针对交互关系模型优化的协作式问题分配

刘茜萍[1], 窦万春[1], 蔡士杰[1], 唐加山[2]
(1. 南京大学计算机软件新技术国家重点实验室, 南京大学计算机科学与技术系, 江苏南京 210093;
2. 南京邮电大学数理学院, 江苏南京 210003)

摘　要: 将协作式问题合理地分配给各实体, 通过对实体之间交互关系模型的优化获得较少交互关系数, 对于减少实体的交互工作量, 简化协作管理行为, 降低通信代价, 从而有效高质地解决该协作式问题具有重要意义. 根据题内交互关系以及题外交互关系的定义, 在分析以最小化实体之间交互关系数为目标的数学模型的基础上, 提出了三个选择原则, 以从每个子问题的各候选团队中选出合适的执行团队, 并在此基础上设计实现了递阶优化分配算法, 给出了模型的较优解.

关键词: 分配; 交互关系数; 选择原则; 递阶优化
中图分类号: TP301　文献标识码: A　文章编号: 0372-2112 (2007) 02-0345-05

Allocation of Cooperative Problem Aiming at Interaction Relationship Model Optimization

LIU Xi-ping[1], DOU Wan-chun[1], CAI Shi-jie[1], TANG Jia-shan[2]
(1. State Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing, Jiangsu 210093, China; 2. College of Mathematics and Physics, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China)

Abstract: To solve a complex cooperative problem effectively, the reasonable allocation is very important. By allocating a cooperative problem to multiple entities based on optimization of the interaction relationship model, the interaction relationship number could shrink, which could decrease the interaction quantity, facilitate the management on cooperation, reduce the communication cost, and sequentially underlie the effective and high quality solving to a complex cooperative problem. Based on the definition of the intra-interaction and extra-interaction relationship, a mathematical model with target function on the minimization of the interaction relationship number is analyzed. With three proposed selection principles, a sequential optimized allocation algorithm is designed and implemented to provide a better solution of that mathematical model. Experiments prove that interaction relationship number allocated by this method is much less than other methods, which guarantees that the cooperative problem could be solved more effectively.

Key words: allocation; interaction relationship number; selection principle; sequential optimization

1　引言

协作式问题[1]是指单个实体无法独力承担, 必须由多个实体以协作执行的方式共同解决的问题. 这里的实体可以是用于制造零件的机器, 或是执行程序的处理器, 抑或是某个代理(agent), 还可以是有着高智能行为的人. 协作式问题常可根据功能或逻辑关系细分为若干个子问题, 每个实体可以参与其中一个或多个子问题的解决, 并与其它实体适时通信以获取必要的相关信息, 进而以分而治之的方式实现一个大规模复杂问题的协作解决. 实体解决协作式问题的行为通常可分为自治处理和交互处理两种形式. 所谓自治处理即各实体独自处理所分配到的子问题; 而交互处理则指实体在解决所分配到的子问题时与其它实体之间互相传递信息、协商讨论等行为. 问题的协作解决性正是通过各实体的交互处理行为得以充分展开.

根据交互处理的复杂性, 可将协作式问题划分为两类. 一类是交互相对简单的协作式问题, 如机械制造工业、海量计算等问题. 通常此类问题的交互仅止于简单的数据传递, 交互量相对固定. 而另一类协作式问题的交互则较为复杂, 如协同认知[2], 协同设计[3]等问题. 各实体间多是交互量相对多变的信息、知识传递, 此时交互处理有可能在整个问题解决活动中占很大比重, 交互处理的质量在很大程度上影响着这类问题的有效解决.

针对上述协作式问题, 往往会在正式解决问题之前, 先在子问题集合与实体集合之间建立一个映射, 以指出实体与子问题之间的执行关系, 即确定某实体是否参与解决某个子问题; 同时还要指出实体集合中的交互关系, 即确定某实体是否应与另一实体交互以获取解决其所负责子问题的必要信息, 此即协作式问题的分配过程. 已有分配方法[4~11]大多没有对复杂协作式问题解决过程中交互的动态性、复杂性给予足够关注, 适用范围多局限于交互相对简单的协作式问题. 然而, 随着跨学科、跨领域合作的广泛开展和逐渐深入, 协作式问题

收稿日期: 2006-03-06; 修回日期: 2006-08-02
基金项目: 国家自然科学基金(No. 60303025, No. 60673017); 江苏省基础研究计划(No. BK2004411, No. BK2005208)

