

基于模糊粒子群优化的计算网格工作调度算法^{*})

王秀坤 程文树 刘洪波

(大连理工大学计算机科学与工程系 大连 116023)

摘要 网格计算是利用网络把分散的计算资源组织起来解决复杂问题的计算模式,工作调度是待解决的主要问题之一。本文提出一种基于模糊粒子群优化的网格计算工作调度算法,该算法利用模糊粒子群优化动态地产生网格计算工作调度的优化方案,使现有计算资源完成所有工作的时间最小化。实验结果表明,与基于遗传算法、模拟退火、蚁群算法的工作调度方法相比,所提出的算法在时间和精度上具有一定的优势。

关键词 网格计算,粒子群优化,遗传算法,退火算法,蚁群算法

Job Scheduling on Computational Grids Using Fuzzy Particle Swarm Optimization

WANG Xiu-Kun CHENG Wen-Shu LIU Hong-Bo

(Department of Computer Science and Engineering, Dalian University of Technology, Dalian 116023)

Abstract Computational Grids are the computing framework to meet the growing computational demands, which re-organizes the resources of many computers in the networks to solve a large of complex problems. Essential grid services contain intelligent functional mechanism for discovery, publishing of resources as well as scheduling, submission and monitoring of jobs. The paper introduces a novel approach based on fuzzy Particle Swarm Optimization (PSO) for scheduling jobs on computational grids. Our approach is to dynamically generate an optimal schedule so as to complete the tasks in a minimum period of time as well as utilizing the resources in an efficient way. We evaluate the performance of our proposed approach with a direct Genetic Algorithm (GA), Simulated Annealing (SA) and Ant Colony Algorithm (ACO) approach. The results illustrated the potentials of our approach, especially on the balance between time and precision.

Keywords Grid computing, PSO, GA, SA, ACO

1 引言

计算网格是为解决当今的一些科学难题而提出的新概念,它是一种利用互联网或专用网络把地理上广泛分布的各种计算资源互连在一起充分共享^[1,2]。作为不同计算资源的“经纪人”,计算网格服务代理提供发现和发布资源并有调度、提交和监管工作等功能,不仅有利于优化现有的计算资源,更快地解决工程和设计问题,而且能够将应用程序的每个部分调整到最适合的系统中去,从而以更短的时间、更低的成本解决整个问题^[2]。然而,计算资源通常存在不同归属,不同的所有者有其不同的访问策略,网格调度程序将保证这些资源被访问时不发生冲突,是 NP 困难问题^[3,4]。最近遗传算法被用来启发式地解决这一问题,寻求平均完成工作时间最少化的调度方案^[5,6]。计算网格中工作调度自身的难度和重要性,一些其它更好的启发式方法值得探索。粒子群优化算法是一种新兴的群体智能优化算法^[7],是对鸟群觅食过程中的迁徙和聚集的模拟,由简单个体组成的群落以及个体之间的互动行为模拟搜索最优解,在许多实际应用中是非常成功的^[8,9]。本文研究一种基于模糊粒子群优化的网格计算工作调度算法,该算法利用模糊粒子群优化动态地产生网格计算工作调度的优化方案,使现有计算资源完成所有工作的时间最少化。

2 网格工作调度问题及模糊矩阵表示

网格资源代理负责动态地发现资源,分派不同的工作到相应的网格结点,并绑定用户应用^[10]。对于工作调度问题,

考虑不同的独立工作 $J_j (j=1, 2, \dots, n)$ 分配给不同种类的网格结点 $N_i (i=1, 2, \dots, m)$ 来完成,其中每个结点有自身的计算速度,每项工作有自身的计算时间长度,每项工作必须分配到至少一个网格结点来处理。不同的工作由不同的网格节点处理,形成不同的工作调度周期,工作调度需要寻求最少完成时间和有效使用结点的调度方案。因而,网格工作调度问题可以进行如下公式化描述:定义 $C_{ij} (i=1, 2, \dots, m; j=1, 2, \dots, n)$ 作为网格结点 N_i 完成工作 J_j 所用的完成时间,用 $\sum C_i$ 代表网格结点 N_i 完成分配其上的所有工作的完成时间,则该分配的最大时间跨度 (makespan) 为 $C_{\max} = \max\{\sum C_i\}$, 该分配的工作流时间为 $\sum_{i=1}^m (\sum C_i)$, 寻求优化的调度方案过程就是使 C_{\max} 和 $\sum_{i=1}^m (\sum C_i)$ 最小化。

