第 35 卷 第 11 期 计 算 机 工 程 2009 年 6 月
Vol.35 No.11 Computer Engineering June 2009
·人工智能及识别技术· 文章编号：1000—3428(2009)11—0221—02 文献标识码：A 中图分类号：TP18

多 Agent 系统任务分配的研究

严建峰，李伟华，刘 明
(西北工业大学计算机学院，西安 710072)

摘 要：针对多 Agent 系统中存在的任务分配问题，提出一种任务与 Agent 之间多对多的分配模式，建立相应的任务分配模型，定义任务分配中的性能影响因子，并推导出进行任务分配优化的目标函数，通过一个实际案例，验证该方法的有效性。
关键词：多 Agent 系统；任务分配；目标函数

Research on Task Allocation in Multi-Agent System

YAN Jian-feng, LI Wei-hua, LIU Ming
(School of Computer, Northwestern Polytechnical University, Xi'an 710072)

【Abstract】Aiming at the task allocation problem in Multi-Agent System(MAS), a novel many-to-many allocation mode between task and Agent is put forward and the corresponding mathematics model is established. The performance influence factor is defined, and the objective function is deduced to conduct optimization of task allocation. This method is proved effective through a practical case.
【Key words】Multi-Agent System(MAS); task allocation; objective function

1 概述

任务分配问题最早出现在各类生产、计划和柔性制造系统中，是种典型的组合优化问题。由于传统任务分配问题的目标是充分利用分布式系统多处理机的并行性，因此将一个问题(problem)分解成多个子任务(task)，通过子任务的并行执行使得问题能在最短的时间内完成。在这类系统中的任务分配大多基于以下的设定：系统所有处理机都能无差别地处理所有任务，即对于同一个任务，所有处理机都应返回相同的结果，因此，同一任务只须分配给一个执行者。任务对处理机是独占的，处理机在完成现在任务之前不会接受新的任务。这种设定在很多应用环境，如车辆运输调度、指派问题、多处理器并行处理、网格计算等应用领域是适用的。

与传统任务分配问题相比，多 Agent 系统(Multi-Agent System, MAS)的任务分配有其特殊性，主要体现在 2 个方面：(1)对同一问题采用不同方法设计的 Agent 的处理结果可能存在不一致[1]，因此，为提高任务的处理精度，在 MAS 中将一个任务分配给多个 Agent 进行并行求解，将不同 Agent 求解的结果进行冲突消解[2]是种自然的方法；(2)在 MAS 中允许 Agent 同时接受多个任务，形成对应的任务列表，从而充分利用 MAS 的并行求解能力。

有效的任务分配是 MAS 中任务处理的前提和基础，直接影响 MAS 的性能，是 MAS 的热点研究领域。文献[3]提出基于合同网的任务列表法并以多机器人为对象进行实验与分析，有效解决了动态环境下多机器人的实时任务分配问题。文献[4]基于集覆盖理论，将任务分配问题转换成集覆盖问题。文献[5]提出基于信息粒化理论，分为主体、任务粒化阶段和粒内规划阶段 2 个阶段完成任务分配。文献[6-7]利用计算智能方法强大的寻优能力进行任务分配机制设计。然而这些工作主要基于传统任务分配问题的设定进行研究，未全面考虑 MAS 任务分配的特殊性。本文给出一种 MAS 任务分配方法，并用一个算例验证了算法的有效性。

2 MAS任务分配问题

传统任务分配问题可描述为：有 n 个任务需要分配给 n 个处理机去完成，系统中任意处理机都有处理任意任务的能力，但每个任务只能分配给 1 个处理机，且每个处理机只能处理 1 个任务；不同的分配花费不同的代价。在这种设定下，任务与处理机之间是一对一的分配模式。任务分配问题的目标是找到 1 种最优分配方案，使其所花费的代价最小。

MAS 任务分配问题可描述为：有 m 个任务需要分配给 n 个 Agent 完成，每个任务可以分配给多个 Agent 处理，并对结果进行冲突消解，得到最终结果，且每个 Agent 允许接受在其能力范围之内的多个任务。因此，MAS 采用的是种多对多的任务分配模式。该分配模式一方面可以综合利用多个 Agent 的求解能力，提高系统的处理性能；另一方面，多对多的任务分配模式与传统一对一的分配模式相比，会产生一定的重复计算，导致更大的分配代价。本文提出的任务分配机制的设计思想综合考虑了 MAS 系统性能和分配代价两方面的影响，以寻求性能和代价之间的一个平衡。

