

文章编号:1001-9081(2010)07-1906-03

多 Agent 系统中基于招投标的任务分配优化

丁芝琴¹, 刘永², 王凯²

(1. 榆林学院 能源工程学院, 陕西榆林 719000; 2. 西安理工大学 机械与精密仪器工程学院, 西安 710048)

(wang_kenvin@gmail.com)

摘要: 在利用多 Agent 系统辅助生产任务分配过程中, 为避免仅凭招投标结果来确定任务分配方案时只能获得局部最优的问题, 提出了一种生产任务分配全局优化方法。建立了基于招投标结果的生产任务分配优化目标函数, 设计了退火进化算法, 实现生产任务的综合评标。通过实例验证说明算法求解该问题可行有效, 便于获得生产任务分配的全局最优方案。

关键词: 多 Agent 系统; 任务分配; 全局优化; 退火进化算法

中图分类号: TP18 文献标志码:A

Bidding-based optimization of task allocation in multi-Agent system

DING Zhi-qin¹, LIU Yong², WANG Kai²

(1. Institute of Energy Engineering, Yulin University, Yulin Shaanxi 719000, China;

2. Faculty of Mechanical and Precision Instrument Engineering, Xi'an University of Technology, Xi'an Shaanxi 710048, China)

Abstract: To avoid the problem of determining task allocation schema only by bidding results which obtain partial optimization in multi-Agent system assisted production task allocation, a method of global optimization was put forward. An objective function of production task allocation was established based on bidding results. Annealing evolution algorithm was designed to realize the synthesized bid evaluations. Instances indicate that this method is feasible and convenient to achieve global optimization of production task allocation.

Key words: multi-Agent system; task allocation; global optimization; Annealing Evolution Algorithm (AEA)

0 引言

生产任务分配是多 Agent 生产系统中生产任务规划过程不可缺少的环节, 该环节主要是由任务 Agent 和资源 Agent 通过招投标形式来完成。针对某个任务集合, 任务 Agent 向资源 Agent 发出招标信息, 资源 Agent 根据自身生产能力提供标书, 然后由任务 Agent 从所有投标资源选择完成某项任务的最优资源, 一般以完成时间和费用为主要考核因素来产生一种调度方案^[1-2]。然而, 通过该策略得到的调度方案往往并不一定是最优方案^[3-4], 特别是在没有专门管理 Agent 的系统中, 一般资源 Agent 和任务 Agent 不具备对整个计划调度问题的全局视点, 只能做到局部优化^[5]。所以在确定分配方案的过程中, 不能以某个投标资源完成单个任务最优为评定标准, 而应以多个任务总体完成效益最佳为目标。

因此, 本文以综合考虑影响生产任务分配的完成提前—延期惩罚和协调交互成本等因素, 对任务分配问题进行研究, 以实现生产任务分配的全局最优。

1 任务分配问题的描述

任务分配问题可以具体描述如下。

定义 1 任务集合 $T = (T_1, T_2, \dots, T_i, \dots, T_n)$, 其中 T_i 是任务集中的单个任务, 称之为任务元, $T_i = \{c_i, t_i, d_i, s_i\}$, n 为任务元数目。其中, c_i, t_i, d_i, s_i 分别表示任务元 T_i 的任务编码、要求交付时间、任务描述和任务状态。

收稿日期: 2010-01-28; 修回日期: 2010-03-05。

作者简介: 丁芝琴(1964-), 女, 陕西米脂人, 副教授, 博士, 主要研究方向: 机械系统性能分析、测试与控制; 刘永(1981-), 男, 山西忻州人, 博士, 主要研究方向: 网络制造、生产计划与过程控制; 王凯(1969-), 男, 河南漯河人, 副教授, 博士, 主要研究方向: 机电系统的性能分析、测试与控制。