不失一般性,对于一个工作调度问题 $N = \{N_1, N_2, \dots, N_n\}$, $J = \{J_1, J_2, \dots, J_n\}$, 其分配关系 S 可表示为如下的模糊矩阵:

$$S = \begin{matrix} & \begin{matrix} 1 & 2 & \cdots & n \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ \vdots \\ m \end{matrix} & \begin{pmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m1} & s_{m2} & \cdots & s_{mn} \end{pmatrix} \end{matrix}$$

其中, s_{ij} 代表在域 N 中第 i 元素 N_i 与在域 J 中第 j 个元素的 J_j 分配的模糊关系度,即在 N 和 J 之间的模糊关系有

$$s_{ij} = \mu_R(N_i, J_j), i=1, 2, \dots, m; j=1, 2, \dots, n \quad (1)$$

μ_R 是成员函数,且对于 s_{ij} 必须满足以下条件:

^{*} 国家“973”重点基础研究发展规划基金项目(2002CB312003, 2005CB321904)。王秀坤 教授,博士生导师,主要研究方向为数据库技术、机器学习、数据挖掘等;程文树 硕士生,研究方向为进化计算、计算网格;刘洪波 博士,副教授,研究方向为智能计算、计算网格、数据挖掘。

$$s_{ij} \in [0, 1], \sum_{i=1}^m s_{ij} = 1$$

3 算法设计

粒子群优化作为一种新颖的群体智能技术,在许多连续问题中已经获得成功,但对离散问题特别是组合优化问题的研究做得很少,文[11]提出一种模糊离散的粒子群算法,解决 TSP 问题,为解决组合优化问题提供了一种方法。成功应用粒子群优化方法,关键在于如何将问题空间映射到粒子群优化空间并合理有效地表达粒子。假定工作和网络节点是根据工作长度和节点处理速度按升序排序,并附注相应历史信息。基于网格工作调度问题的模糊矩阵表达,粒子群算法中粒子的位置可以表示为

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

在矩阵 X 中的元素与上节中所定义的分配关系模糊矩阵 S 有相同的意义。相应地,矩阵 X 中的元素必须满足以下条件:

$$x_{ij} \in [0, 1], \sum_{i=1}^m x_{ij} = 1 \quad (2)$$

类似地,定义粒子的速度为

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{bmatrix}$$

则模糊粒子群优化算法中粒子的位置和速度的更新遵循矩阵之间的运算法则,其具体形式为

$$v(t+1) = w \otimes V(t) \oplus (c_1 * r_1) \otimes (X^{\#}(t) \ominus X(t)) \oplus (c_2 * r_2) \otimes (X^*(t) \ominus X(t)) \quad (3)$$

$$X(t+1) = X(t) \oplus V(t+1) \quad (4)$$

其中 $V(t)$, $X(t)$ 分别表示 t 时粒子速度和位置, $X^{\#}(t)$ 表示 t 时粒子当前极值, $X^*(t)$ 表示 t 时粒子群当前极值, c_1 , c_2 是学习因子, r_1 , r_2 是介于 $[0, 1]$ 之间的随机数。

