

CPU 时间片组合拍卖遗传算法

刘爱珍¹, 王嘉祯¹, 张西红¹, 贾红丽²

(1. 军械工程学院计算机工程系, 石家庄 050000; 2. 军械工程学院装备保障工程系, 石家庄 050000)

摘要: 提出一种综合考虑 Agent 时间片和执行截止期限要求的 CPU 时间片组合拍卖遗传算法。该算法定义了问题模型, 采用可以去除不具备竞争力标的预选择策略, 减少遗传算法的计算复杂度。在遗传算法求解过程中, 设计适合该问题的变长染色体编码方式、交叉算子、换序算子和换标算子。仿真实验结果表明, 该算法取得了符合要求的最优解。

关键词: 组合拍卖; 遗传算法; 截止期限; CPU 时间片

Genetic Algorithm for CPU Time Slice Combinational Auction

LIU Ai-zhen¹, WANG Jia-zhen¹, ZHANG Xi-hong¹, JIA Hong-li²

(1. Department of Computer Engineering, Ordnance Engineering College, Shijiazhuang 050000;

2. Department of Equipment Engineering, Ordnance Engineering College, Shijiazhuang 050000)

【Abstract】 This paper proposes a Genetic Algorithm(GA) for CPU time slice combinational auction in which both Agent time slice demand and execution deadline demand are considered. The problem model is defined, and the preselecting method is adopted to filter non-competitive bid to decreasing computing complexity of genetic algorithm. Variable chromosome coding, crossover operator, adjusting-bid-order operator and replacing-bid operator are designed to solve the problem in genetic algorithm. Experimental results show that this algorithm can obtain satisfaction optimal solution.

【Key words】 combinational auction; Genetic Algorithm(GA); deadline; CPU time slice

计算机及网络技术的快速发展使各种相关设备被广泛使用, 其中包括低能力计算机设备。资源共享尤其是计算资源、信息资源的共享成为必然。移动 Agent 是一种新型分布式范型, 其可移动性、自主性等特点可以优化对资源的使用, 但也可能导致资源滥用。由于 Agent 代表其创建者, 即用户的利益, 因此其目标是用最短的时间获得最多资源。Agent 不关心执行主机的利益, 移动 Agent 系统为保证自己的利益, 必须采取有效的资源分配机制, 以控制 Agent 对其资源的使用。研究有效的资源分配方法, 尤其是 CPU 时间片资源分配方法具有重要意义。

1 CPU 时间片调度策略

当前 Agent 平台的 CPU 时间片调度策略主要包括 3 种: (1) 传统资源分配调度策略, 如先来先分配策略(Aglet 平台); (2) 以保护主机资源不被滥用或非授权使用为目的的资源分配调度策略^[1]; (3) 引入市场机制和拍卖机制的调度策略^[2-6]。在第 (3) 种策略中, 移动 Agent 一般以电子货币来购买(交换)对资源的使用权限或数量。文献[4-6]用第一价格密封拍卖协议对若干 CPU 时间片进行拍卖分配, 并用动态规划方法求得以近似最大化收入为目标的资源分配方案。该方法 Agent 投标的特点是对时间片执行完毕截止时间的要求相同, 即在每次拍卖的 CPU 时间片内完成, 没有考虑对截止期限有各自具体要求的 Agent 投标情况。一些 Agent 对执行时间有严格要求, 必须在某个截止期限前执行完毕, 因此, 本文研究了 Agent 具有最大截止期限要求的 CPU 时间片组合拍卖问题。

