

# 移动 Agent 多任务调度算法

刘爱珍, 王嘉祯, 张西红, 陈立云

(军械工程学院计算机工程系, 石家庄 050000)

**摘要:** 提出一种可覆盖全部解空间的移动 agent 多任务分配与调度混合遗传算法。给出问题模型及染色体表示方法, 采用禁忌表加随机算法生成初始种群, 设计新的交叉机制保证交叉进化解的合法性。为促进算法的收敛, 变异个体使用禁忌及任务均衡启发变异算子。还采用保持解的不降性的最佳个体保留策略。2 种任务节点、3 种通信代价、3 种主机节点共 18 组图的仿真结果表明该算法进化的最优解较标准遗传算法有 37.1% 的平均改进量。

**关键词:** 移动 Agent; 多任务调度; 遗传算法; 交叉机制

## Mobile Agent Multi-task Scheduling Algorithm

LIU Ai-zhen, WANG Jia-zhen, ZHANG Xi-hong, CHEN Li-yun

(Department of Computer Engineering, Ordnance Engineering College, Shijiazhuang 050000)

**【Abstract】** A hybrid genetic algorithm is proposed to search the optimal solution of mobile agent multi-task matching and scheduling problem. The problem model and chromosome representation are defined. The initial population is generated by the taboo search and random selecting method. And a new crossover mechanism is designed to make the new scheduling evolved by crossover mechanism valid. To accelerate the algorithm convergence, taboo search and tasks load mutation operator is adopted. Also the best chromosome preserving strategy is adopted to keep optimal solution non-decreasing performance. Simulation results of 18 graphs combined by 2 kinds of task nodes, 3 kinds of communication costs and 3 kinds of host nodes show that the algorithm can get 37.1% improvement compared with standard genetic algorithm.

**【Key words】** mobile Agent; multi-task scheduling; Genetic Algorithm(GA); crossover mechanism

### 1 概述

移动 Agent 是一种新型的分布式应用, 由多个有先后约束关系和不同数据传输量的任务组成, 通过在异构网络的主机间迁移来寻找适合其各个任务所需数据及资源的主机来执行。由于满足各任务执行条件的主机往往有多个, 因此在异构网络环境下, 研究如何在最短的时间内分配及完成所有任务, 即研究移动 Agent 的多任务分配与调度算法, 对移动 Agent 的执行效率乃至系统有着直接的影响<sup>[1-2]</sup>。

目前, 人们对移动 Agent 的多任务调度算法进行了大量研究, 尽管研究前提不一, 但都采用传统的计算方法<sup>[1-2]</sup>。移动 Agent 任务静态调度问题是一个 NP 完全问题, 不可能在多项式时间内找到问题的最优解<sup>[2]</sup>。

本文研究用遗传算法(GA)求解该问题。GA 是模拟生物界自然进化和遗传过程的随机搜索算法, 具有在空间求解问题近似最优解的能力, 已经应用于同构或异构系统中的任务分配与调度<sup>[3-7]</sup>。但是由于多任务调度问题既要满足任务间的约束关系, 又要满足任务的唯一性和完整性, 因此各种常规 GA 难以直接应用。

文献[3-5]皆采用按任务高度子集的方法生成种群及进行各种遗传操作, 只能覆盖部分的解空间, 对于那些最优调度不符合高度排序的情况, 则无法求得最优解。

文献[6-7]均在全部解空间求解, 虽各具特色, 但都存在各种遗传操作计算复杂且初始种群庞大、运行时间长的缺点。本文提出的遗传算法可以改善上述不足。

### 2 问题模型

移动 Agent 的多任务调度特点如下<sup>[2]</sup>：

- (1) 异构环境；
- (2) 可执行的主机数量有限；
- (3) 任务间约束关系已知且不限；
- (4) 数据传输延迟不限；
- (5) 不允许任务复制。

文献[2]用一简单的有向无回路图 DAG 表述移动 Agent 的多任务调度模型。这里用 5 元组表示：

$$\Pi = (H, R, G, P, \Phi)$$

其中，

$H = \{h_1, h_2, \dots, h_m\}$  表示可完成所有任务的主机集合。

$R = \{r(h_i, h_j) | h_i, h_j \in H\}$  表示主机间的数据传输速率。同一主机上执行的 2 个任务间有数据传输时, 无需通信延迟, 即  $r(h_k, h_k) = \infty$ 。

$P = \{p(T_i, h_j) | T_i \in T, h_j \in H\}$  是各任务在各主机执行的计算时间矩阵。 $p(T_i, h_j)$  表示任务  $T_i$  在主机  $h_j$  上执行时所需时间。如果某任务  $T_i$  不能在某主机  $h_j$  上执行, 则  $p(T_i, h_j) = \infty$ 。