粒子的位置矩阵经过一些更新处理后可能会违反公式(2)的约束,需要对其正则化,其过程是:首先将矩阵中所有的负数置零,如果矩阵某列所有元素是零,则用 0 和 1 之间的随机数重新设置该列的值,然后对整个位置矩阵做如下变换:

$$X_{\text{normal}} = \begin{bmatrix} x_{11} / \sum_{i=1}^m x_{i1} & x_{12} / \sum_{i=1}^m x_{i2} & \cdots & x_{1n} / \sum_{i=1}^m x_{in} \\ x_{21} / \sum_{i=1}^m x_{i1} & x_{22} / \sum_{i=1}^m x_{i2} & \cdots & x_{2n} / \sum_{i=1}^m x_{in} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} / \sum_{i=1}^m x_{i1} & x_{m2} / \sum_{i=1}^m x_{i2} & \cdots & x_{mn} / \sum_{i=1}^m x_{in} \end{bmatrix}$$

位置矩阵只是可行调度方案的模糊表达,在方案评估时需要按列采用最大数字法反模糊化处理,将位置矩阵解码为可行方案。粒子群优化的适应函数为

$$f(x) = \min(\max\{\sum C_i\}), i=1, 2, \dots, m \quad (5)$$

在网格环境中,对于工作调度问题需要建立工作清单和网络节点清单:

- JList₁: 所有需处理工作的清单;
- JList₂: 正在被处理工作的清单;
- JList₃: 正在待处理工作的清单 (JList₃ = JList₁ - JList₂);
- GList₁: 可利用网络节点清单;

- GList₂: 已经分配工作的网络节点清单;
- GList₃: 空闲网络节点清单 (GList₃ = GList₁ - GList₂)。

为了优化分配的最大时间跨度和 workflow 时间,当每次新的工作分配到相应的网络节点,本文采用交替地使用长工作分配到快节点规则和短工作分配到快节点规则,并启发式地选择其中之一。如果工作数量比网络节点还少,工作分配是根据先来先服务并尽可能运用长工作分配到快节点规则;如果工作数量比网络节点要多,则需要搜索最大时间跨度和 workflow 时间最小化的调度方案。具体按下列算法步骤进行:

0. 如果网格处于激活态并且 (JList₁ = 0), 而且没有新工作被提交,等待新的工作被提交,否则更新 GList₁ 和 JList₁。

1. 如果 GList₁ = 0, 工作调度处于等待状态,直到有新的可利用的网络节点资源发布;如果 JList₁ > 0, 更新 JList₂。如果 JList₂ < GList₁, 分配工作是以先来先服务为基础。如果可能,分配最长的工作到最快的网络节点上。如果 JList₁ > GList₁, 刷新 JList₂ 和 GList₃ 接, 转步骤 2;

2. 构造分配关系模糊矩阵 S ;

3. 启动模糊离散粒子群寻优进程;

3.0. 初始化粒子群的种群规模 p 和其它参数大小;

3.1. 随机初始化每个粒子的位置矩阵和速度矩阵,并正则化位置矩阵;

3.2. 如果满足结束判据, 转步骤 3.3; 否则转步骤 3.2.0;

3.2.0. $t = t + 1$;

3.2.1. 用最大数字法反模糊化解码位置矩阵为可行的分配方案,评估该分配的最大时间跨度和 workflow 时间;

3.2.2. 对每个粒子, $X_i^{\#}(t) = \operatorname{argmin}_{i=1}^n (f(X_i^{\#}(t-1)), f(X_i(t)))$;

3.2.3. 对粒子群,

$X^* = \operatorname{argmin}_{i=1}^n (f(X^*(t-1)), f(X_1(t)), f(X_2(t)), \dots, f(X_n(t)), \dots, f(X_n(t)))$;

3.2.4 根据公式(3), (4) 更新粒子的位置和速度,并正则化位置矩阵, 转步骤 3.2;

3.3 输出工作调度方案, 结束模糊粒子群寻优进程;

4. 按照工作调度方案调度网络节点和工作,更新 JList₂, JList₃, GList₂ 和 GList₃。如果有被取消分配的工作(不可行调度方案或没利用的网络节点),这些工作将被转移 JList₁ 为重新编排或将被单独处理;