定义 2 投标 Agent 集合 $A = (A_1, A_2, \dots, A_j, \dots, A_m)$, 其中 m 为投标 Agent 数目。

定义 3 标书矩阵 RT 记录所有投标 Agent 集合 A 对任务 T 的投标结果, 元素 rt_{ij} 为 A_j 对任务元 T_i 进行投标的标书信息, 则 RT 可表示为:

$$RT = T^T \cdot A = [T_1, T_2, \dots, T_n]^T \cdot [A_1, A_2, \dots, A_m] = \begin{bmatrix} rt_{11} & rt_{12} & \cdots & rt_{1m} \\ rt_{21} & rt_{22} & \cdots & rt_{2m} \\ \vdots & \vdots & & \vdots \\ rt_{n1} & rt_{n2} & \cdots & rt_{nm} \end{bmatrix} \quad (1)$$

$$rt_{ij} = T_i \cdot A_j = \begin{cases} \{c_i, ec_{ij}, et_{ij}\}, & A_j \text{ 有能力承担任务 } T_i \\ \emptyset, & A_j \text{ 无法承担任务 } T_i \end{cases}$$

其中: c_i 为 T_i 的任务代码; ec_{ij} 、 et_{ij} 分别表示 A_j 完成任务 T_i 消耗的成本和最晚完成时间。当选取 A_j 完成任务 T_i 时, 记 $g_{ij} = 1$; 否则 $g_{ij} = 0$, 并且对于任意一个任务只能由一个服务资源来完成, 即:

$$\sum_{j=1}^m g_{ij} = 1; \forall i \quad (2)$$

基于上述定义和描述, 以最小化任务执行成本为目标, 建立如下任务分配目标函数:

$$\min \left\{ \sum_{i=1}^n \sum_{j=1}^m g_{ij} (ec_{ij} + \alpha \cdot |t_i - et_{ij}|) - \beta \cdot \sum_{j=1}^m \lfloor \sum_{i=1}^n g_{ij}/2 \rfloor \right\} \quad (3)$$

其中: α 为提前/延期惩罚系数; β 为奖励系数; $\lfloor X \rfloor$ 符号表

示 X 向下取整。 α, β 均为不固定值,可以根据不同的目标需求,适当进行调整。

在式(3)的目标函数中,前面部分表示分配方案所消耗的成本,由执行费用和提前/延期惩罚费用组成,用来控制方案的综合成本;后面部分表示盟主企业与合作伙伴的协作奖励费用,目的是使承担任务的企业最少,以减少企业间的协调与交互。

本文所建分配问题的核心在于满足一定分配目标和约束的前提下,从标书矩阵 RT 中找出最优的分配方案 P 。对于该问题的求解,本文设计了下面的优化算法。

2 任务分配的退火进化算法

退火进化算法(A Annealing Evolution Algorithm, AEA)综合了模拟退火算法(Simulated Annealing, SA)和遗传算法(Genetic Algorithm, GA),优势互补,发挥了SA局部搜索能力和GA全局搜索能力,克服SA全局搜索能力差及效率不高的问题以及GA局部搜索能力差及其早熟现象。AEA把SA算法与GA结合在一起,通过变异与选择不断改善解群体,并行搜索解空间,从而有可能更迅速地找到全局最优解^[6-8]。

AEA算法的研究对象是由一个参数集所确定的某种配置。设计一个基于该配置的目标函数 f , f 的极小化过程模拟自然界的退火过程,由一个逐步冷却的温度 tmp 来控制。对每一个温度值尝试一定的极小化操作,如通过复制、变异、交叉等进化操作产生新一代群体。重复产生新群体,直到系统冷却不再产生更好的配置为止。可见,实现退火进化算法至少需要五个要素:1)某个系统配置的准确描述;2)新方案个体的随机产生机制;3)新一代个体的进化策略;4)一个定量的包含多种因素折中的目标函数;5)一个冷却机制 $tmp = \lambda \cdot tmp$, λ 称为收敛率, $0 < \lambda < 1$ 。

