

# 基于云模型的网格任务调度遗传算法研究

郑世明<sup>1</sup>, 高志年<sup>1</sup>, 韦伟<sup>1</sup>, 苗壮<sup>2</sup>, 邵荣明<sup>3</sup>

(1. 南京陆军指挥学院作战实验中心 南京 210045; 2. 解放军理工大学指挥信息系统学院 南京 210007;  
3. 解放军沈阳炮兵学院训练部 沈阳 110867)

**【摘要】**针对网格环境动态多变性的特点, 为了克服传统遗传算法易陷入局部最优的缺陷, 提出了一种基于云模型的网格任务调度遗传算法。该算法由正态云模型的Y条件云发生器实现交叉操作, 由基本云发生器实现变异操作, 对调度模型进行优化求解, 并在任务调度中对初始种群的产生、选择、变异和交叉操作进行了改进, 通过实验分析, 表明了该算法的可靠性、有效性和实用性。

**关键词** 云模型; 遗传算法; 网格; 任务调度

中图分类号 TP29

文献标识码 A

doi:10.3969/j.issn.1001-0548.2012.06.018

## Grid Task Scheduling Genetic Algorithm Based on Cloud Model

ZHENG Shi-ming<sup>1</sup>, GAO Zhi-nian<sup>1</sup>, WEI Wei<sup>1</sup>, MIAO Zhuang<sup>2</sup>, and SHAO Rong-ming<sup>3</sup>

(1. Operational Experiment Center, Nanjing Army Command College Nanjing 210045;

2. Institute of Command Information Systems, PLA University of Science and Technology Nanjing 210007;

3. Training Department, Shenyang Artillery Academy of PLA Shenyang 110867)

**Abstract** Aiming at the dynamic characteristic of grid, to overcome the shortcomings of genetic algorithms which easily get a local optimum solution, a cloud-based genetic algorithm (CGA) for grid task scheduling is proposed. CGA is based on both the idea of GA and the properties of randomness and stable tendency of a normal cloud model. In this algorithm, Y-conditional cloud generator is used for the crossover operator, and basic cloud generator is used for the mutation operator. CGA can optimize the solution with genetic algorithm based on cloud-model, ascertain the optimal scenario for scheduling, and improve on the arithmetic operators of population initialization, select, crossover, mutation and reinsertion in the process of task scheduling. The experiment validates the feasibility, validity, and practicality of the algorithm.

**Key words** cloud model; genetic algorithms; grid; task scheduling

网格是构筑在互联网上的一组新兴技术和基础设施, 实现互联网上所有资源的全面连通, 网格环境下的任务调度是一个复杂的过程, 现有的一些任务调度方法并不能很好的适应网格特点, 如调度算法的高效性、资源的异构性以及资源调度的并行性和分布性等。文献[1]提出的基于个体适应度值和种群平均适应度值的自适应遗传算法(adaptive genetic algorithm, AGA)的影响比较广泛, AGA在维持种群多样性和寻优能力上都有显著改善, 但也容易使进化走向局部最优解的可能性增加。文献[2]提出了一种基于云自适应的遗传算法, 利用云发生器自适应

调整交叉、变异概率, 避免陷入局部最优。文献[3]提出了一种应用遗传算法解决DAG图表示的任务调度问题, 该方法可以处理包含多个有前后约束关系的子任务, 适用于大规模的网格应用。文献[4]针对网格环境下的硬实时任务调度问题, 设计了新的选择、交叉算子保证种群多样性, 具有较好的调度成功率和系统吞吐率。为了克服传统遗传算法易陷入局部最优解的缺陷, 本文利用云模型的随机性和稳定倾向性的特点, 提出了一种基于云模型的网格任务调度遗传算法, 使概率值既有传统的趋势性, 满足快速寻优能力, 又具有随机性, 从而提高了避免

收稿日期: 2011-02-17; 修回日期: 2011-06-29

基金项目: 国家863项目(2007AA01Z126); 部级预研基金; 江苏省自然科学基金(BK2012512)