5. 当网络节点被激活时, 重复步骤 0~4。

4 实验设置、结果和讨论

本节采用遗传算法(GA)、模拟退火(SA)、蚁群算法(ACO)、粒子群算法(PSO)等 4 种算法处理工作调度问题,并比较它们的性能和结果。实验参数设置如表 1 所示。为了跟踪算法性能,首先测试 3 个网络节点和 13 项工作的小规模工作调度问题即(3, 13),其结点速度分别为: $N_1 = 4$, $N_2 = 3$, $N_3 = 2$; 工作长度分别为: $J_1 = 6$, $J_2 = 12$, $J_3 = 16$, $J_4 = 20$, $J_5 = 24$, $J_6 = 28$, $J_7 = 30$, $J_8 = 36$, $J_9 = 40$, $J_{10} = 42$, $J_{11} = 48$, $J_{12} = 52$, $J_{13} = 60$ 。GA 运行 10 次的结果是 {47, 46, 47, 47.3333, 46, 47, 47, 47, 47.3333, 49}, 平均值为 47.1167; SA 运行 10 次的结果 {46.5, 46.5, 46, 46, 46, 46.6667, 47, 47.3333, 47, 47}, 具有平均的值为 46.6; ACO 运行 10 次的结果是 {46, 46, 46, 46, 46.5, 46.5, 46.5, 46, 46, 46}, 平

均值为 46.2667; PSO 运行 10 次的结果是 {46, 46, 46, 46, 46.5, 46.5, 46.5, 46, 46.5, 46.6667}, 平均值为 46.2667。所有算法求得的最好结果是 46, 图 1 是其相应的调度方案; 图 2 是 4 种算法的性能比较曲线。进一步, 测试一个规模相对较大的工作调度问题, (5,100), 其 4 种算法的性能比较曲线如图 3 所示。表 2 给出了 10 次实验平均最大时间跨度、标准方差和分配时间。图 2 和图 3 对比分析可知, 虽然 GA 处理(3,13)时最大时间跨度比 SA、ACO 要差一些, 但处理规模相对较大的(5,100)时情况却相反。结合表 2 中的结果可知, 处理(3,13)实验中, PSO 所得的最大时间跨度比 GA、SA 要好, 与 ACO 相当; 处理(5,100)实验中, PSO 所得的最大时间跨度比其它算法都要好; 值得注意的是, PSO 在分配问题处理上所消耗的时间最短。

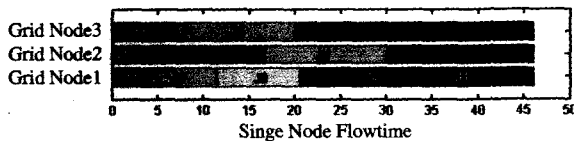


图 1 (3,13) 工作调度问题的一种调度方案

表 1 算法的参数设置

算法	参数名称	参数值
GA	种群大小	20
	交叉概率 Pc	0.8
	变异概率 Pm	0.02
SA	在温度调整前的操作次数	20
	温度降低因子	0.85
	控制步长向量	2
	初始温度	50
ACO	蚂蚁个数	5
	激素跟踪权重 α	1
	启发式信息权重 β	5
	激素蒸发参数 ρ	0.2
	激素更新常数 Q	10
PSO	种群大小	20
	自意识学习因子 c_1	1.49
	群意识学习因子 c_2	1.49
	惯性因子	0.9→0.1

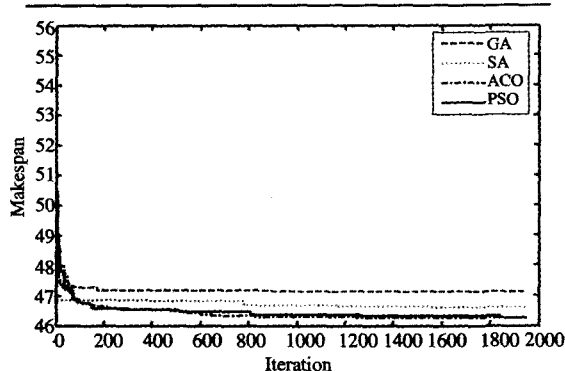


图 2 算法性能比较曲线 (3,13)