在对上述五个要素进行适当设计后,就可将其用于实际问题的求解。针对本文所涉及的任务分配方案优化问题,在上述问题描述模型的基础上,算法步骤设计如下:

步骤1 初始设置。定义初始分配方案群体 $D[G]$, G 为群体中个体规模数。 $D[g] = [rt_{1h}, rt_{2k}, \dots, rt_{nl}]$ 表示方案个体,其中 $1 \leq g \leq G, h, k, l \in [1, m]$ 。建立两个存储配置方案的群体 $D_0[G]$ 和 $D_N[G]$,分别代表原方案和新方案。设置起始温度值 tmp ,收敛率 λ 。

步骤2 随机产生配置方案的初始群体:产生初始群体 $D_0[G]$,其个体的产生方法如下:从任务集 T 中按顺序读取一个任务元 T_i ,在 RT 中找出对应任务元 T_i 的投标结果集 $RT_i = [rt_{i1}, rt_{i2}, \dots, rt_{im}]$,从中随机提取某个Agent A_j 的投标信息 rt_{ij} 作为该任务元的配置方案。如此反复提取直到所有任务元得到一个配置方案为止,其提取原理如图1所示。将结果保存到配置方案的集合 $D_0[g]$ 中,然后进行其他个体的生成。

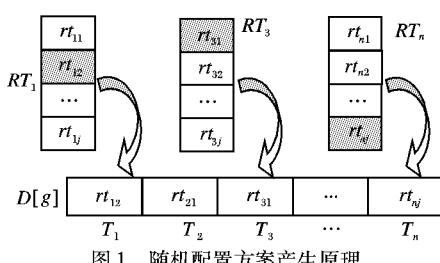


图1 随机配置方案产生原理

随机操作的程序伪代码如下:

```

For ( g = 1; g <= G; g ++ )
{
    For ( i = 1; i <= n; i ++ )
    {
        j = Rand( 1, RTi.size() );
        // 在 1 到 RTi 元素个数之间随机产生一个整数
        D[g][i] = rtij; // 为 D[g] 中第 i 个任务配置执行方案
    }
}

```

步骤3 计算群体中各个体的目标函数值。将群体 $D_0[G]$ 中的每个个体分配方案依次提取代入目标函数式(3)中,计算各自的函数值 $f(g)$,分别进行比较,找出最优的方案 $D_0[g']$ 。

步骤4 进行复制进化操作:复制最优方案 $D_0[g']$ 入新群体 $D_N[G]$,并按照下面的概率方法选择群体 $D_0[G]$ 中的个体复制到新群体 $D_N[G]$ 中。

由于 $D_0[g']$ 为最优方案,因此必然有 $f(g) > f(g')$, $g \neq g'$,此时计算概率值 $prob = \exp[-\Delta f/(k \times tmp)]$, $\Delta f = f(g) - f(g')$,这里的 k 是玻尔兹曼常数。然后在(0,1)上产生随机数 $rand$,如果 $prob \geq rand$,则将 $D_0[g]$ 复制到群体 $D_N[G]$ 中;如果 $prob < rand$,则复制 $D_0[g']$ 到 $D_N[G]$ 中,但要求 $D_0[g']$ 的个数不能超过 $G/5$ 。群体数不足时用随机方案补充。相应程序的伪代码如下。

```

int bstNm = 0;
For ( int g = 1; g <= G; g ++ )
{
    If ( g == g' )
        { D_N[g] = D_0[g']; }
    Else {
        Δf = f(g) - f(g');
        prob = exp[-Δf/(k × tmp)];
        rand = random(0,1);
        If ( prob ≥ rand )
            { D_N[g] = D_0[g]; }
        Else
            { If ( bstNm < G/5 )
                { D_N[g] = D_0[g']; bstNm ++; }
            else { 为 D_N[g] 配置随机方案; }
        }
    }
}