作者简介: 郑世明(1981-), 男, 博士, 主要从事网格、数据挖掘、人工智能、作战仿真方面的研究。

陷入局部最优的能力。

## 1 云模型理论

云模型是语言值表示的某个定性概念与其定量表示之间的不确定性转换模型,反映自然语言中概念的不确定性、随机性和模糊性的关联性,构成定性和定量之间的映射。云模型用期望值 $Ex$ 、熵 $En$ 和超熵 $He$  3个数字特征来整体表征一个概念。生成云滴的算法或硬件称为云发生器,3个主要云发生器分别是基本正态云发生器、 $X$ 条件云发生器和 $Y$ 条件云发生器<sup>[5]</sup>。

云模型是基于概率的云滴及其确定度的分布模型,对模糊性和随机性的关联性,建立了简单、合理的映射关系。

**定义 1**  $U$ 上的定性模糊概念,若定量值 $x \in U$ 且 $x$ 是定性概念 $C$ 的一次随机实现, $x$ 对 $C$ 的确定度 $u_c(x) \in [0,1]$ 是有稳定倾向的随机数 $u$ ,每一个 $x$ 对应一个 $u_c(x)$ ,则 $x$ 在论域 $U$ 上的分布称为云,每一个 $x$ 称为一个云滴<sup>[6]</sup>。

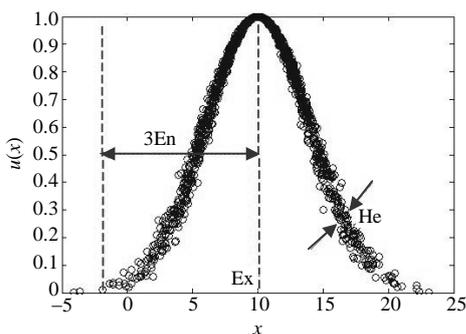


图1 云模型数字特征示意图

图1是一维正态云模型 $Cloud(10,4,0.25,1,500)$ 特征示意图,具有云模型典型特征的数值10、4、0.25,分别代表云重心、云覆盖区域和云厚度,重心代表了种群的整体分布情况,熵说明了种群分布跨度,超熵则反映了种群中个体间的离散程度。

## 2 问题描述

由于实际的网格任务调度具有众多复杂的参数,为简化模型作如下假设:1) 采用批处理模式,即等待所有的任务到达后开始执行,能充分利用资源并给出合理的任务映射策略;2) 所有任务为相互独立的任务(无约束关系和执行的先后关系),即元任务;3) 模拟调度为分布式调度;4) 子任务的数量大于当前可调用的资源数量,否则只需简单地按任务大小分配给速度较快的资源即可<sup>[7]</sup>;5) 资源性能不

完全相同,即各个任务在不同资源上的执行时间可能不同;6) 资源是时间共享还是空间共享随机产生<sup>[8]</sup>;7) 不考虑任务所占用的存储资源、通信开销和网络延迟,即只考虑任务消耗的计算资源;8) 若第 $j$ 个任务在第 $i$ 个资源的执行时间为 $H_{ij}^j$ ,则多个任务在该资源上同时进行的总时间为单个任务在该资源上执行时间之和。

下面给出网格任务调度的问题定义。

**定义 2**  $n$ 个相互独立的子任务集合记为 $T=\{T_1, T_2, \dots, T_n\}$ ,其中 $T_i$ 为第 $i$ 个子任务。

**定义 3**  $m$ 个异构分布的计算资源集合为 $P=\{P_1, P_2, \dots, P_m\}$ ,其中 $P_j$ 为第 $j$ 个计算资源。

**定义 4**  $H$ 为一个 $m \times n$ 矩阵, $h_{ji}$ 为第 $i$ 个子任务在第 $j$ 个计算资源需要运行的时间。

$$H = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1n} \\ h_{21} & h_{22} & \dots & h_{2n} \\ \vdots & \vdots & h_{ji} & \vdots \\ h_{m1} & h_{m2} & \dots & h_{mn} \end{bmatrix}$$