结束语 本文提出一种基于模糊粒子群优化的网格计算工作调度算法, 该算法利用模糊粒子群优化动态地产生网格计算工作调度的优化方案, 使现有计算资源完成所有工作的时间最小化。实验结果表明, 所提出的算法在处理规模相对较小的工作调度问题时, 求得的工作分配方案的最大时间跨度通常比遗传算法、模拟退火要好, 与蚁群算法相当; 在处理

规模相对较小的工作调度问题时, 求得的工作分配方案的最大时间跨度通常比其它算法都要好; 在分配问题处理上所消耗的时间最短。

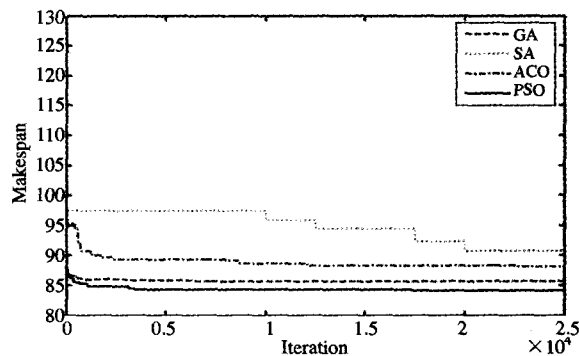


图 3 算法性能比较曲线 (5,100)

表 2 3 种算法的性能比较

算法	指标	实例	
		(3,13)	(5,100)
GA	跨度	47.1167	85.7431
	方差	± 0.7700	± 0.6217
	时间	302.9210	2415.9
SA	跨度	46.6000	90.7338
	方差	± 0.4856	± 6.3833
	时间	332.5000	6567.8
ACO	跨度	46.2667	88.1575
	方差	± 0.2854	± 0.6423
	时间	340.3750	6758.3
PSO	跨度	46.2667	84.0544
	方差	± 0.2854	± 0.5030
	时间	106.2030	1485.6

参考文献

- 1 Foster I, Kesselman C. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 2004
- 2 李维宏, 张绍华, 祝精荟. 网格 workflow 研究现状及存在问题[J]. 计算机科学, 2005, 32(11):11~15
- 3 Laforenza D. Grid programming: some indications where we are headed author [J]. Parallel Computing, 2002, 28(12):1733~1752
- 4 刘会斌, 都志辉. 网格环境下基于虚拟组织的调度模型[J]. 计算机科学, 2004, 31(9):65~67
- 5 Martino V D, Mililotti M. Sub optimal scheduling in a grid using genetic algorithms [J]. Parallel Computing, 2004, 30:553~565
- 6 Gao Y, Rong H Q, Huang J Z. Adaptive grid job scheduling with genetic algorithms [J]. Future Generation Computer Systems, 2005, 21:151~161
- 7 Kennedy J, Eberhart R. Swarm Intelligence. Morgan Kaufmann, 2001
- 8 李宁, 刘飞, 孙德宝. 基于带变异算子粒子群优化算法的约束布局优化研究[J]. 计算机学报, 2004, 27(7):897~903
- 9 Schute J F, Groenwold A A. A study of global optimization using particle swarms [J]. Journal of Global Optimization, 2005, 31:93~108
- 10 Abraham A, Buyya R, Nath B. Nature's heuristics for scheduling jobs on computational grids. In: Proceedings of the 8th IEEE International Conference on Advanced Computing and Communications (ADCOM 2000). IEEE Computer Society Press, 2000. 14~16
- 11 Pang W, Wang K P, Zhou C G, et al. Fuzzy discrete particle swarm optimization for solving traveling salesman problem. In: Proceedings of the 4th International Conference on Computer and Information Technology, IEEE Computer Society Press, 2004. 2378~2383