

48-50

MAS 合作策略 人工智能

多智能体系统

MAS 中合作策略的比较

Comparisons of Cooperation Strategies in MAS

丁晓明 刘博勤

TP18

(西南师范大学计算机系 重庆 400715)

Abstract According to a cooperation mechanism based on the network coordinate computing, this paper presents three cooperation strategies to deal with the cooperation problem of the agents in MAS (Multi-Agent System). These cooperation strategies are non-communication and noncentralized control. Based on the experimental results, this paper compares these strategies in their system efficiency, stability and dynamic property.

Keywords MAS, Cooperation strategy, Efficiency, Stability, Dynamic property

多智能体系统(MAS: Multi-Agent System)是分布式人工智能的主要研究方向之一。在MAS中, Agent群体的合作行为及其优化程度又是一个重要的研究分支。Agent之间的合作,分为系统中是否存在集中控制和在Agent之间是否存在通信。Agent间既无通信又无集中控制的合作虽然存在一些困难,但并非不可实现,在网络协同计算环境中^[1],通过Agent分工及任务招标可以完成这种合作,通过订金、劳动力价格调整及参照历史信息来协调Agent的行为,可以提高合作成功率,从而使智能系统达到较高的性能。

1 三种合作策略

按照文[1]中的合作机制及模型,本文提出如下三种Agent之间合作的策略。

1.1 打工策略

打工策略是一种保守的稳妥策略,采用打工策略的Agent在决策时首选可以独立完成(粒度为1)的任务,如果目前没有这种任务,Agent便去选择一个名望值最高的承包者为其打工,如果这种情况目前也不存在,Agent考虑承包一个任务,Agent对可承包的任务逐个计算估算效用 U_e ,找出 U_e 最大(Max_U_e)的那个任务,如果 $Max_U_e > 0$ 则承包此任务,否则作出决策,目前没有合适的任务,等待一段时间后重新决策。算法如下:

- (1)if 任务公告板上有任务T未被承包且粒度为1 then {Agent作出决策:承包此任务;返回};
- (2) $Max_fame=0$; /* Max_fame 纪录 Agent 的最大名望值 */

- $No=0$; /* No 记录名望值最大的 Agent 编号 */
- (3)从任务公告板上读取一个任务T;
if T已被承包且有空缺职位 then {查此T承包者的名望值 $iAgent_fame$;
if $iAgent_fame > Max_fame$ then { $Max_fame=iAgent_fame$; $No=iAgent_No$ };
- (4)if 还有未检查的任务 then 转(3)else 转(5);
- (5)if $No > 0$ then {Agent作出决策:为编号为No的承包者打工;返回};
- (6)查询历史信息,得到以往所承包完成过的最大任务粒度 Max_Rec ;
- (7) $Max_U_e=0$; /* Max_U_e 记录任务的最大估算效用 */
 $VT_No=0$; /* VT_No 记录最大估算效用的任务编号 */
- (8)从任务公告板上读取一个任务T;
if T未被承包 then {if T的粒度 $\leq Max_Rec$ then $h=1$ else $h=\min$; 计算 U_e ;
if $U_e > Max_U_e$ then { $Max_U_e=U_e$; $VT_No=Task_No$ };
- (9)if 还有未被检查的任务 then 转(8)else 转(10);
- (10)if $Max_U_e > 0$ then {Agent作出决策:承包编号为 VT_No 的任务;返回};
- (11)/* 没有合适的工作可做 */ {Agent作出决策:等待一段时间后再重新决策;结束}。

1.2 平衡策略

平衡策略是一种比较理性的策略。采用平衡策略的Agent对目前所有可参与的任务进行效用估算分析,并参加到估算效用 U_e 最大且有利可图的那个任务中去。在所有任务的 U_e 都计算出来之后,挑选出 U_e 最大的任务 Opt_Choice 。若 Opt_Choice 的 $U_e \leq 0$,表明没有有利可图的任务,Agent决策为等待一段时间再重新决策;若 Opt_Choice 的 U_e 刚好为 $Labor_Price * Work_Time$ 则表示为任务 Opt_Choice 的承包者打工是最有利的,于是Agent

决策为打工;否则承包 Opt-Choice 的估算效用最大,Agent 决策为承包 Opt-Choice 任务,算法如下:

