 - 245.

[15] 李惠君,王志宇,贾勤,等. Agent 技术在虚拟企业创建过程中的应用研究[J]. 计算机工程与设计, 2006, 27(15): 2805 - 2807.

LI Huijun, WANG Zhiyu, JIA Qin, et al. Application study of agent technology in creating process of virtual enterprise [J]. Computer Engineering and Design, 2006, 27(15): 2805 - 2807.

[16] 李卫宁. 基于 MAS 的智能诊断技术集成方法研究[J]. 运筹与管理, 2005, 14(5): 13 - 17.

LI Weinin. Research on the method of expert's evaluation in Fault diagnosis system based on MAS [J]. Operations Research and Management Science, 2005, 14(5): 13 - 17.

[17] 蒋建国,夏娜. 基于 MAS 的分布式智能控制初探[J]. 合肥工业大学学报(自然科学版), 2002, 28(9): 1085 - 1088

JIANG Jianguo, XIA Na. Research on the MAS based intelligent distributed control system [J]. Journal of Hefei University of Technology (Natural Science), 28(9): 1085 - 1088.

[18] KENNEDY J, EBERHART R C. Particle swarm optimization[A]. In Kennedy J ed. Proc IEEE Int Conf on Neural Networks[C]. Perth, 1995.

[19] EBERHART R C, KENNEDY J. A new optimizer using particle swarm theory [A]. In Eberhart R C ed. Proc 6th International Symposium on Micro Machine and Human Science[C]. Nagoya, 1995.

[20] 高尚,杨静宇. 群智能算法及其应用[M]. 北京:中国水利水电出版社, 2006.

作者简介:



蒋建国,男,1955年生,教授,博士生导师,合肥工业大学计算机与信息学院院长,全国信息与电子学科研究生教育委员会理事,安徽省计算机学会副理事长,主要研究方向为智能信息处理、分布式智能系统、数字图像分析与处理. 发表论文30余篇,出版专著1部
E-mail: jgjiang@hfut.edu.cn.



吴琼,女,1983年生,硕士研究生,主要研究方向为智能计算.



夏娜,男,1979年生,副教授,主要研究方向为分布式人工智能、智能控制等.