所涉及的交互日趋复杂，迫切需要对交互复杂的协作式问题分配方法进行研究。针对这种现状，我们从交互关系视角建立了数学模型并提出了一种有效的递阶优化方法以分配此类复杂协作式问题。

2 待分配问题模型以及分配结果模型定义

一个协作式问题往往可细分为若干个子问题，这些子问题对应着协作过程中更细粒度的、具体详实的问题描述，子问题之间可能存在依赖关系，它表明一子问题将从其它子问题处获取自己所需信息或向其它子问题提供信息。一个子问题可能对应多个有能力解决该子问题的候选对象，一个候选对象可能包含一个或多个实体，又称为候选团队，其中成员为候选队员[12]。当一个候选团队被选定用以解决某个子问题时，即成为该子问题的执行团队。相应的，选定候选团队中的所有候选队员即成为对应子问题的执行队员。图 1 给出了分配所涉及的概念及其之间的关系。

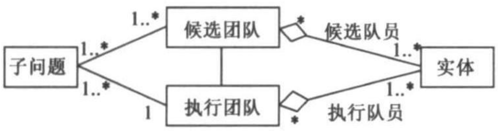


图 1 协作式问题分配中的概念

定义 1 待分配问题模型为$(S, DEPS\text{-}S, P, CAND)$。其中，$S$ 表示某个待分配的协作式问题，它最多可细分为 m 个子问题，有 $S=\{s_i, i=1, 2, \cdots, m\}$；$DEPS\text{-}S$ 表示各子问题之间的依赖关系，有 $DEPS\text{-}S=\{(s_{i_1}, s_{i_2})$：$s_{i_1}$与 s_{i_2}之间存在依赖，其中 $s_{i_1}, s_{i_2}\in S$，且 $i_1\neq i_2\}$；P 表示可能参与问题解决的实体集合，有 $P=\{p_j, j=1, 2, \cdots, n\}$；$CAND$ 表示各子问题的候选团队集合，有 $CAND=\bigcup_i C_i$，而 $C_i=\bigcup_t C_{i,t}(t\in[1, n])$ 为 s_i 的候选团队集合，其中 $C_{i,t}$表示s_i的所有成员数为t 的候选团队的集合，而候选团队成员即为 P 中的元素。

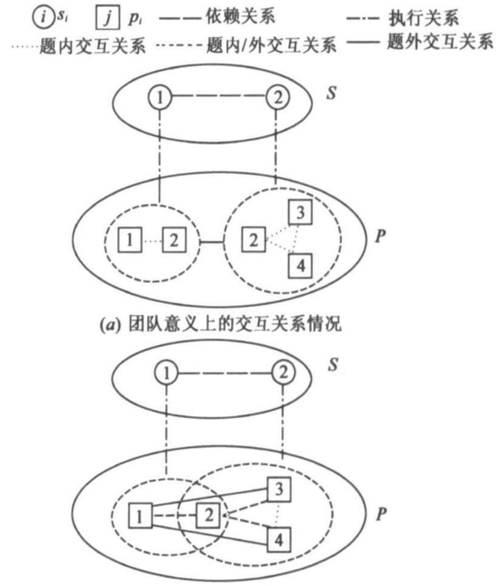
在选定执行团队之后，还需结合问题模型中子问题之间的依赖关系推导出各实体之间的交互关系。交互关系又可细分为题外和题内两种形式，如图 2 所示，具体定义如下。

定义 2 题外交互关系，指从原问题模型中各子问题之间的依赖关系过渡而得的，位于不同子问题的执行队员之间的交互关系。题内交互关系，指由子问题执行队员的不唯一性得出的，位于同一子问题的多个执行队员之间的交互关系。

由于对复杂协作式问题而言，存在交互关系的两个实体之间交互量的多少取决于协作式问题的实际解决过程，是动态可变的，无法在分配阶段提前预知，故在分配阶段即本文研究内容中，仅讨论实体之间交互关系的数量，而对执行过程中某交互关系对应的实际交互量不作考虑，仅计任两个实体之间的交互关系数为 0 或 1。