**定义 5** 分配的任务集合 $\psi_{P_1} = \{T_i \mid i \text{ 为分配给第1个计算资源的任务编号}\}$ , $H_{P_1}^i = \sum_{i \in \psi_{P_1}} h_{ji}$ 表示第 $j$ 个计算资源完成分配的任务所需要的总时间,分配的任务集合 $\psi_{P_j} = \{T_i \mid i \text{ 为分配给第}j\text{个计算资源的任务编号}\}$ 。

**定义 6**  $h_j$ 表示各子任务在不同计算资源上的估计执行时间(根据任务的计算量和各计算资源的计算能力得出),其中 $H_{P_1}^i = \sum_{i \in \psi_{P_1}} h_{ji}$ 表示第1个计算资源完成分配的任务所需要的时间。

设 $S$ 为某次的一个调度策略, $H_s(P_j)$ 为调度策略 $S$ 中,当计算资源 $P_j$ 完成分配到该节点上的最后一个任务所花费的总时间,设 $H(S)$ 为整个调度 $S$ 所花费的总时间,称为一次调度 $S$ 的完成时间,有: $H(S) = \max(H_s(P_i))$ , $1 < i < n$ ,目标是求 $\min\{H(S)\}$ ,即通过遗传算法寻找一个最佳调度策略 $S$ ,使 $H(S)$ 最小。

## 3 云模型的网格任务调度遗传算法

### 3.1 基于云模型的分布式遗传调度算法

首先给出交叉概率 $P_c$ 、变异概率 $P_m$ 的求解算法。算法1(求解交叉概率 $P_c$ ):采用基本云发生器。输入:  $\{Ex, En, He\}$ ,  $n$ //数字特征和云滴数。输出:  $P_c$ 。

$$Ex = \bar{f}_x = \frac{\sum_{i=1}^N f_i(x)}{N}, S^2 = \frac{\sum_{i=1}^N (f_i(x) - \bar{f}_x)^2}{N-1}$$

$$En = [\alpha(f_{\max} - \bar{f}_x) + \beta(\bar{f}_x - f_{\min})]/c_1 \quad \alpha + \beta = 1$$

$$He = \sqrt{S^2 - En^2}/c_2$$

for  $i=1$  to  $n$

$$En' = \text{RANDN}(En, He)$$

$$f^* = \text{RANDN}(Ex, En')$$

$$P_c = \begin{cases} c_3 e^{-\frac{(f^* - Ex)^2}{2(En')^2}} & f^* \geq \bar{f} \\ c_4 & f^* < \bar{f} \end{cases}$$

算法2(求解变异概率 $P_m$ ): 采用Y条件云发生器。

输入:  $\{Ex, En, He\}$ ,  $\mu_0, n$ 。输出:  $P_m$ 。

$$Ex = \bar{f}_x = \frac{\sum_{i=1}^N f_i(x)}{N}$$

$$En = [\alpha(f_{\max} - f_x) + \beta(f_x - f_{\min})]/c_1 \quad \alpha + \beta = 1$$

$$He = En/c_2$$

for  $i=1$  to  $n/2$

$$En' = \text{RANDN}(En, He)$$

$$\mu^* = \text{RANDN}(Ex, En')$$

$$P_m = \begin{cases} c_3 (Ex - En' \sqrt{-2 \ln(\mu_0)}) & \mu_0 \geq \mu^* \\ c_4 (Ex + En' \sqrt{-2 \ln(\mu_0)}) & \mu_0 < \mu^* \end{cases}$$

随着种群规模扩大, 对于参数交叉和变异概率的合理设置是困难的, 本文借助云模型能反映随机性和模糊性的关联性, 构成定性和定量之间映射的特点, 提出一种基于云模型的分布式遗传调度算法(distributed genetic scheduling algorithm based on cloud model, DGSABCM), 结合云模型的特性产生交叉概率 $P_c$ 和变异概率 $P_m$ 。这样处理的优点在于能够很好地控制交叉和变异概率, 从而防止遗传算法过早陷入局部收敛。

基于云模型的分布式遗传调度算法步骤如下:

- 1) G-Web根据计算任务选择编码方式。
- 2) G-Web根据相应的编码随机生成 $N$ 个子种群。
- 3) G-Web将 $N$ 个子种群的相关参数分别送给 $N$ 个不同的L-Web进行节点服务调用(每个L-Web获得一个子种群)。
- 4) 每个局部资源管理代理(L-Web)在约束条件下计算染色体适应度, 独立地进行选择、交叉、变异操作。

① 选择操作: 采用最佳保留选择策略; ② 交叉操作: 采用启发式交叉策略; ③ 变异操作: 采用

基于Aloplex算法的自适应变异策略<sup>[9]</sup>; ④ 适应度的计算。

5) 当染色体进化满足一定的条件后, L-Web提交适应度相对较高的部分染色体至G-Web, 完成以下操作:

- ① 染色体适应度计算;
- ② 选择操作: 采用排序选择策略;
- ③ 交叉操作: 采用交叉概率 $P_c$ 的算法;
- ④ 变异操作: 采用变异概率 $P_m$ 的算法。

6) 当某一个L-Web符合终止条件时, 提交适应度最高的染色体给G-Web并终止L-Web, 转步骤7), 否则转步骤4)。

7) G-Web收集所有的局部管理代理节点传送来的染色体, 计算并比较所有染色体的适应度, 选择适应度最高的染色体作为求解结果。

8) 根据解码选择计算节点完成网格计算任务。

### 3.2 基于自适应云模型的遗传调度算法

对于概率的控制能从一定程度上解决局部收敛的问题, 在研究过程中发现了一种更方便的方法直接对个体进行变异: 通过云模型把进化和遗传作为一种定性知识来描述, 再进行不确定的定性和定量转换<sup>[10]</sup>。本文提出一种基于自适应云模型的遗传调度算法(based-adaptive cloud Model genetic scheduling algorithm, BACMGSA), 充分利用云模型对模糊性和随机性的关联性, 不通过云模型产生变异和交叉概率, 而是运用云模型直接产生交叉和变异个体。

排序选择策略的基本思想: 对群体中的所有个体按照其适应度大小进行排序, 在此基础上来分配各个体被选中的概率。

基于自适应云模型的遗传调度算法步骤如下:

- 1) G-Web根据计算任务选择编码方式。
- 2) G-Web根据相应的编码随机生成 $N$ 个子种群。
- 3) G-Web将 $N$ 个子种群的相关参数分别送给 $N$ 个不同的L-Web进行节点服务调用(每个L-Web获得一个子种群)。
- 4) 每个局部资源管理代理(L-Web)在约束条件下计算染色体适应度, 独立地进行选择、交叉、变异操作。

① 选择操作: 采用排序选择策略; ② 交叉操作: 采用算术交叉策略; ③ 变异操作: 采用均匀变异策略; ④ 适应度计算。

5) 当染色体进化满足一定的条件后, L-Web提交适应度相对较高的部分染色体至G-Web, 完成以下操作:

- ① 染色体适应度计算;
- ② 选择操作: 采用最优保存策略;
- ③ 交叉操作: I. 由均匀分布随机生成隶属度  $\mu_0$ ; II.  $E_x$ 根据父母双方的适应度值加权得到, 具体方法是  $E_x = \omega_1 f(X) + \omega_2 f(Y)$ ,  $f(X)$ 表示母亲个体的适应度值,  $f(Y)$ 表示父亲个体的适应度值,  $\omega_1 + \omega_2 = 1$ ; III. 计算云模型的熵:  $En = [\alpha(f_{max} - \bar{f}_x) + \beta(\bar{f}_x - f_{min})] / c_1$   $\alpha + \beta = 1$
- IV. 计算云模型的标准差:  $He = En / c_2$ ; V. 产生  $n/2$  对后代个体:

for  $i=1$  to  $n/2$   
 $En' = \text{RANDN}(En, He)$   
 个体  $A_i = E_x - En' \sqrt{-2 \ln(\mu_0)}$   
 个体  $B_i = E_x + En' \sqrt{-2 \ln(\mu_0)}$