- (1) 查询历史信息,得到以往所承包完成过的最大任务粒度 Max-Rec;
- (2) 从任务公告板上读取一个任务 T;
- (3) 对任务 T 作评价
if T 已被承包 then if 没有空缺 then 则该任务的估算效用 $U_e=0$;
else /* 有空缺 */ $U_e = \text{labor-Price} \cdot \text{Work-Time}$;
if T 未被承包 then if 当前任务粒度小于或等于 Max-Rec then $h=1$ else $h=\min$; 计算 U_e ;
- (4) if 还有未被评价的任务 then 转(2); else 转(5);
- (5) 得到 U_e 最大的任务 opt-choice;
- (6) if opt-choice 的 $U_e \leq 0$ then 返回; /* Agent 的决策结果为:等待一段时间后再重新决策 */
- (7) if opt-choice == Labor-Price * Work-Time then {opt-choice 为空缺最少的任务;返回; /* Agent 的决策结果为:为 opt-choice 作 worker */
else 返回 /* Agent 的决策结果为:承包 opt-choice */
- (8) 结束。

1.3 投机策略

投机策略是一种冒险策略。采用投机策略的 Agent 将对任务公告板上目前公布的所有任务逐一作出评价;对那些未被承包的任务,从中找出粒度最大的一个并记下此任务的粒度 Max-Rec 和任务编号 VT-No;对其它可参与的任务,则计算其 U_e ,从中找出 Max- U_e 任务并记下其承包者的编号 No。在评价全部任务后,Agent 首先检查 Max-Rec 是否大于1。若是则作出决策承包编号为:VT-No 的任务;否则便检查 Max- U_e 是否大于0,若是则决策为编号 No 的 Agent 打工;如果 Max- $U_e \leq 0$,Agent 便检查 Max-Rec 是否等于1,若是就决策承包编号为 VT-No 的任务;如果 Max-Rec=1 的任务都没有,表明目前没有合适的任务,Agent 决策等待一段时间再重新决策,算法如下:

- (1) Max- $U_e=0$; Max-Rec=0; VT-No=0;
No=0;
- (2) 从任务公告板上读取一个任务 T;
- (3) 对任务 T 作评价
if T 未被承包 then if T-Rec > Max-Rec then
{Max-Rec=T-Rec; VT-No=iTask-No};
if T 已被承包且有空缺职位 then { $U_e = \text{Labor-Price} \cdot \text{Work-Time}$;
if $U_e > \text{Max-}U_e$ then {Max- $U_e = U_e$;
No=iAgent-No};
- (4) if 还有未被评价的任务 then 转(2) else 转(5);
- (5) if Max-Rec > 1 then /* 作承包者 */ /* Agent 作

- 出决策:承包编号为 VT-No 的任务;返回;);
- (6) if Max- $U_e > 0$ then /* 作工人 */ /* Agent 作出决策:为编号为 No 的 Agent 打工;返回;);
 - (7) if Max-Rec=1 then {Agent 作出决策:承包编号为 VT-No 的任务;返回;);
 - (8) /* 没有合适的任务 */ /* Agent 作出决策:等待一段时间再重新决策;返回;);
 - (9) 结束。

2 结果比较

1 系统效率 根据文[1]中的系统模型,在设定15,25,40个 Agent 的系统中分别采用打工策略、平衡策略和投机策略进行了实验,并由实验结果绘制的效率曲线图如图1。

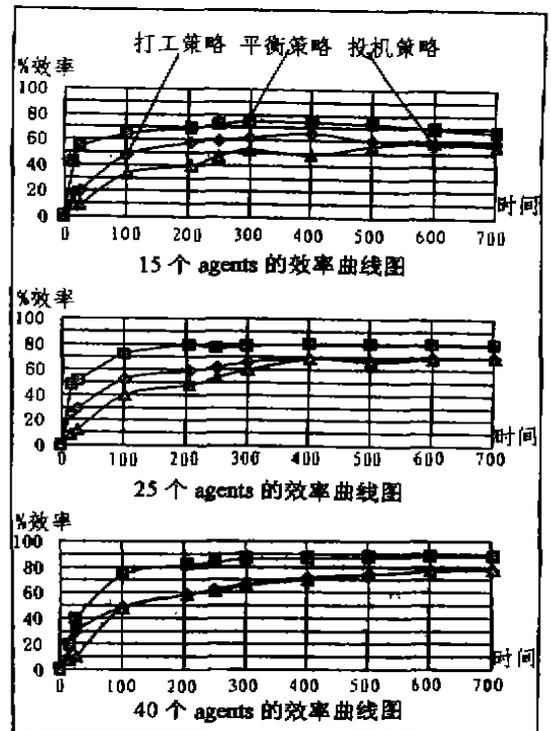


图1

从这些效率曲线图知道,三种合作策略在15,25,40个 Agent 的系统中都达到了可以接受的效率,而且采用平衡策略的系统效率明显高于采用打工策略或投机策略的系统效率。