$\Phi = \{d(T_i, T_j) | (T_i, T_j), T_i \neq T_j\}$ , 表示任务  $T_i$  需发送给任务  $T_j$  的数据量。

$G = (T, E)$  是 DAG 图。 $T = \{T_1, T_2, \dots, T_n\}$  表示移动 Agent 的任务集合。 $E = \{(T_i, T_j) | T_i, T_j \in T, T_i \neq T_j\}$  表示有向边集合, 有向边

**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; 陈立云, 副教授、硕士

**收稿日期:** 2007-08-30 **E-mail:** liuaizhen\_163@163.com

$(T_i, T_j)$ 表示任务 $T_i$ 和 $T_j$ 之间存在数据通信和优先约束关系,任务 $T_j$ 必须在任务 $T_i$ 执行完毕,且所需数据到达后,才能开始执行。并称 $T_i$ 为 $T_j$ 的一个直接前继任务, $T_j$ 为 $T_i$ 的一个直接后继任务。 $T_j$ 所有的直接前继任务构成 $T_j$ 的直接前继任务集。

基于上述模型的 Agent 调度的目标是将各任务映射到合适的主机上执行后,寻找符合约束关系且调度时间最短的那个调度。

### 3 本文算法

#### 3.1 编码方式

用 GA 求解多任务调度问题,其调度编码表示必须使遗传操作可以有效地检验新的调度来寻找最优调度。这里采用如下的编码方式:

任务:  $T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8$

主机:  $h_2 h_1 h_2 h_1 h_1 h_2 h_1 h_2$

该编码表示无论有多少主机,只使用 2 个列表:任务调度顺序列表和任务对应的调度主机列表。和常用的主机列表法<sup>[3-5]</sup>相比,其编程实现简单易行,且便于进行各种遗传操作。

#### 3.2 初始群体的生成

较好的初始种群有利于高效快速地进化最优解。任务图中的有向边 $d(T_i, T_j)$ 大小不一,其传输时间也不同,若 $T_i, T_j$ 的执行主机相同,则为 0,否则 $d(T_i, T_j)$ 越大,传输时间越长(假设主机间传送速率相同)。因此,在生成 DAG 图时,将 $d(T_i, T_j)$ 较大的任务对列出,放入禁忌表;在生成初始种群及进行其他操作时,作为参考。在生成初始种群前,先作如下定义:

**定义 1** 设 $T_i$ 为任务图 DAG 中的一个任务节点,如果 $PRED(T_i) = \emptyset$ ,则称 $T_i$ 为任务图的入节点。

**定义 2** 将原始任务图中所有入节点构成的集合称为第 1 次指派可选任务集合。

**定义 3** 从第 1 次指派可选任务集合中任选一个进行调度,而后将该任务及其有向边从任务图中删除,得到新的任务图。将新任务图中入节点构成的集合称为第 2 次指派可选任务集合。依次类推,可得到第  $n$  次指派可选任务集合。

生成有效调度的步骤如下:

(1)  $i=1$ 。

(2) 从第  $i$  次指派可选任务集合中随机选择一任务 $T_k$ 和一执行主机 $h_j$ ,分别加入任务列表和主机列表。

(3) 从第  $i$  次指派任务时对应的任务图中删去(2)中选定的任务 $T_k$ 及其有向边,得到第  $i+1$  次指派任务时对应的任务图及第  $i+1$  次指派的可选任务集合。

(4)  $i=i+1$ 。

(5) 随机从第  $i$  次指派的可选任务集合挑选一任务,查看禁忌表有无匹配的任务对,若有,则将该任务的主机指派为其已调度的前继任务的主机;否则,随机指派一主机。

(6) 重复(3)~(5),直至所有任务调度完毕,即可生成一有效调度。

设定种群规模,重复(1)~(6)即可生成初始群体。

#### 3.3 适应度函数

同文献[5],适应度函数 $f(s)$ 为

$$f(s) = C - FT(S)$$

其中, $FT(S)$ 为调度长度; $C$ 是一个可保证 $f(s) > 0$ 的常数。

#### 3.4 遗传操作机制

##### 3.4.1 选择算子

采用文献[5]轮盘赌方法选择  $N-1$  个体,为确保最佳个体被选中,将最佳个体作为第  $N$  个个体。

##### 3.4.2 杂交机制

多任务调度问题受限于各任务间的约束关系,2 个合法解,通过常规 GA 的标准杂交算子形成的新解,可能是非法的。因此需用特定规则控制杂交机制,以确保杂交后的解为合法解。常见的交叉机制主要有顺序交叉、PMX 交叉以及循环交叉等。但其中只有顺序交叉可保证一定产生合法解。这里提出一种新的杂交机制,其生成过程如下:

(1) 对于 2 个合法调度 $s_1$ 和 $s_2$ ,随机选择一任务,称该任务为交叉点任务。记下该任务在 2 个调度中的调度位置(分别设为 $p1, p2$ )。

(2)  $p1, p2$  分别把调度 $s_1, s_2$ 的任务列表集合分成前后 2 个子集,分别记为 $LS_1, RS_1, LS_2, RS_2$ 。相应的主机序列也分成 $HLs_1, HRs_1, HLs_2, HRs_2$ ;另设 $L's_1=LS_1, L's_2=LS_2$ 。

(3) 计算 $ss'_1=LS_2-Ls_1; ss'_2=LS_1-Ls_2$ 。

若 $ss'_1 \neq \emptyset$ ,将 $ss'_1$ 集合中的任务元素及 $s_2$ 中相应的调度主机依次加入到 $L's_1$ 和 $HLs_1$ ,此时 $L's_1=LS_1(Ls_2-Ls_1)$ 。

若 $ss'_2 \neq \emptyset$ ,则将 $ss'_2$ 集合中的任务元素及 $s_1$ 中相应的调度主机依次加入到 $L's_2$ 和 $HLs_2$ ,此时 $L's_2=LS_2(Ls_1-Ls_2)$ 。

(4) 分别将 $L's_1, HLs_1, L's_2, HLs_2$ 作为 2 个新调度 $s'_1$ 和 $s'_2$ 的任务序列和主机序列前半部分;将 $RS_2, HRs_2, RS_1, HRs_1$ 作为 $s'_1$ 和 $s'_2$ 的后半部分,加入到 2 个新调度中,加入过程中需去掉重复的任务及调度主机。此时得到 2 个新调度 $s'_1$ 和 $s'_2$ 的任务序列及主机序列。

显然,这里的交叉算子简单易行,没有复杂计算,虽不同于常规的单点和多点交叉,但根据其进化的新调度一定是可行解空间上的合法调度,限于篇幅,省略证明过程。

##### 3.4.3 变异算子

(1) 随机选定染色体。

(2) 找到任务数最多和最少的 2 个主机  $h_1, h_2$ 。

(3) 从基因为  $h_1$  的各基因位中任选一基因位,若其对应的任务不在禁忌表中,则该基因位的主机变为  $h_2$ ;否则再随机选择一基因位,将其主机变为  $h_2$ 。

该算子可提高解的多样性,也可平衡负载,促进算法收敛。

#### 3.5 算法框架

(1) 按 3.2 节生成规模为  $N$  的初始种群。

(2) 评价初始种群,记录最优个体  $p$ 。

(3) 置群体代数  $g=1$ 。

(4) 实施各算子,生成新一代种群。

(5) 评价新一代种群的适应度,并以  $p$  替代最差个体,同时保留本代最优个体  $p$ 。

(6) 检验收敛准则,若满足,转(7);否则  $g=g+1$ ,转(4);

(7) 输出最好个体。

#### 3.6 其他参数的选择

群体规模 $>35$ 时,算法运行时间变长,最优解没有明显改进,规模大小在 16~26 间即可,无需大种群。交叉、变异概率在推荐范围内对最优解影响不大,取常规值即可。

#### 4 仿真实验与结果分析

为检验本文算法的搜索能力,通过 Matlab 仿真来比较本文算法及文献[3]中的标准 GA 算法。多任务 DAG 图随机生成:分别仿真任务节点 $=\{50, 100\}$ ,通信计算代价比为: $CCR=\{0.1, 1.0, 10\}$ ,主机个数 $=\{3, 5, 11\}$ ,共 18 个图的情况。

每个任务节点有 1~4 个后继,每个任务的计算时间服从均值为 40(最小值=2,最大值=78)的均匀分布。有向边的传输时间根据 CCR 随机产生,服从均值为  $40 \times CCR$  的均匀分布。文献[3]参数见原文。本文算法主要参数:种群大小为 20,交叉和变异概率分别为:  $P_c=0.8, P_m=0.05$ 。

仿真实验运行平台是 P4/3.0 GHz/512 MB 的 PC,各算法解均为进化到 1 000 代时的最优解。算法运行多次,本文算法总能得到最佳的最优解。表 1 列出了 18 种算法在 1 000 代内得到的最优解对比。从表 1 可以看出,本文算法有更强的搜索能力,得到的最优解比标准 GA 算法明显更优。