不失一般性，设 1 个任务可以有多个 Agent 中标，但为避免中标 Agent 数量过多，引起不必要的资源开销，设定 1 个中标 Agent 上限阈值 u，同时，允许每个 Agent 能同时接受多个任务，为避免 Agent 的任务列表过大而引起任务并发执行时产生瓶颈，设定一个任务列表上限阈值 v，与传统任务分配单纯考虑分配代价最小的设计思想不同，本文的设计思想是综合考虑分配代价和 MAS 的性能，寻找代价与性能之间的平衡点。据此，本文讨论的任务分配机制的设计目标

基金项目：国家部委基金资助项目(51315080404, 9140A17050206HK03)
作者简介：严建峰(1978－)，男，博士研究生，主研方向：智能故障诊断系统；李伟华，教授、博士生导师；刘 明，硕士研究生
收稿日期：2008-12-30 **E-mail：**yjf@mail.cdgdc.edu.cn

是：在中标 Agent 数量不超出阈值 u，并且任务列表不超出阈值 v 的前提下，对同一任务尽可能综合利用多种 Agent 方法来提高 MAS 的处理性能，并使任务执行的综合代价最小。

3 任务分配问题的数学模型和目标函数

根据 MAS 任务分配的定义，给出任务分配的数学模型，设计任务分配代价函数来求解任务分配代价，引入性能影响因子来评估多对多分配模式对 MAS 的性能提升，并结合任务分配代价函数和性能影响因子，给出任务分配的目标函数。

3.1 任务分配模型

不同 Agent 执行任务的代价有差异，为衡量并对比这种差异，需要求解系统中每个 Agent 对每个任务类型的求解代价。用矩阵的方式来表示求解代价，矩阵的行表示 Agent，矩阵的列表示任务类型，矩阵的元素表示 Agent 求解该任务的代价，称为全局代价矩阵。基于全局代价矩阵，可以构建任务分配中的 3 个数学模型：先从全局代价矩阵中提取与具体问题相关的元素来构建问题的代价矩阵，称为局部代价矩阵；其次根据局部代价矩阵，构建无约束分配矩阵，并进一步构建约束分配矩阵；最后根据约束分配矩阵和局部代价矩阵，构建该问题对应的分配代价矩阵。

(1)全局代价矩阵

设系统中共有 p 个 Agent、q 个任务类型且 Agent i 对任务 j 的完成代价为 c_{ij}，可构建代价矩阵 $\boldsymbol{C}_{p\times q}=(c_{ij})$，其中，$c_{ij}$ 包括时间、系统软硬件开销等资源。考虑到在各类代价中，时间代价是个重要因素，在各类 MAS 中的重要性可能有所不同，因此，将 c_{ij} 表述为

$$c_{ij}=\omega_1\times t_{ij}+\omega_2\times\sum_{k=1}^{s}\lambda_{ijk} \tag{1}$$

其中，t_{ij} 为时间资源开销；λ_{ijk} 为其他资源开销；s 为其他资源开销的数量；ω_1，ω_2 为对应权重，且 $\omega_1+\omega_2=1$，专家可以根据不同的系统应用需求来调整 ω_1，ω_2 的值，并约定若该 Agent 不能完成某类任务，则 $c_{ij}=N$。

(2)局部代价矩阵

设某问题分解成 n 个任务，共有 m 个 Agent 是潜在任务中标者，从全局代价矩阵中提取与该问题相关的元素构成局部代价矩阵，具体方式为从全局代价矩阵的 p 行(Agent)中抽取 m 行，q 列(任务)中抽取 n 列，则 m 行与 n 列交叉的元素构成局部代价矩阵。

(3)无约束分配矩阵和约束分配矩阵

对局部代价矩阵中的元素作一定变换即可构成无约束分配矩阵 $\overline{\boldsymbol{R}}_{m\times n}$，变换方式为：若 $c_{ij}=N$，则 $\overline{r}_{ij}=N$；若 c_{ij} 不等于 N，则令 $\overline{r}_{ij}=1$。