2 CPU 时间片组合拍卖问题

假设每轮待拍卖的 CPU 时间片为 $M=mT$, 其中, T 表示一

个时间片单元; m 为正整数。假设有 N 个 Agent 要投标, N 个投标方的时间片总和总是大于待拍卖的时间片总和。任意 Agent i 可以根据一定原则(如用户需要、对当前环境的估计)确定所需时间片数量 s_i ($s_i \leq M$ 且是 T 的整数倍)、 s_i 的最大允许执行截止时间 d_i ($d_i \leq M$) 及其基于上述要求的出价 $b_i(s_i, d_i)$ 。投标时, 拍卖方根据收益最大化原则, 确定最后的竞胜标, 以最大化拍卖方的收益为目标, 采用密封标组合拍卖协议, 计算公式如下:

$$\max_{X \in A} \sum_{b_i(s_i, d_i) \in X} b_i(s_i, d_i) \quad (1)$$

其中, X 为一个可行解, 满足每个时间片至多可分配给一个标及每个标的截止期限要求; A 为可行解空间, 由所有可行解 X 组成。可行解确定的竞胜标的时间片总和小于等于 mT 。

上述问题形式简单, 但属于 NP 难问题^[5]。与文献[5]组合拍卖只考虑自身利益及保证每个时间片只卖一标的前提相比, 基于 Agent 最大截止期限的 CPU 拍卖还必须考虑投标者的最大允许截止期限问题。因此, 其求解过程更复杂, 很难用传统算法求解。针对上述问题, 本文设计了一种求解该组合拍卖问题的遗传算法。

3 CPU 时间片组合拍卖遗传算法

CPU 时间片组合拍卖遗传算法分为 2 个阶段: (1) 预选择

基金项目: 国家自然科学基金资助项目(60672143); 河北省科技攻关基金资助项目(052435179D)

作者简介: 刘爱珍(1972-), 女, 博士研究生, 主研方向: 移动 Agent 智能算法; 王嘉祯, 教授; 张西红, 副教授、博士后; 贾红丽, 副教授、博士

收稿日期: 2008-01-22 **E-mail:** liuaizhen_163@163.com

阶段,主要目的是采用各种方法,去除不具竞争力的投标者,以减少运算空间;(2)使用 GA 对第(1)阶段得到的投标方集合进行运算,得到最优解或近似最优解。

3.1 预选择策略

在截止期限 d_i 相同的各个标中,对 $s_k > d_i/2$ 的标,每个 s_k 只取价格最高的标;对 $d_i/3 < s_k \leq d_i/2$ 的标,每个 s_k 只取2个出价最高的标;对 $d_i/4 < s_k \leq d_i/3$,每个 s_k 至多选取3个出价最高的标,依此类推,当 $s_k = T$ 时,至多选取 d_i 个出价最高的标。“至多”一词的意义在于,当 d_i 越大、 s_k 越小时,除非其他 s_k 较大的标都不被选择,而只选该标,才会需要选取 d_i/s_k 个投标方,此情况一般不会出现。因此,为减少搜索空间,可以只选取部分出价较高的标。假设共有 n 个Agent参与投标,经预处理后选出 m 个有竞争力的标。 m 越小,应用GA时的搜索空间越小。

3.2 遗传算法设计

3.2.1 染色体编码

采用直接编码法,基因表达采用变长表示法。染色体中的基因构成一个可行解。每个基因对应不同投标者的标号 i (i 代表的投标方为 $b_i(s_i, d_i)$)。由于投标者对时间片的需求不同,因此染色体中的基因数不固定,即染色体长度可变。假定经过3.1节所述预处理后,选出 m 个投标方,记为 $B = \{1, 2, \dots, i, i+1, \dots, m\}$,可用式(2)表示一个染色体。

$$X = (1, 4, 7, i, i+1, \dots, j) \quad (2)$$

其中, $i \in \{k | k \in B, s'_k = s_k, d'_k = d_k\}$,且 X 满足 $s_1 + s_4 + s_7 + s_i + s_{i+1} + \dots + s_k \leq M$, s'_k, d'_k 分别为 X 分配给标号为 k 的Agent的时间片和实际执行截止期限。

3.2.2 初始种群的生成

初始种群的生成步骤如下:

(1)随机选取投标方 $i \in B$ 作为首基因,解选中的第1个标为 i ,此时 $X = (i)$ 。

(2)若 $\sum_{i \in X} s_i < M$,则寻找满足 $s_k \leq M - \sum_{i \in X} s_i$ 和 $d_k > \sum_{i \in X} s_i$ 的标集合 U 。如果 $U = \emptyset$,则转步骤(5),否则转步骤(3)。

(3)随机选取一个标 $k \in B$,若满足 $d_k \leq \sum_{i \in X} s_i + s_k$,则将其加入到 X ,此时 $X = (i, k)$,计算此时解 X 的时间片数量并判断是否满足 $\sum_{i \in X} s_i = \sum_{i \in X} s_i + s_k$,满足则转步骤(2),否则转步骤(4)。

(4)从 U 中删除标 k ,若 $U = \emptyset$,则转步骤(3)继续寻找满足要求的标,否则转步骤(5)。

(5) $U = \emptyset$ 表明已不存在满足条件的标,解生成完毕。

(6)重复步骤(1)~步骤(5),直至达到种群规模。

由上述过程可知,染色体长度由依次选取的投标方的截止期限和时间片数量和共同决定。

3.2.3 适应度函数

因为目标函数用于求拍卖收益最大值,所以染色体的适应度函数定义为 $F(X) = k \times f(X)$,其中, $f(X)$ 为目标函数值; $0 < k \leq 1$ 为定标参数。

3.2.4 交叉算子

由于解用变长染色体表示,且每个基因对应的时间片不同,因此应用常规的PMX, CX, OX等交叉算子很难产生合法解。为使杂交算子生成合法解,对于随机选择的2个个体 P_1, P_2 ,若其染色体长度均大于1,则使用如下交叉算子:

(1)随机选定一个基因位 r ($1 < r < \min(\text{len}(P_1), \text{len}(P_2))$)。

(2)分别计算 P_1, P_2 在基因位 r 之前的标的时间片数量和 $L_1,$

L_2 ,假设 $L_1 > L_2$ 。

(3)计算 $L' = L_2 - L_1$ 。

(4)若 $L' = 0$,则分别查找 P_1, P_2 的基因位 r 之后的标是否在另一个染色体的前 r 个基因位出现,若无重复标,则交叉后的新染色体 P'_1 等于原染色体 P_1 的前 r 个标加原染色体 P_2 的 r 之后的标。若 P'_2 等于原染色体 P_2 前 r 个标中的 x 个标加原染色体 P_1 的 r 之后的标,则转(6),否则,转(5)。

(5)交叉后的新染色体 P'_1 的生成过程如下:

1)以 P_1 的前 r 个标作为新染色体 P'_1 的前 r 个标;

2)将染色体 P_2 的 r 之后的标集中,将不重复出现且满足截止期要求和时间片数量要求的标,依次加入新染色体 P'_1 ;

3)计算此时新染色体 P'_1 的时间片数量和 L'_1 ;

4)若 $L'_1 < M$,则在未被 P'_1 调度的标集合中寻找 $d > L'_1$ 且 $s \leq M - L'_1$ 的标集合 U 。如果 $U = \emptyset$,则从 U 中随机选取一个标 $b_k(s_k, d_k)$,若满足 $d_k \leq L'_1 + s_k$,则将 b_k 加入到 P'_1 中,计算此时的时间片数量和 $L'_1 = L'_1 + s_k$,并循环执行此过程,否则转(6)。

(6)输出新染色体 P'_1 ,结束。

新染色体 P'_2 的生成过程类似。

本文使用的交叉算子包括如下2种:

(1)换标算子

对换标算子描述如下:

针对选定解的所有基因位,寻找时间片数量相同但出价的标集,对标集中的标按以下规则进行处理:

1)若截止期大于当前标的截止期,则替换。

2)若截止期小于当前标的截止期,但如果替换当前标,可满足其截止期要求,则替换。

应用换标算子可使拍卖方的收益变大。

(2)换序算子

对换序算子描述如下:

从选定解的标中,寻找时间片数量相同的标集,若标数大于1,就将它们按各标对应的截止期限从小到大排序,若解的这些标对应的基因位置与此排序不符,则按此顺序替换各标在解中的基因位。

换序算子不会改变选中的标,只改变标的选中顺序,而在满足解的各标截止期要求的前提下,放宽整个解的截止期要求,进而放宽应用其他算子的限制。

3.2.5 选择策略

采用精英保留策略,保留上代 u 个最优个体,并从 u 个个体+本代 n 个个体中,选择 n 个较好的个体作为下一代种群,以加速解群的收敛。

3.2.6 收敛条件

若连续5代最优解没有进化,则随意加入2个新解,替代种群中的最差解,继续进化,若连续3次加入新解,种群最优解依然没有进化,则认为解已收敛,退出。

3.2.7 本文算法中的GA流程

本文算法中的GA流程如下:

(1)按3.2.2节生成一个规模为 N 的初始种群。

(2)评价初始种群,记录 u 个最优个体。

(3)置群体代数 $g = 1$ 。

(4)依次实施交叉算子、换标算子、换序算子,初步得到新一代种群。

(5)评价新一代种群的适应度,将上代保留的 u 个最优个体加入本代群体。从 $u + N$ 个个体中,按适应度从大到小依次

选择 N 个个体作为确定的新一代群体，并记录 u 个新一代最优个体。

(6) 检验收敛准则，若满足则转(7)，否则 $g=g+1$ ，转(4)。

(7) 输出最好的个体。

4 实验分析及结论

为检验本算法的性能，通过 Matlab 仿真进行 4 组共 6 个实验，每个实验分别运行 20 次。为了检测种群大小对算法性能的影响，第 2 组、第 4 组除种群规模不同外其余参数相同。所有实验的各投标方产生的投标矩阵形式为(时间片数量，截止日期，出价，标号)，各标的时间片数量为 $[1, M]$ 范围内的随机整数，截止期限为 $[$ 各标的时间片数量, $M]$ 范围内的随机整数(该范围可保证各标的最大截止时间不小于其所投时间片数量)，出价空间为 $[2, 5M]$ 范围内的随机整数。标号设为由到达次序确定的整数。其余参数设置如表 1 所示。GA 各算子经多次实验，确定交叉概率为 1，换标算子的操作概率为 0.8，换序算子的操作概率为 0.8，定标参数 $k=1$ 。

表 1 实验参数设置

序号	实验组别	拍卖时间片数 M	投标人数 N	种群规模	指定运行代数	备注
1	1	10	10	10	200	
2	2-1	10	50	10	200	实验投标矩阵相同
3	2-2	10	50	20	200	实验投标矩阵相同
4	3	50	50	10	200	
5	4-1	50	200	10	500	实验投标矩阵相同
6	4-2	50	200	20	500	实验投标矩阵相同

表 2 给出了 6 个实验的运行结果，根据每个实验运行 20 次得到的最优解、最差解、平均最优解、标准差、到指定运行代数的平均运行时间以及获得最优解时的平均运行代数等参数来评价本算法的性能。图 1~图 3 描述了前 2 组实验随机产生的初始种群最优解、最差解、平均最优解随进化代数变化的全过程。

表 2 各组实验的综合结果

序号	最优解	最差解	平均最优解	标准差	平均运行时间/s	预选后标数	平均运行代数	FCFS 解
1	115	115	115.0	0.0	0.40	10	1	31
2	178	169	170.9	3.7	0.50	31	25	47
3	178	169	172.2	4.4	0.92	31	25	47
4	1505	1467	1501.2	11.7	0.88	48	38	75
5	2237	1724	1952.3	119.9	4.06	178	160	147
6	2073	1820	1966.6	66.9	7.36	178	142	147