- ④ 变异操作: 给定变异概率  $P_{m_0}$ :
  - I. 若  $P_{m_0} < \mu_0$ ,
    - for  $i=1$  to  $n$
    - $En' = \text{RANDN}(En, He)$
    - $x_i^* = \text{RANDN}(E_x, En')$
  - II. 若  $P_{m_0} \geq \mu_0$ , 直接按照概率  $P_{m_0}$  进行变异。
- 6) G-Web计算并比较所有的染色体的适应度, 选择适应度最高的染色体作为求解结果。
- 7) 根据解码选择计算节点完成网格计算任务。

### 4 实验结果分析

本文对算法进行了仿真实验, 模拟了由3个站点(500个计算节点)组成的网格计算环境下的任务调度, 每个站点的资源情况如表1所示, 算法主要参数为: 种群大小为500, 最大遗传次数为50, 代沟GGAP为0.9, 实验结果如图2所示。

表1 实验环境配置

站点名称	节点数量配置
站点1	250个计算节点
站点2	150个计算节点
站点3	100个计算节点

图2表示的是BACMGSA、DGSABCM、CGA、AGA和SGA在不同迭代次数下适应度值的变化情况。从图中可以看出, DGSABCM是一种基于云模型的分布式调度算法, 总体适应度值比较平稳, 反映了算法良好的稳定性; BACMGSA是一种基于云模型的自适应算法, 初始的适应值大小与初始种群选择有关, 随着迭代的深入, 能随约束条件的限制自适应地向最优进化发展, 实验发现经过10次遗传

迭代以后, 算法就表现出了良好的收敛性, 是一种全局收敛, 在精度要求不高的前提下, 在迭代次数很少的情况下可以大大降低算法的复杂度, 节约算法的执行时间, 达到较好的效果; CGA是文献[2]提出的一种云遗传算法, 获取最优解的搜索能力不强; AGA是文献[1]提出的一种自适应遗传算法, 受初始种群的影响比较大, 获得的适应度函数值比较低, 即寻找优秀个体的能力相对较弱; SGA是标准的遗传算法, 算法性能有时会超过CGA和AGA, 在大规模的网格任务调度中, 效果不理想, 稳定性不够好<sup>[3]</sup>。

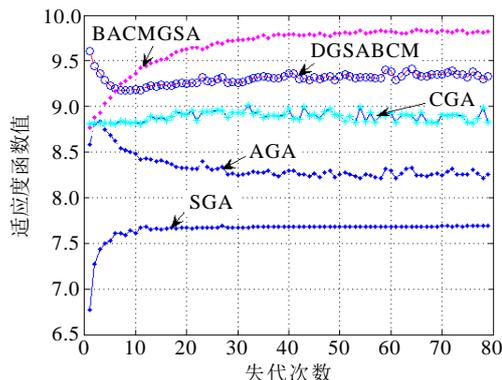


图2 BACMGSA、DGSABCM、CGA、AGA和SGA算法目标函数适应值比较

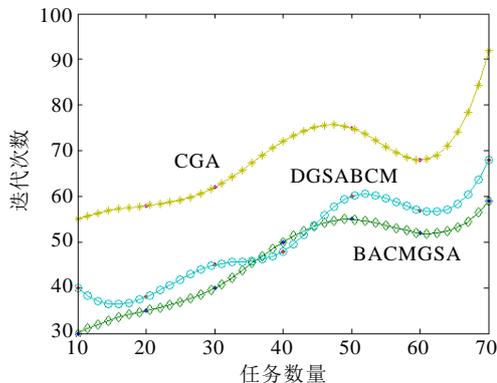


图3 BACMGSA、DGSABCM和CGA算法迭代次数随任务数量变化情况

图3给出了一定计算资源数目条件下DGSABCM、BACMGSA、CGA算法的迭代次数随任务数量增加而变化的情况, 迭代次数均随任务数量的增加而增加。经过40次的实验记录, 通过插值方法拟合出了图3所示的结果。从结果可以看出, CGA算法的迭代次数高于本文提出的BACMGSA算法。在任务数量为50~60时, 迭代次数均有下降趋势, 原因是在实验中进行了特别设置: 任务数量为10~50和60~70时, 可利用的资源数量均为30。为了验证计算资源数量对迭代次数的影响, 当任务数量在50~60变化时, 将可利用的计算资源数量设置