```

步骤5 对个体进行交叉繁殖操作。在新群体 $D_N[G]$ 中以一定的概率发生交叉繁殖,随机提取两个个体的配置方案,随机产生一个交换点,然后对交换点后的任务元的配置结果进行交换,建立起两个新的配置方案,完成所选两个个体的进化,如图2所示。

步骤6 对个体进行变异操作。在新群体 $D_N[G]$ 中以一定的概率发生变异操作,随机提取一个个体,改变其中某个任务元 T_i 的执行方案 rt_{ij} ,使该个体发生变异。

步骤7 退火降温。判断当前温度,使系统趋于冷却,令 $tmp = \lambda \cdot tmp$, $D_0[G] = D_N[G]$,此时如果 $tmp > 1$,则转到步骤3的操作;如果 $tmp < 1$,则进入下一步操作。

步骤8 任务分配定标。采用步骤3中的方法,评价最后一代群体中的配置方案,提取出最优的任务分配方案 $D_0[g']$,则该方案为所要求解的分配方案。

在复制操作过程中,主要考虑了 $prob$ 值对方案配置的影响。

响; k 是常数, 在一定温度下进行方案选择时, tmp 也可以看做常数, 所以 $prob$ 只与 Δf 有关, 而且随 Δf 递增而递减。 Δf 越小, 表明某个体方案虽次于最优个体方案, 但接近最优个体, 因而是较“好”的方案, 而此时 $prob$ 也越大, $prob \geq rand$ 成立的可能性也越大, 越容易被接受。当 $prob < rand$ 时直接复制最优

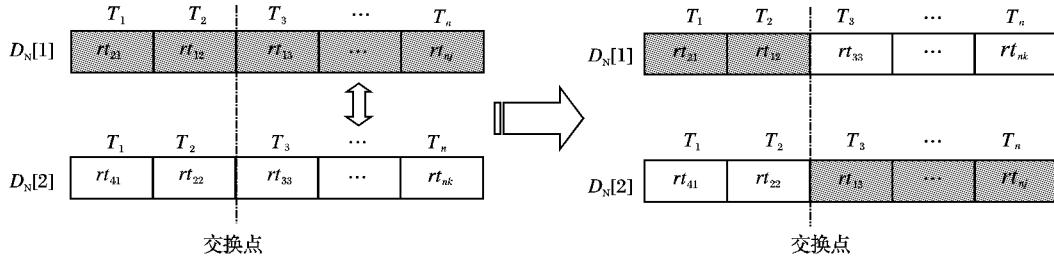


图 2 交叉繁殖示意图

3 实例分析

下面通过一个实例来说明采用退火进化算法来评标的优劣性。该实例是 7 个外协任务 (T_1, T_2, \dots, T_7), 由 6 个服务资源 Agent (A_1, A_2, \dots, A_6) 参与投标的优化评标问题。经过投标, 得到下面一组数据, 所有任务的执行费用 (ec_{ij}) 和最迟完成时间 (et_{ij}) 见表 1。

表 1 任务投标结果

任务 T_i	交付时间 t_i	Agent 投标参数 ec_{ij} / et_{ij}					
		A_1	A_2	A_3	A_4	A_5	A_6
T_1	10	20/8	—	24/9	17/11	—	17/10
T_2	12	28/12	34/12	—	—	30/14	—
T_3	8	32/9	33/9	—	25/11	30/9	—
T_4	20	—	—	27/19	—	—	30/20
T_5	16	—	18/13	20/11	23/11	—	15/16
T_6	9	—	15/6	14/10	—	16/9	13/6
T_7	13	19/6	—	13/15	15/13	18/12	—

利用本文设计的算法, 选取参数初始值为 $tmp_0 = 1000$ 、 $\alpha = 5$ 、 $\beta = 10$ 、 $\lambda = 0.9$ 、 $G = 10$ 、交叉概率为 0.9、变异概率为 0.1, 得到的最优分配方案如表 2 所示(目标函数 f 为 131)。