定义 3 一个协作式问题分配的最终结果模型为$(EXES\text{-}P, INTP\text{-}P)$，其中 $EXES\text{-}P=\{(s_i, p_j): p_j$ 为 s_i 的一个执行队员，$s_i\in S, p_j\in P\}$；$INTP\text{-}P=\{(p_{j_1}, p_{j_2}): p_{j_1}$和 p_{j_2}之间存在交互关系，$p_{j_1}, p_{j_2}\in P$，且 $p_{j_1}\neq p_{j_2}\}$。

协作式问题的分配过程即为对任意待分配问题模型$(S, DEPS\text{-}S, P, CAND)$，按照一定策略获取分配结果模型$(EXES\text{-}P, INTP\text{-}P)$的过程。



(a) 团队意义上的交互关系情况

(b) 单个执行队员上的交互关系情况

图 2 题内交互关系和题外交互关系示例

3 分配标准分析与数学模型描述

分配的策略由不同的标准决定，而分配涉及两方面基本准则：功能和性能。对功能要求而言，必须确保为子问题挑选的执行队员确实有能力处理该问题，即执行团队必须来自子问题对应的候选团队集合。对性能要求而言，本文以最小化各实体之间的交互关系数量为根本目标，将其作为有效高质解决复杂协作式问题的一个重要指标，这主要出于如下考虑：首先，一个实体的过多交互处理行为将直接牵制其在自治处理行为上的时间和精力，通过交互关系的减少，可以在客观上间接为各实体减轻交互的负担，同时也减少了一个实体在与多个实体的交互关系之间无缝切换的困扰。其次，减少交互关系可以缩小管理范围，降低管理难度，从而实现有序高效的管理。再次，避免过多交互关系的建立，对减少通信开销有重大意义。尤其是对通信建立代价与数据传输代价比值较高环境下的协作式问题，如移动设备上进行的协作行为，交互关系的减少将显得更为重要。

表 1 模型所用符号说明

名称	意义	解释	备注
$D(i_1, i_2)$ $i_1, i_2=1, \cdots, m$	两个子问题之间的依赖关系	若 s_{i_1}与 s_{i_2}存在依赖关系则 $D(i_1, i_2)=1$；否则 $D(i_1, i_2)=0$	数据来自问题模型中的 $DEPS\text{-}S$
$R(i, j)$ $i=1, \cdots, m$; $j=1, \cdots, n$	子问题与实体的执行关系	若 p_j 参与了 s_i 的执行则$R(i, j)=1$；否则 $R(i, j)=0$	反映分配结果模型中 $EXES\text{-}P$
$INT(j_1, j_2)$ $j_1, j_2=1, \cdots, n$	各实体之间的交互关系	若 p_{j_1}需要与 p_{j_2}进行交互则 $INT(j_1, j_2)=1$；否则 $INT(j_1, j_2)=0$	反映分配结果模型中的 $INTP\text{-}P$

分配策略的不同会造成结果模型中不同的交互关系数[12]. 本文采用如下数学模型来形式化地对协作式问题的分配进行描述, 模型所用符号的说明如表 1 所示. 分配可描述为如下优化问题: 为每个子问题选择适当的实体来完成子问题的解决, 使得

$$\min \sum_{j_1=1}^{n-1} \sum_{j_2=j_1+1}^{n} INT(j_1, j_2)$$

$$\text{s.t.}$$

$$\forall i, \bigcup_{R(i,j)=1} \{P_j\} \in C_i$$

$$INT(j_1, j_2) = \bigvee_{i_1, i_2=1}^{m} ((R(i_1, j_1) \wedge R(i_2, j_2) \wedge D(i_1, i_2)) \vee (R(i_1, j_1) \wedge R(i_1, j_2)))$$

其中 $R(i_1, j_1) \wedge R(i_2, j_2) \wedge D(i_1, i_2)$ 表示 p_{j_1} 和 p_{j_2} 是否存在相对于 s_{i_1} 和 s_{i_2} 的题外交互关系, 而 $R(i_1, j_1) \wedge R(i_1, j_2)$ 则表示 p_{j_1} 和 p_{j_2} 是否存在相对于 s_{i_1} 的题内交互关系. 该模型的求解艰难复杂, 一种常见方法是列出所有可能分配方案并计算其交互关系数, 再从中选择最小者作为最优解, 然而这种穷举方法计算量巨大, 对于稍具规模的协作系统而言非常低效且求解周期太长, 故理论上虽能求得最优解却不具实用价值. 从相关数学理论可知, 这类组合优化问题极有可能是个 NP 完备问题[13], 不能保证一定获得有效的最优解. 在下一节中, 我们设计实现了一个分配方法, 它能给出一个较优的分配方案, 是获得最优解的一个有效尝试.

