

## 一种改进的网格资源配置算法的研究与设计

杨庆娜<sup>1</sup>, 李定主<sup>2</sup>

(1 中北大学 电子与计算机科学技术学院, 山西 太原 030051; 2 北方自动控制技术研究所, 山西 太原 030006)

**摘 要:** 网格资源优化配置问题是一个 NP 问题, 遗传算法可以有效地解决这类问题. 针对基本遗传算法收敛速度慢的不足, 提出一种改进的遗传算法. 该算法通过产生符合条件的初始种群和保护优秀个体的交叉策略, 提高了算法的搜索能力和收敛速度. 提出的算法综合考虑了资源任务的通信代价、任务执行时间和资源利用率等参数; 并结合网格资源的特点, 将所有资源统一描述和对待, 扩展了资源的概念, 具有实际应用意义.

**关键词:** 网格; 资源优化配置; 改进遗传算法

**中图分类号:** TP393

**文献标识码:** A

**文章编号:** 1000-7180(2011)01-0150-04

## Research and Design of an Improved Algorithm of Grid Resources Allocation

YANG Qing-na<sup>1</sup>, LI Ding-zhu<sup>2</sup>

(1 Institute of Electronic and Computer Science, North University of China, Taiyuan 030051, China;

2 Institute of Automatic Control Technology Research, Taiyuan 030006, China)

**Abstract:** Grid resource optimized allocation is a NP problem, which can be solved by genetic algorithm efficiently. An improved genetic algorithm is proposed in order to overcome the defect of slow convergence speed in this algorithm. Qualified initial population and excellent members protecting crossover strategies are brought in, they all can help to enhance the searching ability and convergence speed. This method plays a practical role by considering many parameters such as the communication cost between the resources and tasks, execution time of task and the utility of resources, and describing resources uniformly in the way of combining the characteristics of grid resources.

**Key words:** grid; resource optimized allocation; improved genetic algorithm

### 1 引言

网格是把地理位置上分散的资源集成起来的一种基础设施, 其目标是实现网络虚拟环境上的高性能资源共享和协同工作, 共同完成一些缺乏有效研究办法的重大应用研究问题<sup>[1]</sup>. 网格资源具有异构性、动态性和自治性等特点<sup>[2]</sup>, 如何对网格中的资源进行合理配置和管理是网格技术研究的一个难点. 显然, 网格资源的配置问题是一个 NP 完全问题. 目前, 对于 NP 完全问题, 人们通常采用启发式算法<sup>[3]</sup>来简化问题, 寻求问题的最优解, 使用最多的方法有贪婪算法、规划论、遗传算法等. 与前两种算法相比较, 遗传算法作为一种有效的启发式全局随机搜索

算法<sup>[4]</sup>, 避免了局部最小问题, 能够得到更满意的结果. 但是直接使用遗传算法存在搜索空间大, 效率不高的问题. 因此, 文中提出一种改进的遗传算法解决网格资源优化配置的问题.

### 2 问题描述及相关定义

典型的资源—任务配置问题描述: 给定一个任务图和可获取的资源集合, 任务图确定了需要处理的所有任务, 任务之间的执行顺序、处理任务的时间需求、对资源的需求能力等基本特征. 资源具备处理任务的功能. 资源和任务之间通过任务的能力需求和资源的提供能力相关联, 以此进行资源任务分配. 简单的说, 资源优化配置就是将  $m$  个异构的可用资

源合理的分配给  $n$  个任务,使得任务完成的总时间最小,并使资源利用率最高.目前所研究的资源分配及资源调度问题,大多数将计算资源和存储资源分为两种不同的因素,本文对所有资源统一对待,资源是指能被分配给任务、执行并完成任务的一个实体,网络资源扩展了资源的概念,不仅包括技术资源如计算资源、通信资源、存储资源、设备仪器等,广义地还包括参与任务的人类资源<sup>[5]</sup>.