在满足约束条件的前提下，对无约束分配矩阵 $\overline{\boldsymbol{R}}_{m\times n}$ 中的每个元素进一步变换，可构建约束分配矩阵 $\boldsymbol{R}_{m\times n}=(r_{ij})$。分配问题的约束条件 u，v 可以表述为

$$\sum_{i=1}^{m}r_{ij}\leqslant u \tag{2}$$

$$\sum_{j=1}^{n}r_{ij}\leqslant v \tag{3}$$

具体的变换如式(4)所示：

$$r_{ij}=\begin{cases}1 & \text{若}\overline{r}_{ij}=1\text{且任务分配给该Agent}\\ 0 & \text{若}\overline{r}_{ij}=1\text{且任务不分配给该Agent}\\ N & \text{若}\overline{r}_{ij}=N\end{cases} \tag{4}$$

(4)分配代价矩阵

根据具体问题的约束分配矩阵和局部代价矩阵，可直接建立该问题对应的分配代价矩阵 $\boldsymbol{D}_{mn}=(d_{ij})$，其中，元素 $d_{ij}=r_{ij}\times c_{ij}$，并约定若 r_{ij} 或 c_{ij} 的值为 N，则 $d_{ij}=N$，即如果 Agent 不能完成任务，那么不需要计算对应分配代价。

(5)任务分配代价函数

建立问题的分配代价矩阵后，给出任务分配代价函数如式(5)所示：

$$C(x)=\sum_{j=1}^{n}\sum_{i=1}^{m}d_{ij} \tag{5}$$

3.2 性能影响因子

在 MAS 中，任务处理的性能是任务中标 Agent 的一个非递减函数[1]，即在多数情况中，增加中标 Agent 的数量能够提高 MAS 的性能。基于该考虑，引入性能影响因子来评估 MAS 的中标 Agent 的数量对性能的影响。根据约束分配矩阵，定义如下性能影响因子：

$$\lambda=\left(\sum_{j=1}^{n}\sum_{i=1}^{m}r_{ij}\right)^{-3} \tag{6}$$

3.3 目标函数

为实现任务分配机制，必须建立任务分配的目标函数，指导 Agent 与任务进行组合优化，求解最优任务分配方案。根据任务分配机制基于性能和代价之间达到平衡的设计目标，将性能影响因子与任务分配代价模型相结合来构建目标函数。

根据任务分配代价函数和性能影响因子，建立如式(7)所示的目标函数：

$$F(x)=\lambda\times C(x)=\sum_{j=1}^{n}\sum_{i=1}^{m}d_{ij}\times\left(\sum_{j=1}^{n}\sum_{i=1}^{m}r_{ij}\right)^{-3} \tag{7}$$

建立 MAS 任务分配的数学模型和目标函数以后，应用各类方法对任务分配矩阵进行组合优化并达到性能与代价之间的平衡的判断依据即为 $F(x)$ 最小。

4 实例分析

设已建立某 MAS 的全局代价矩阵，某问题被分解成 5 个任务，系统中有 5 个潜在 Agent，设其对应的局部代价矩阵如式(8)所示：

$$\boldsymbol{C}_{5\times5}=\begin{pmatrix}1&5&N&N&8\\2&N&N&3&5\\N&6&N&N&1\\1&3&9&1&N\\N&3&4&2&N\end{pmatrix} \tag{8}$$

对式(8)进行变换，则该问题的无约束分配矩阵如式(9)所示：

$$\boldsymbol{R}_{5\times5}=\begin{pmatrix}1&1&N&N&1\\1&N&N&1&1\\N&1&N&N&1\\1&1&1&1&N\\N&1&1&1&N\end{pmatrix} \tag{9}$$

设约束条件为 $u\leqslant2$，$v\leqslant2$，即 $\sum_{i=1}^{5}r_{ij}\leqslant2$；$\sum_{j=1}^{5}r_{ij}\leqslant2$，则根据约束条件可得到约束矩阵 $\boldsymbol{A}$, $\boldsymbol{B}$，其中，$\boldsymbol{A}$ 倾向于利用多个 Agent 求解来提高 MAS 性能；$\boldsymbol{B}$ 倾向于减少中标 Agent 的数量来节省计算资源。结合局部代价矩阵得出 $\boldsymbol{A}$, $\boldsymbol{B}$ 对应的代价分配矩阵 $\boldsymbol{D}_A$, $\boldsymbol{D}_B$ 分别如式(10)、式(11)所示：

(下转第 225 页)