4　递阶优化分配

递阶优化分配通过 m 个步骤进行, 每一步完成一个子问题的分配, 通过对已分配结果的分析和未来可能分配情况的预测来防止更多交互关系的产生, 使每一步分配造成的新增交互关系数最小. 为方便论述, 下文以 s_i 表示第 i 步待分配的子问题, $i \in [1, m]$.

4.1　题内交互关系的减少

对同一子问题, 其执行队员越少, 题内交互关系数也会越少, 相应的, 该问题就越能以更自治的方式得到处理. 因此成员数 t 越小的候选团队优先级别越高, 称候选团队集合中拥有最小 t 的候选团队为首选候选团队, 构成的集合为首选候选团队集合, 表示为 $C_{i, t_{\min}}$, $i = 1, 2, \dots, m$.

选择原则 1: 执行团队必须来自首选候选团队集合, 即必须是成员个数最少的候选团队.

将 s_i 分配给不同首选候选团队会得到不同的交互关系模型, 对应新增题内交互关系数可由定理 1 求得.

定理 1　分配 s_i 时相对上一步分配结果模型新增的题内交互关系数计为

$$INTRAINT = \begin{cases} 0, & t_{\min} = 1 \quad (1) \\ C_{t_{\min}}^2 - \sum\limits_{R(i,j_1)=1, R(i,j_2)=1} INT_{i-1}(j_1, j_2), & t_{\min} > 1 \quad (2) \end{cases}$$

其中 $INT_{i-1}(j_1, j_2)$ 表示上一步结果模型中两执行队员的交互关系情况.

证明　(1) $t_{\min} = 1$ 时, 由一个实体独力处理 s_i, 故无需进行题内交互, 所以 $INTRAINT = 0$.

(2) $t_{\min} > 1$ 时, 各执行队员之间需建立两两交互, 而其中部分交互关系可能在上一步结果模型中已经存在, 故无需重复添加, 则实际新增题内交互关系数如式(2)所示.

4.2　题外交互关系的减少

如果某个已分配过的子问题 $s_{i'}(i' \in [1, i-1])$ 与 s_i 存在依赖关系, 则对应执行队员须与 s_i 的执行队员建立交互关系, 这样的实体集合称作 $PTOINT$, 表示为 $PTOINT = \{p_j \in P: D(i', i) \wedge R(i', j) = 1, i' = 1, \dots, i-1\}$, 下文以 $I(i, j)$ 表示 p_j 是否为 s_i 的 $PTOINT$ 中一员, 是则为 1, 否则为 0.

定理 2　分配 s_i 时, 相对上一步结果模型新增的题外交互关系数 $EXTRAINT$ 计为

$$\begin{aligned} EXTRAINT = {} & |PTOINT| \times t_{\min} - \sum_{I(i,j)=1} R(i, j) \\ & - \sum_{j_1 \neq j_2, R(i,j_1)=1, I(i,j_1)=1} INT_{i-1}(j_1, j_2) \\ & - \sum_{\substack{j_1=1, \\ R(i,j_1)=1}}^{n-1} \sum_{\substack{j_2 > j_1, R(i,j_2)=1, \\ INT_{i-1}(j_1,j_2)=0}}^{n} I(i, j_1) \wedge I(i, j_2) \end{aligned} \quad (3)$$

证明　子问题 s_i 的 $t_{\min}$ 个执行队员都应与所有 $PTOINT$ 的成员建立交互关系, 故应建交互关系数为 $|PTOINT| \times t_{\min}$. 这其中有部分交互关系可以不必建立或应该避免重复建立, 如以下三种情况:

(1) 同一实体既是 $PTOINT$ 的成员又是 s_i 的执行队员, 该实体同时处理 s_i 及与之有依赖关系的子问题, 则对应的题外交互关系消融于该实体内部, 故无需建立. 此类情况如式(3)第二项所示.

(2) 上一步结果模型中已经存在的部分应建交互关系, 计数如式(3)中第三项.

(3) 上一步结果模型中不存在的, 应建交互关系中的重复交互关系, 即两个同时具有 s_i 执行队员与 $PTOINT$ 成员双重身份的实体之间的交互关系, 计数为式(3)中第四项.

综上所述, 最后实际新增题外交互关系数应计为应建交互关系数除去可不必建或已建的交互关系, 即为式(3).