2 稳定性 从系统效率曲线图可以观察到,合作策略对系统效率的稳定性有着决定性的影响,相对而言,打工策略和平衡策略的系统效率稳定性要

好一点,投机策略的稳定性较差,特别是在15个 Agent 和25个 Agent 组成的系统中,都存在比较明显的效率波动。

3 动态特性 动态特性指在每一个实验中系统效率的收敛速度,即从时间0开始到系统效率达到稳定所经历的时间。从三组实验所绘制的效率曲线中可以看到各种情况下的收敛特性。不管采取何种合作策略,Agent 数目少的系统要比 Agent 数目多的系统收敛得更快。这可能是系统内 Agent 数目太大使得劳动力价格对个体 Agent 的行为变得迟钝,即系统的应变能力更强,这种迟钝又被 Agent 以历史信息的形式记录下来,从而导致了系统的慢速收敛。

从效率曲线图还可以发现不同合作策略对系统动态特性的影响,投机策略的收敛速度慢于打工策略和平衡策略,且在收敛过程中,效率波动也明显多于其余两种合作策略。

4 Agent 数目对系统性能的影响 从实验结果可以观察到,不管采用何种策略,Agent 数目的差异对系统性能都存在明显的影响。结果表明:Agent 数目的大小与系统的稳定后效率成正比;与系统稳定性高低成正比;与系统的收敛速度成反比。

5 劳动力价格的变化情况 实验对15个 Agent 组成的系统,分别采用三种合作策略时,记载了劳动力价格变化的情况,如图2。

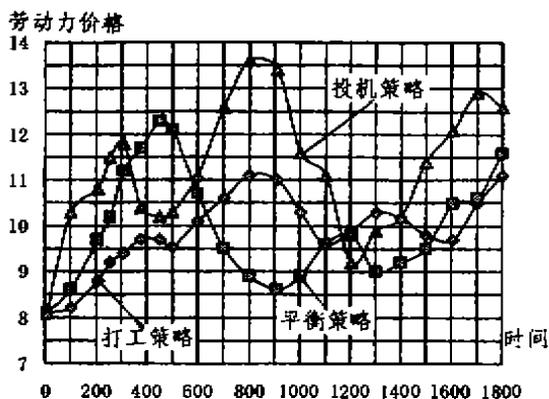


图2 15个 Agents 系统的劳动力价格变化曲线

可以观察到在各种决策策略下,劳动力价格都不断在波动。比较三种合作策略下的劳动力价格变化曲线,可以发现采用打工策略的系统中,劳动力价格变化幅度最小,采用平衡策略的系统次之,而采用投机策略的系统劳动力价格变动幅度最大。这说明劳动力价格的变化受决策策略因素的影响是比较

明显的。

结束语 在由独立自治 Agent 组成的 MAS 中,由于缺乏通信,Agent 之间的合作产生存在许多困难,但在要求 Agent 遵循一定的合作规则和增加一些协调机制之后,合作是可以没有通讯的情况下产生的。本文提出的三种不需通讯且无集中控制的合作策略,实验表明都是可行的。全面衡量系统性能,可以认为平衡策略是三种合作策略中最好的一种,打工策略要稍差些,而投机策略更差一些。从实验结果分析可见,引入系统中的协调机制发挥了较为理想的作用。另外,能够良好合作并得到较高效率的系统不能由较少的 Agent 组成。

从实验结果也能发现这三种合作策略存在的问题。由于 Agent 之间不存在通信,系统中也不存在集中控制,Agent 在决策时只能按照一种固定的方式作出判断,加上所有 Agent 均采用同一种决策策略,因此系统中 Agent 的行为具有趋同性,这样造成系统的稳定性变差,而且效率也受到影响。可以有两种改进的研究方案。一种是采用混合策略,即系统中的 Agent 可以在本文提出的三种合作策略中任意选择一种,这样会降低 Agent 行为的趋同性,从而提高系统的稳定性和效率。另一种是在加进一定集中控制的情况下采用本文中的三种合作策略,即采用一种计划性和市场性并存的方法,以期进一步提高系统性能。

参考文献

- 1 丁晓明,刘博勤. MAS 中合作机制的研究. 计算机科学, 1999, 26(2)
- 2 Cammarata S. Strategies of Cooperation in Distributed Problem Solving. IJCAI83
- 3 Rosenschein J S. Communication-Free Interaction Among Rational Agents: A Probabilistic Approach. WS DAI, 1988
- 4 Rosenschein J S, Genesereth M. Cooperation Without Communication. In: Readings in DAI, 1986
- 5 Axelrod R. The Evolution of Cooperation. Basic Books Inc, 1984
- 6 Lux A, Steiner D. Understanding Cooperation: an Agent's perspective. In: ICMAS'95, Lux 261-268
- 7 Shoham Y. Agent-oriented programming. In: Artificial Intelligence, 1993, 60: 51~92
- 8 Werner E. Distributed Algorithms for Cooperating Agents. In: Huhns M N eds. WS DAI, 1990