表 2 优化评标结果及执行参数

任务	中标 Agent	投标参数
T_1	A_4	17/11
T_2	A_1	28/12
T_3	A_5	30/9
T_4	A_6	30/20
T_5	A_6	15/16
T_6	A_5	16/9
T_7	A_4	15/13

通过采集不同退火温度下的目标函数值, 得到如图 3 所示的目标函数变化曲线。从其变化趋势同普通模拟退火算法相比可以看出, 退火进化算法有效地克服了普通模拟退火算法波动较大、收敛性较差以及遗传算法收敛速度慢的缺陷。图 3 表明该算法用于任务分配求解的可行性和有效性, 并在求解过程中表现出良好的稳定性。

4 结语

本文重在提出实现生产任务分配的全局优化方法, 以克

个体到下一代。可以看出, 结合了模拟退火算法原有的方案保留规则和遗传算法中择优选取的原则, 使下一代群体中既存在最优秀的个体, 又保留了具有进化潜力的次优个体, 从而使下代群体的整体水平优于原群体, 因此更有利于最优方案的快速出现。

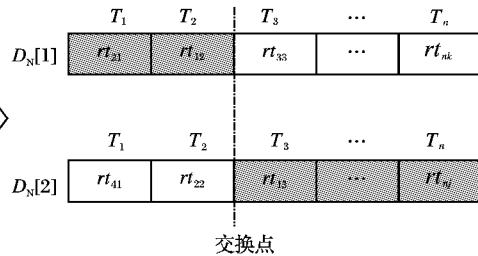


图 3 目标函数值变化趋势

服仅凭招投标结果来判定生产任务承担者时只能获得局部最优的缺陷。通过构建任务执行成本最小化的函数, 确立任务分配优化目标。简单起见, 文中仅考虑了影响任务执行成本的部分因素, 结合问题规模和复杂程度, 设计了适合求解该问题的退火进化算法, 并给出了详细的求解步骤。通过实例验证, 表明所提方法可行有效。

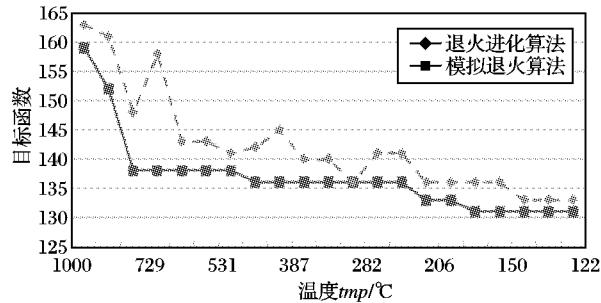


图 3 目标函数值变化趋势

参考文献:

- [1] LU TAPING, YIH Y. An agent-based production control framework for multiple-line collaborative manufacturing[J]. International Journal of Production Research, 2001, 39(10):2155–2176.
- [2] LI SHUJUAN, LI YAN, LIU YONG, et al. A GA-based NN approach for makespan estimation[J]. Applied Mathematics and Computation, 2007, 185(2):1003–1014.
- [3] 何炎祥, 陈莘萌. Agent 和多 Agent 系统的设计与应用[M]. 武汉: 武汉大学出版社, 2001.
- [4] JUN BIN. Research and realization of production scheduling system based on multi-Agent mechanism[J]. Computer Technology and Automation, 2002(9):105–108.
- [5] VANCZA J, MARKUS A. An Agent model for incentive-based production scheduling[J]. Computers in Industry, 2000, 43(2):173–187.
- [6] 朱福喜, 何炎祥. 并行分布计算中的调度算法理论与设计[M]. 武汉: 武汉大学出版社, 2003.
- [7] 戴上平, 高丽, 朱长武. 基于遗传模拟退火算法的任务分配与调度[J]. 广西师范大学学报: 自然科学版, 2006, 24(4):151–154.
- [8] LEVITIN G, RUBINOVITZ J. A genetic algorithm for robotic assembly line balancing[J]. European Journal of Operational Research, 2006, 168(3):811–825.