综合定理 1 和定理 2 即可得出 s_i 的分配对上一步分配结果模型而言的新增交互关系总数, 从而为执行团队的选择提供依据, 这将在 4.3 节中具体讨论.

以上分析主要讨论在 s_i 之前分配过的子问题对分配 s_i 产生的影响, 除此以外, 还应适当考虑在 s_i 之后分配的子问题与当前 s_i 分配之间的关系.

定理 3　若子问题 s_i 分配给某首选候选团队 $CANDTEAM$ 执行, 则该候选团队的未来免建题外交互关系度计为

$$\sum_{i'=i+1, D(i', i)=1}^{m} \max_r |CANDTEAM \cap (C_{i', t'_{\min}})_r| \quad (4)$$

其中 $C_{i', t'_{\min}}$ 表示与 s_i 有依赖关系的子问题 $s_{i'}(i' = i+1, \dots, m)$ 的首选候选团队集合, 而 $(C_{i', t'_{\min}})_r$ 代表 $s_{i'}$ 第 r 个首选候选团队. 所谓未来免建题外交互关系, 即在将来分配 $s_{i'}$ 时可以不必建立的题外交互关系.

证明　在分配某个 $s_{i'}$ 时, s_i 的每一个执行队员都是 $s_{i'}$ 对应 $PTOINT$ 中一员. 类似定理 2 中第一种情况, 如果 $s_{i'}$ 第 r 个

首选候选团队中正好含有 s_i 的某个执行队员 p_j，且该候选团队即为执行团队，则身为 s_i 执行队员的 p_j 无需与同时身为 $s_{i'}$ 执行队员的自己建立对应于 s_i 和 $s_{i'}$ 的题外交互关系，即产生一个免建题外交互关系。进而，若该候选团队中有若干个成员都同时是 s_i 的执行队员，则将出现 $|CANDTEAM \cap (C_{i', t'_{\min}})_r|$ 个免建交互关系。又由于某候选团队成为 $s_{i'}$ 执行团队的概率均等，而 $s_{i'}$ 只能在若干候选团队中选定一个执行团队，故对此 $s_{i'}$ 而言，该 $CANDTEAM$ 引起的免建交互关系度应为与候选团队交集的最大势。从而，对于所有与 s_i 有依赖关系且尚未分配的子问题而言，s_i 的某个候选团队 $CANDTEAM$ 造成免建交互关系的度之和为式(4)所示。

选择原则 2：未来免建题外交互关系度最大的候选团队优先，以减少未来分配时可能产生的交互关系。

4.3 递阶优化分配算法

定理 4 子问题 s_i 分配之后相对上一步结果模型中的交互关系改变量可计算如下

$$INCREASEDINT = INTRAINT + EXTRAINT - \sum_{j_1=1,\ R(i,j_1)=1}^{n-1} \ \sum_{\substack{j_2>j_1,\ R(i,j_2)=1,\\ Int_{i-1}(j_1,j_2)=0}}^{n} I(i,j_1) \vee I(i,j_2) \tag{5}$$

其中 $INTRAINT$, $EXTRAINT$ 分别由定理 1, 定理 2 获得。

证明 新增的实体之间的交互关系要么对应某个题内交互关系，要么对应某个题外交互关系，或者同时对应两种交互关系。如果同时对应两种交互关系，则应除去重复计数部分，从而可得最终新增交互关系数如式(5)所示。