假设网格中有  $m$  个异构的节点,每个节点提供  $h$  种不同类型的资源,执行任务需要一定的时间;提交到网格中的应用已经被分解成多个子任务,子任务之间的数据依赖关系已知,每个任务都有一个  $h$  维的资源能力需求,要求在一定的地理位置执行.记  $K = \{k_1, k_2, \dots, k_q\}$  为资源类型集,  $R = \{r_1, r_2, \dots, r_h\}$  为资源集,每一个资源有唯一的资源类型,即对任意一个资源  $r_i \in R$ ,有且只有一个  $k_i \in K$  与之对应.用  $J = \{j_1, j_2, \dots, j_m\}$  表示网格中不同位置的节点集,每个资源处于各个节点.记任务集为  $T = \{t_1, t_2, \dots, t_n\}$ ,  $n$  是分解得到的子任务数量.  $B(j_p, j_q)$  表示不同节点间的传输带宽,  $D(j_p, j_q)$  表示不同节点间的传输延迟,  $Q(t_i, r_{j_l})$  表示在任务  $t_i$  和资源  $r_{j_l}$  之间的数据传输量,在这里假定资源的分配是一个原子事务,即一个任务所需的资源可在同一时间得到.在得到任务所需的全部资源后,任务  $t_i$  才开始执行,记  $S_{(t_i)}$  为  $t_i$  执行时为其分配的资源节点集合.在任务执行结束后释放这些资源,使它们变为可用资源.下面做如下定义:

**定义 1** 任务  $t_i$  最早可执行时间  $\text{Start}(t_i)$ ,即其前置任务的最晚完成时间.

**定义 2** 任务  $t_i$  的估计执行时间  $E(t_i)$ ,定义为分配给任务的资源组合中执行任务花费的最大时间.

**定义 3** 任务  $t_i$  和所提供资源集  $S_{(t_i)}$  之间的传输时间

$$d_i = \begin{cases} 0, & \text{若 } t_i \text{ 与 } t_j \text{ 处于同一节点} \\ \sum_{j_j \in S_{(t_i)}} (Q(t_i, r_{j_l}) \times B(j_i, j_j) + D(j_i, j_j)), & \text{否则} \end{cases} \quad (1)$$

上式表示当任务和所分配的资源处于同一节点位置时,没有节点延迟,也不需要数据传输;  $j_i$  表示任务  $t_i$  所在的节点位置,  $j_j$  表示  $r_{j_l}$  所在的资源节点位置.

**定义 4** 任务  $t_i$  的最早完成时间  $\text{Comp}(t_i) = \text{Start}(t_i) + E(t_i) + d_i$ .

### 3 基于改进遗传算法的资源优化配置

#### 3.1 遗传算法简介

遗传算法 (Genetic Algorithms, GA) 是一种借鉴生物界自然选择和自然遗传机制的高度并行、随机、自适应搜索算法.遗传算法在本质上是一种不依赖具体问题的直接搜索方法,它从某一初始群体出发,遵从“优胜劣汰”的原则,通过复制、杂交、变异等算子,不断迭代运行而逼近最优解,因此非常适合于优化问题.文中提出的改进遗传算法,提高了遗传算法的收敛速度和质量.

#### 3.2 改进遗传算法设计

##### (1) 编码

染色体编码是从表现型到基因型的映射,可以用二进制、整数、和实数变量等表示问题的编码,本文采用整数表示.任务和资源都用它们的序号表示,染色体的长度为就绪队列中任务的个数.染色体表示如图 1 所示.

|       |       |
|-------|-------|
| $t_1$ | $r_1$ |
| $t_2$ | $r_2$ |
| ...   | ...   |
| $t_i$ | $r_i$ |
| ...   | ...   |
| $t_n$ | $r_n$ |

图 1 染色体的编码方式

其中,  $t_i$  表示任务的编号,对应的  $r_i$  表示任务  $t_i$  执行时占用资源的节点编号.每个元素为二元组(任务号,所需资源节点序列),每个任务对应的资源需求可由一个节点的资源或局部节点的资源组合而成,在得到全部所需资源后任务才能执行.