为20。从实验结果来看,对于BACMGSA算法,尽管任务数量的规模在扩大,但迭代次数反而降低,原因是计算资源数量降低,意味着计算节点减少,使得种群进化在相对集中的范围内进行,降低了迭代次数,说明基于自适应云模型的遗传算法的迭代次数受可用计算资源数目的影响,因此,在使用该算法调度时需要控制好计算资源的数量。

## 5 结束语

本文利用云模型云滴的随机性和稳定倾向性特点,提出了基于云模型的网格任务调度遗传算法,借助云模型解决了遗传算法中交叉和变异概率存在随机性和模糊性的问题,通过云模型将交叉与变异实现了很好的关联。实验表明,通过与CGA、SGA和AGA的结果比较可知,本文提出的算法时间复杂度低,收敛速度快,且陷入局部最优的次数少,体现了算法的可靠性、高效性。

### 参 考 文 献

- [1] DOULAMIS N D, DOULAMIS A D, VARVARIGOS E A, et al. Fair scheduling algorithms in grids[J]. IEEE Trans on Parallel and Distributed Systems, 2007, 18(11): 1638-1645.
- [2] 戴朝华, 朱云芳, 陈维荣. 云自适应遗传算法[J]. 控制理论与应用, 2007, 24(4): 645-647.  
DAI Chao-hua, ZHU Yun-fang, CHEN Wei-rong. Adaptive genetic algorithm based on cloud theory[J]. Control Theory and Applications, 2007, 24(4): 645-647.
- [3] 朱海, 王宇平. 安全驱动的实时任务调度遗传算法[J]. 系统工程与电子技术, 2010, 32(4): 855-857.  
ZHU Hai, WANG Yu-ping. Security-driven real-time task scheduling based on genetic algorithm[J]. Systems Engineering and Electronics, 2010, 32(4): 855-857.
- [4] 马学彬, 温涛. 一种基于遗传算法的网格任务调度算法[J]. 东北大学学报(自然科学版), 2007, 28(7): 974-976.  
MA Xue-bin, WEN Tao. GA-based algorithm for task scheduling on computational grid[J]. Journal of Northeastern University (Nature Science), 2007, 28(7): 974-976.
- [5] 李德毅, 杜鹁. 不确定性人工智能[M]. 北京: 国防工业出版社, 2005: 143-149.  
LI De-yi, DU Yi. Artificial intelligence with uncertainty[M]. Beijing: National Defense Industry Press, 2005: 143-149.
- [6] XIE T, QIN X. Security-aware resource allocation for real-time parallel jobs on homogeneous and heterogeneous clusters[J]. IEEE Trans on Parallel and Distributed Systems, 2008, 19(5): 692-697.
- [7] HE Xiao-chuan, JIA Yan. Leakage-aware energy efficient scheduling for fixed-priority tasks with preemption thresholds[C]//Advanced Data Mining and Applications - 4th International Conference, ADMA 2008, Proceedings. Berlin: Springer-Verlag, 2008: 379-390.
- [8] 李绍军, 张小广. 一种基于Aloperx的进化优化算法[J]. 模式识别与人工智能, 2009, 35(3): 261-265.  
LI Shao-jun, ZHANG Xiao-guang. An Aloperx based evolutionary optimization algorithm[J]. Patter Recognition and Artificial Intelligence, 2009, 35(3): 261-265.
- [9] GHATTAS R, DEAN A G. Preemption threshold scheduling: stack optimality, enhancements and analysis[C]//Real-Time and Embedded Technology and Applications Symposium. Washington, DC: IEEE Computer Society, 2007: 147-157.
- [10] 罗威, 阳富民, 庞丽萍. 异构分布式系统中实时周期任务的容错调度算法[J]. 计算机学报, 2007, 30(10): 1743-1749.  
LUO Wei, YANG Fu-min, PANG Li-ping. A real-time fault-tolerant scheduling algorithm of periodic tasks in heterogeneous distributed systems[J]. Chinese Journal of Computers, 2007, 30(10): 1743-1749.

编辑 漆蓉