选择原则 3：选择造成 $INCREASEDINT$ 最少的候选团队作为执行团队。

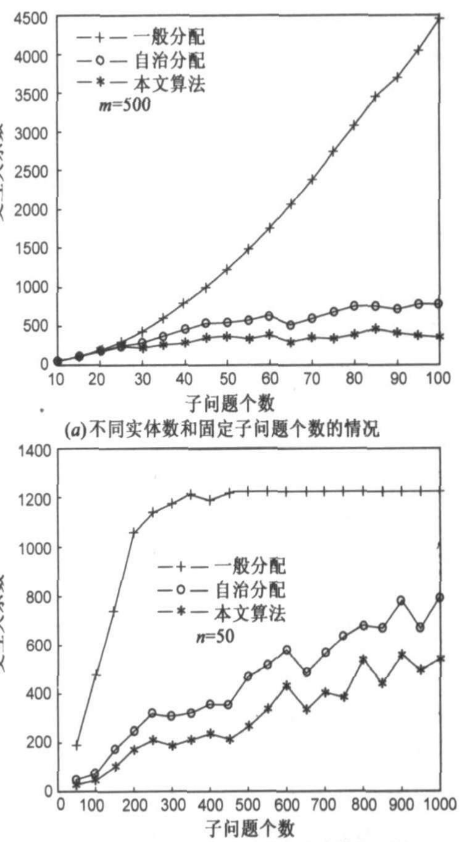
```
为各子问题确定首先候选团队集合
                    //根据选择原则 1 选择有最小 t 的候选团队
为各子问题确定分配顺序  //最少首选候选团队的子问题优先
for 1 to m{          //i 表示第 i 个被分配的子问题
  对每个首选候选团队{
        根据定理 1 计算 INTRAINT
        根据定理 2 计算 EXTRAINT
        根据定理 4 计算 INCREASEDINT
  }
  根据选择原则 3 找出最小 INCREASEDINT
  if 不只一个候选团队对应最小 INCREASEDINT {
        根据定理 3, 计算这些候选团队的未来免建交互度
        根据选择原则 2 从中选择执行团队
  }
  else
        将对应的唯一候选团队作为执行团队
  修改结果模型(EXES-P, INTP-P)
}
```

图 3 协作式问题分配算法描述

除了给出上述 3 个原则以对单个子问题的候选团队进行选择，还须给定 m 个子问题的分配顺序。由于对某子问题的分配需要充分利用已有的分配结果，如果让选择空间大的多候选团队子问题在前分配，则可以利用的已有交互关系有限，该子问题即便有很大的候选团队选择空间也无法充分利用选择原则进行选择；而后分配的少候选团队子问题虽然有更多的已有交互关系可以利用，但由于其自身选择空间小，并不能很好地利用已有结果模型的大量信息，造成资源浪费，这样，选择原则始终得不到有效的利用。故而，以每个子问题首选候选团队集合的势为分配顺序的依据，越小者越先分配，以利于选择原则的有效发挥。具体算法见图 3。

5 实验结果和讨论

就算法性能而言，以 u 表示子问题候选团队的最大个数，m 为子问题数，n 为实体数，则本文递阶优化算法的时间复杂性可表示为 $O(m^2u^2n + mun^2)$，为多项式时间算法，对复杂协作系统的分配问题具有较高的实践意义和实用价值。实验情况表明，对于 500 个子问题 50 个实体的协作系统，在 RAM 为 512MB，处理器为 Pentium4 的电脑上模拟多次分配过程的平均运行时间为 100ms 左右，能够满足一般应用要求。



(a) 不同实体数和固定子问题个数的情况

(b) 不同子问题个数和固定实体个数的情况

图 4 递阶优化分配方法的实验结果

就算法效果而言，本文对若干随机确定的待分配协作式问题分别应用本文算法和其它方法进行分配，以比较它们在交互关系优化上的优劣。实验结果如图 4 所示，其中一般方法的分配策略实为将子问题随机分配给某一候选团队，而自治分配为文献[12]所用算法。对相同的协作式问题，使用本文设计的递阶优化分配方法的交互关系数明显比另两种方法更优。如图 4(a) 中，对于有 50 个实体参与的由 500 个子问题组

成的某协作式问题而言, 使用一般分配方法将出现 1225 个交互关系, 自治分配为 544 个, 而用本文方法则仅需在各实体之间建立 366 个交互关系.

实验结果证明, 递阶优化分配方法在很大程度上减少了交互关系的数量, 这对复杂协作式问题的有效高质解决具有重大意义. 如传感器网络中将各子问题分配给若干传感器时, 由于传感器受能源约束较大, 其中传感器定位等通信开销是能源损耗的主要途径之一, 故应用本文分配方法以大幅减少交互关系数将对整个传感器网络的能源优化起到重要作用, 而这种优化策略在一般负载平衡分配方法中则少有提及.