##### (2) 初始种群的产生

种群的大小直接影响收敛速度和最优解的产生,群体太大会降低收敛速度,太小难以产生最优解,初始种群一般都是随机产生的.本算法产生初始种群个体时,只查找那些能够满足任务需求的资源节点来进行匹配,在满足要求的解中再随机地选取样本,相比基本遗传算法而言这样可以更快的达到最优解.

##### (3) 适应度计算

遗传算法在进化搜索中基本不利用外部信息,仅以适应度函数为依据,利用种群中每个个体的适应度值来进行搜索.因此适应度函数的选取至关重要,选择适当的适应度函数可以对每次迭代进行较

好的评估. 一般而言, 适应度函数由目标函数变换而成. 综合文中所述, 该问题的目标函数有:

目标函数  $F_1$ : 最早开始任务的开始时间与最晚完成任务的完成时间的时间跨度最小.

$$F_{\max} = \max\{(\text{Comp}(t_j) - \text{Start}(t_i), t_i, t_j \in T)\}$$

其中,  $t_i$  表示最早开始任务,  $t_j$  表示最晚执行的任任务. 目标函数为  $F_1 = \min\{F_{\max}\}$

目标函数  $F_2$ : 资源利用率最大.

$$F_2 = \max\left(\sum_{i=1}^n \sum_{j=1}^m \text{Cap}_i^j / \sum_{i=1}^n \sum_{j=1}^m S \text{Cap}_i^j\right)$$

其中,  $\text{Cap}_i^j$  表示任务  $i$  的资源需求能力第  $j$  位的值,  $S \text{Cap}_i^j$  表示提供给任务资源的节点资源提供能力第  $j$  位的值.

将目标函数  $F_1$  转换为任务时间最大缩短程度目标函数  $F'_1$ , 有

$$F'_1 = \max\left(\frac{E(TE) - \min(F_{\max})}{E(TE)}\right)$$

其中,  $E(TE)$  指的是任务期望的完成时间.

采用多目标加权法将多目标的优化问题转换为单目标的优化问题, 最终得到的适应度函数  $F$  为:

$$F = w_1 F'_1 + w_2 F_2$$

其中,  $w_1, w_2$  为决策者给定的权重值.

(4) 选择操作: 先计算单个染色体的适应度值  $f_i$ , 并将适应度最高的染色体直接复制到下一代; 计算单个染色体的存活概率  $p_i = f_i / \sum_{i=1}^n f_i$ , 根据轮盘赌选择法对染色体进行选择, 选中的染色体保留到下一代, 否则被淘汰.

(5) 交叉

选择染色体进行交叉操作. 相比基本遗传算法, 该算法在交叉操作中只选择父子两代中适应度值大的两条保存到下一代中. 交叉操作可分为两种类型: 任务交叉和资源交叉. 任务交叉就是在两个染色体中选出任务相同的两组整体进行互换; 资源交叉就是选取任务相同的两组, 仅将两组中资源部分进行互换. 交叉概率  $P_c = 0.8$ , 然后以等概率进行任务交叉或资源交叉. 任务交叉如图 2 所示.

(6) 变异

变异也可分为任务变异和资源变异, 对于资源变异就是将染色体中某个任务所需要的资源组合中某个资源替换为同类型的其它节点资源. 任务变异就是将染色体中某两个任务组位置互换. 变异操作中, 变异概率不能太大, 如果太大, 遗传算法就会退化为随机搜索, 变异算法的一些重要数学特征和搜

|   |       |   |     |   |       |   |     |
|---|-------|---|-----|---|-------|---|-----|
| 1 | 2     | 4 | 7,8 | 5 | 1,2   | 4 | 7,8 |
| 2 | 4,1,3 | 1 | 6   | 2 | 4,1,3 | 5 | 1,8 |
| 3 | 5,6   | 2 | 3,5 | 3 | 5,6   | 2 | 3,5 |
| 4 | 3,7   | 5 | 1,2 | 4 | 3,7   | 1 | 2   |
| 5 | 1,8   | 3 | 4,6 | 1 | 6     | 3 | 4,6 |

Chrom1      Chrom1 2      new Chrom1      new Chrom2

图 2 任务交叉的例子

索能力将会失去, 本文选取变异率  $P_m = 0.2$ , 然后以等概率进行任务变异或资源变异.