表 1 2 种算法最优个体数比较

任务数	CCR	主机数	文献[3]方法	本文方法	改进量/(%)
50	0.1	3	699	624	10.7
		5	457	328	28.2
		11	319	200	37.3
	1	3	737	533	27.7
		5	517	384	25.7
		11	423	265	37.4
10	3	1 918	1 202	37.3	
	5	1 793	1 039	42.1	
	11	1 353	965	28.7	
100	0.1	3	1 455	1 193	18.0
		5	899	678	24.6
		11	508	425	16.3
	1	3	1 666	1 153	30.8
		5	1 212	719	40.7
		11	908	568	37.4
10	3	3 747	2 203	41.2	
	5	4 218	2 077	50.8	
	11	3 175	2 042	35.7	

图 1 示出了 5 个主机、50 个节点、 $CCR=0.1$  时 2 种算法的初始最优个体及随代数进化的曲线。

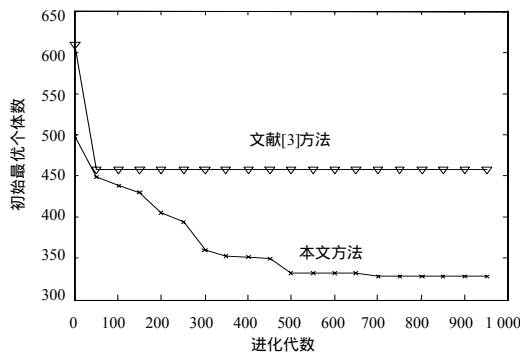


图 1 最优解进化曲线( $CCR=0.1$ )

图 2 示出了 11 个主机、100 个节点、 $CCR=10$  时 2 种算法的初始最优个体及随代数进化的曲线。从图 2 可以看出,在  $CCR$  较大的情况下,本文算法生成的初始种群的最佳个体明显优于标准 GA,甚至超过标准 GA 进化后的最优解,在此基础上进化解能迅速收敛到最优解。在  $CCR$  较小的情况下,本文算法生成的初始种群的最佳个体部分优于标准 GA,

但不一定(图 1)。在算法运行时间及收敛代数方面,本文算法不如标准 GA,但一般可在几秒内(100~500 代内)收敛到最优解,和求得的解的改进量相比,完全可以接受。

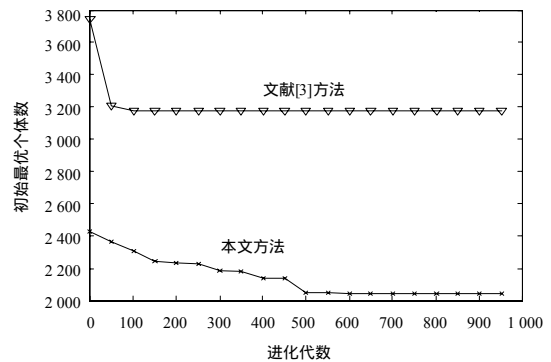


图 2 最优解进化曲线( $CCR=10$ )

## 5 结束语

本文给出了一种用于求解移动 Agent 多任务分配与调度问题的 GA,该算法可在整个可行解空间上进化,种群的搜索进化能力强;编码策略及交叉机制简单易行、无需复杂计算;设计的禁忌表、启发变异算子及最佳个体保留策略加速了算法收敛性,保证了最优解的不降性;充分考虑了异构环境中主机的差异及任务间数据传送开销,适用于异构计算系统模型。

## 参考文献

- [1] Pils C, Deibner K, Diepolder S, et al. The Performance Server: Rational Server Selection for Mobile Agents[C]//Proc. of the 9th IEEE Symposium on Computers and Communications. [S. l.]: IEEE Press, 2004: 13-18.
- [2] Rong Xie, Daniela, Stein C. Scheduling Multi-task Agents[C]//Proc. of the 5th International Conference on Mobile Agents. [S. l.]: IEEE Press, 2001: 260-276.
- [3] Edwin S H, Ansari N. Genetic Algorithm for Multiprocessor Scheduling[J]. IEEE Trans. on Parallel and Distributed Systems, 1994, 5(2): 113-120.
- [4] 钟求喜, 谢涛, 陈火旺. 基于遗传算法的任务分配与调度[J]. 计算机研究与发展, 2000, 37(10): 1197-1203.
- [5] 马义忠, 张聪, 周立文, 等. 基于异构计算系统的任务分配与调度算法[J]. 甘肃科学学报, 2005, 17(3): 94-98.
- [6] Annie S W, Yu Han, Jin Shiyuan, et al. An Incremental Genetic Algorithm Approach to Multiprocessor Scheduling[J]. IEEE Transactions on Parallel and Distributed Systems, 2004, 15(9): 824-834.
- [7] 王小英, 赵海. 异构计算系统任务调度的遗传算法及改进[J]. 系统仿真学报, 2006, 18(1): 26-32.