6　总结和展望

随着计算机应用和网络技术的不断进步, 跨领域、跨地域的复杂协作式问题有着越来越广阔的发展空间, 也相应产生了许多有价值的研究课题, 其中, 协作式问题日趋复杂的交互行为对问题的分配研究提出了新的挑战. 本文针对交互复杂的大型协作式问题, 如软件开发、协作认知、协商设计等等, 从优化交互关系模型的角度进行分配, 推出了三个选择原则分别从子问题被解决的自治性, 当前分配引起的题内、题外交互关系增量以及前瞻性的未来免建交互关系预测这三个方面对某一子问题的各候选团队进行衡量和选择, 并在此基础上设计实现了递阶优化分配算法, 尽可能的将多实体之间的交互处理行为转化为单个实体的自治处理行为, 减少了实体之间的交互关系数, 从而紧缩了实体的交互工作量, 简化了协作管理行为, 降低了通信代价, 对有效高质地解决此类复杂协作式问题具有重要意义.

在进一步研究工作中, 将加入对一些约束条件的处理, 比如对各实体的负载平衡要求等等. 另外, 考虑到不同交互关系的实现可能对应着各实体不同的交互处理负载以及相互间不同的通信开销, 可以根据实际情况对各交互关系赋予不同权重值, 在本文研究基础上对加权交互关系模型进行优化以获取更优解.

参考文献:

[1] Uwe M Borghoff, J H Schlichter. Computer-Supported Cooperative Work: Introduction to Distributed Applications[M]. New York: Springer Verlag, 2000.

[2] Rogers, Erika. Cognitive cooperation through visual interaction[J]. Knowledge-Based Systems, 1995, 8(2-3): 117-125.

[3] 冯国奇, 尹朝万, 王成恩. 产品协同设计中发言权控制机制的研究与实现[J]. 小型微型计算机系统, 2003, 24(12): 2321-2323.
Feng GQ, Yin CW, Wang CE. Research of floor control in product collaborative design[J]. Mini-Micro SystemS, 2003, 24(12), 2321-2323. (in Chinese)

[4] 毛国君, 王薇, 杨名生. 广义超立方体和它的任务分配问题[J]. 软件学报, 1998, 9(6): 419-425.
Mao GJ, Wang W, Yang MS. Extended hypercube and its task allocation[J]. Journal of Software, 1998, 9(6): 419-425. (in Chinese)

[5] Sarit Kraus, Tatjana Plotkin. Algorithms of distributed task allocation for cooperative agents[J]. Theoretical Computer Science, 2000, 242(1-2): 1-27.

[6] Chen WH, Lin CS. A hybrid heuristic to solve a task allocation problem[J]. Computers & Operations Research, 2000, 27(3): 287-303.

[7] 宾雪莲, 杨玉海, 金士尧. 一种有限优先级的静态优先级分配算法[J]. 软件学报, 2004, 15(6): 815-822.
Bin XL, Yang YH, Jin SY. An assignment algorithm of static priority for limited priority levels[J]. Journal of Software, 2004, 15(6): 815-822. (in Chinese)

[8] Ma YC, Chung CP. A dominance relation enhanced branch-and-bound task allocation[J]. Journal of Systems and Software, 2001, 58(2): 125-134.

[9] Ma YC, Chen TF, Chung CP. Branch-and-bound task allocation with task clustering-based pruning[J]. Journal of Parallel Distributed Computing, 2004, 64(11): 1223-1240.

[10] Menon S. Effective reformulations for task allocation in distributed systems with a large number of communicating tasks[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(12): 1497-1508.

[11] Bataineh SM. Toward an analytical solution to task allocation, processor assignment, and performance evaluation of network processors[J]. Journal of Parallel and Distributed Computing, 2005, 65(1): 29-47.

[12] Liu XP, Dou WC, Chen GH, Cai SJ, Tang JS. Autonomous-centered problem allocation oriented to cooperation[A]. Cao Jian-Nong et al. Lecture Notes in Computer Science 3756: Advanced Parallel Processing Technologies[C]. Berlin: Springer, 2005. 91-100.

[13] 越民义. 组合优化导论[M]. 浙江: 浙江科学技术出版社, 2001. 86.

作者简介:

刘茜萍　女, 博士研究生, 1981 年生于四川资中. 研究方向为协同计算、认知协作. E-mail: xixi_liu@graphics.nju.edu.cn

窦万春　男, 副教授, 1971 年生于江苏徐州. 研究方向为工作流技术、协同计算. E-mail: douwc@nju.edu.cn

蔡士杰　男, 教授, 1944 年生于江苏太仓. 研究方向为计算机图形学、CAD、图形识别. E-mail: sjcai@nju.edu.cn

唐加山　男, 副教授, 1968 年生于安徽天长. 研究方向为随机过程及其应用、排队论、信号与信息处理.