(7) 算法终止条件

文中设置以下两个终止条件: (1) 预设最大迭代代数, 当达到最大迭代代数时, 执行终止, 输出最优结果; (2) 当种群适应度值最大的染色体连续 30 代适应度值不变时, 执行终止, 将最优结果输出给决策者.

## 4 算法流程

(1) 初始化种群大小参数  $M$ , 最大迭代次数  $\text{Maxnum}$ , 令当前执行代数  $\text{generation} = 0$ .

(2) 根据初始种群生成条件生成合适的染色体放入种群中, 计算每条染色体的适应度值.

(3) 按照概率选择染色体进行交叉判断染色体是否为合格染色体(根据生成初始染色体的条件), 若合格计算新染色体的适应度值, 选出父子两代中适应度值最大的两条放入下一代种群中, 否则转向(3).

(4) 按照概率选择染色体进行变异操作(根据生成初始染色体的条件), 判断是否为合格染色体, 若合格将新染色体加入种群, 否则转(4).

(5) 计算染色体的适应度值生成下一代种群,  $\text{generation}++$ .

(6) 判断是否满足终止条件, 若满足, 执行终止输出最优结果; 如果不满足, 则返回(3).

## 5 测试算例分析

假设有表 1 的任务需要处理, 则有表 2 的资源可以使用.

根据文中提出的算法进行资源优化配置, 设置初始种群大小为 20,  $w_1, w_2$  都取为 1.0, 在第 78 代算法终止, 获得最优结果如下:

(1) 最佳适应度值为 1.3177;

(2) 最佳适应度值对应的最优结果: (1, (10)), (2, (3, 8)), (3, (5)), (4, (1, 2)), (5, (4, 8)), (6,

(3,6,7));

(3) 最大资源利用率为 0.7838.

表 1 任务表

| 任务 | 参数 | 资源需求  |       |       |       |       | 计划开 | 计划结 | 前置 | 后置  | 位置        |
|----|----|-------|-------|-------|-------|-------|-----|-----|----|-----|-----------|
|    |    | $r_1$ | $r_2$ | $r_3$ | $r_4$ | $r_5$ | 始时间 | 束时间 | 任务 | 任务  | ( $x,y$ ) |
| 1  |    | 3     | 0     | 2     | 2     | 1     | 1   | 3   | —  | 2,3 | (0,0)     |
| 2  |    | 0     | 0     | 4     | 4     | 0     | 3   | 6   | 1  | 4   | (0,1)     |
| 3  |    | 1     | 4     | 0     | 0     | 0     | 3   | 7   | 1  | 5   | (2,2)     |
| 4  |    | 1     | 0     | 4     | 0     | 3     | 6   | 9   | 2  | 6   | (3,4)     |
| 5  |    | 0     | 4     | 7     | 0     | 0     | 7   | 10  | 3  | 6   | (3,6)     |
| 6  |    | 0     | 6     | 0     | 12    | 0     | 10  | 16  | —  | 4,5 | (4,2)     |