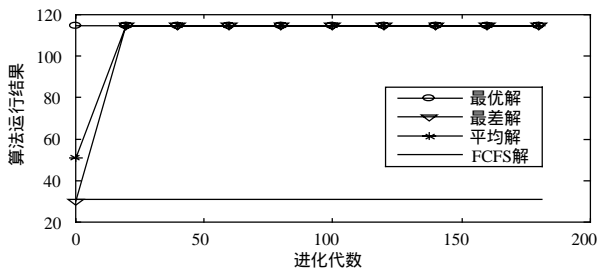


图 1 第 1 组实验各个解的进化曲线

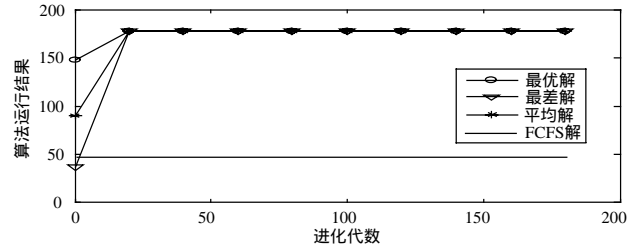


图 2 第 2-1 组实验各个解的进化曲线

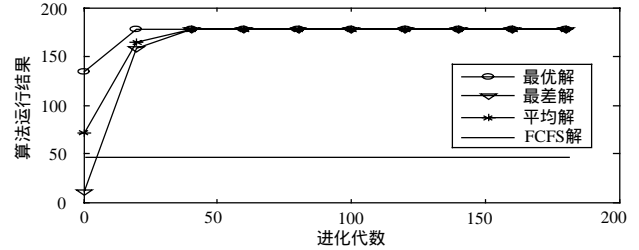


图 3 第 2-2 组实验各个解的进化曲线

先来先服务(FCFS)算法是传统操作系统常用的调度方法之一，表 2 给出了该算法的解(收益)。该算法按投标方 Agent 的到达顺序依次调度，而非按收入最大化原则确定执行哪些 Agent，因此，求得解是 M 个时间片内按到达顺序执行的、满足 Agent 要求的 Agent 的出价之和。

由表 2 和图 1~图 3 可以看出，本文算法收敛速度快、鲁棒性强，在 4 组不同规模的实验中均表现出良好的适应性，其解的质量明显优于传统 FCFS 算法。由表 1 中 2-1、2-2、4-1、4-2 的结果及图 2、图 3 可以看出，种群规模对本文算法所得最优解的影响不明显，因此，种群规模取 10~20 之间均可。笔者在实验中发现，在规模较小的第 1 组实验的 20 次实验中，第 2 次实验在种群进化了约 10 代后，获得了最优解，其余 18 次实验均在初始种群就得到了最优解，即问题的理论最优解(图 1)。各组实验结果表明，本文算法所得解的质量很高。

FCFS 算法一般不计算法运行开销，而本文算法根据收益最大化原则来确定投标方的调度顺序，因此，需要运行开销。本算法使拍卖方可以高效地解决 Agent 有截止期限要求的时间片组合拍卖问题。

参考文献

- [1] Manoj L, Paney R. A Scheduling Scheme for Controlling Allocation of CPU Resources for Mobile Programs[J]. Autonomous Agents and Multi-Agent Systems, 2002, 5(1): 7-43.
- [2] 张凡, 房鼎益. 基于市场机制的移动 Agent 系统资源管理[J]. 西北大学学报: 自然科学版, 2003, 33(6): 649-652.
- [3] Wellman M P, MacKie M J K. Exploring Bidding Strategy for Market-based Scheduling[C]//Proceedings of the 4th ACM Conference on Electronic Commerce. California, USA: ACM Press, 2003.
- [4] Jonathan L B. Market-based Control of Mobile-Agent Systems[D]. Hanover, USA: Dartmouth College, 2002.
- [5] Reggers B. Markets for Computational Resources[D]. Maastricht, Holland: Universiteit Maastricht, 2004.
- [6] 刘爱珍, 王嘉祯, 彭德云, 等. 一种高效的基于拍卖背包机制的移动 Agent 调度策略[J]. 计算机应用研究, 2007, 24(6): 52-54.