表 2 资源表

| 节点 | 参数 | 资源能力  |       |       |       |       | 机动 | 执行任务 | 初始    |
|----|----|-------|-------|-------|-------|-------|----|------|-------|
|    |    | $r_1$ | $r_2$ | $r_3$ | $r_4$ | $r_5$ | 速度 | 的时间  | 位置    |
| 1  |    | 0     | 0     | 4     | 0     | 5     | 2  | 1    | (3,4) |
| 2  |    | 2     | 4     | 0     | 0     | 1     | 2  | 2    | (3,5) |
| 3  |    | 0     | 2     | 0     | 4     | 0     | 2  | 1    | (0,2) |
| 4  |    | 0     | 4     | 5     | 3     | 0     | 1  | 2    | (5,6) |
| 5  |    | 1     | 5     | 0     | 2     | 0     | 1  | 1    | (2,2) |
| 6  |    | 0     | 6     | 0     | 0     | 2     | 2  | 2    | (4,2) |
| 7  |    | 0     | 0     | 0     | 8     | 0     | 1  | 1    | (4,5) |
| 8  |    | 0     | 0     | 5     | 0     | 0     | 2  | 1    | (3,0) |
| 9  |    | 0     | 0     | 0     | 5     | 5     | 1  | 2    | (6,8) |
| 10 |    | 3     | 0     | 2     | 4     | 2     | 2  | 1    | (6,0) |

在基本遗传算法下执行需要迭代 97 次,因此可以看出该算法大大缩小了搜索空间,加快了收敛速度.由于该算法综合考虑了协作资源和任务之间的通信量,任务的执行时间,资源利用率等参数,具有实际的应用价值.

6 结束语

文中在进行资源分配时结合网格资源的特点,将所有异构资源统一描述和对待,扩展了资源的概念,并针对基本遗传算法在进行资源优化配置时的不足,对遗传算法进行了改进.从测试算例中可以看出,该算法能在获得最优解的条件下提高搜索能力、加快收敛速度,可以有效的解决网格资源优化配置问题.

参考文献:

[1] Fo ster I, Kesselman C, Tuecke S. The anatomy of the grid: enabling scalable virtual organizations[J]. International Journal of Supercomputer Applications, 2001, 15 (3): 200—222.

[2] 罗宇平. 基于 Min2Min 改进后的网格调度算法[J]. 微电子学与计算机, 2009, 26(3): 86—88.

[3] Alhusaini A H, Prasanna V K, Raghavendra C S. A framework for mapping with resource coallocation in heterogeneous computing systems [C]// Proc. of 9th Heterogeneous Computing Workshop. Mexico, Cancun, 2000: 273—286.

[4] 王小平, 曹立明. 遗传算法理论、应用与软件实现[M]. 西安: 西安交通大学出版社, 2002.

[5] 何琨, 赵勇. 网格环境下资源调度问题的统一建模与分析[J]. 华中科技大学学报: 自然科学版, 2006, 34(3): 35—38.

作者简介:

杨庆娜 女, (1985—), 硕士研究生. 研究方向为网格计算.

李定主 男, (1965—), 研究生导师, 研究员级高级工程师, 教授. 研究方向为多媒体与计算机仿真、网络技术应用.

(上接第 149 页)

[2] 田有文, 王文涛, 王思宇. 基于 BP 神经网络的农村用电量预测的研究[J]. 沈阳农业大学学报, 2005, 36(4): 506—507.

[3] Bo Jin, Tang Y C, Zhang Yanqing. Support vector machines with genetic fuzzy feature transformation for bio-medical data classification [J]. Information Sciences, 2007, 177(2): 476—489.

[4] Matteo Pardo, Giorgio Sberveglieri. Classification of electronic nose data with support vector machines [J]. Sensors and Actuators B: Chemical, 2005, 107(2): 730—737.

[5] 孙向军, 赵斯强, 严宗睿. 基于粒子群优化的反潜搜索研究[J]. 微电子学与计算机, 2008, 25(10): 91—93.

[6] 韦杏琼, 周永权. 基于粒子群算法的数值积分方法研究[J]. 微电子学与计算机, 2009, 26(7): 117—119.

作者简介:

曾勍炜 男, (1973—), 硕士, 副研究员. 研究方向为应用软件开发.

徐知海 男, (1975—), 硕士, 助理研究员. 研究方向为应用软件开发.

吴 键 男, (1973—), 高级工程师. 研究方向为电力调度.