

不确定执行时间的云计算资源调度

李成严, 曹克翰, 冯世祥, 孙 巍

(哈尔滨理工大学 计算机科学与技术学院, 黑龙江 哈尔滨 150080)

摘 要: 针对执行时间不确定情况下的云计算资源调度问题, 基于模糊规划理论建立了时间 - 成本约束条件下的模糊云资源调度模型, 使用三角模糊数表示不确定的任务执行时间, 以最小化评价函数的平均值和不确定度作为调度目标。提出一种改进的混沌蚁群算法对模型进行求解, 算法引入精英策略优化了信息素的更新, 采用折叠次数无穷的混沌映射进行混沌搜索, 并设计了自适应混沌扰动机制以增强算法的全局搜索能力。在 Cloudsim 平台上用仿真数值实例对模型和算法进行验证, 证明了模型的可靠性, 实验结果表明改进算法在收敛速度、求解能力和负载均衡上均有较好的性能。

关键词: 云计算; 资源调度; 模糊规划; 混沌扰动

DOI: 10.15938/j.jhust.2019.01.014

中图分类号: TP399 文献标志码: A 文章编号: 1007-2683(2019)01-0085-07

Resource Scheduling with Uncertain Execution Time in Cloud Computing

LI Cheng-yan, CAO Ke-han, FENG Shi-xiang, SUN Wei

(School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China)

Abstract: For the problem of cloud computing resource scheduling, based on the fuzzy programming theory, a fuzzy cloud resource scheduling model under time-cost constraint was set up, the uncertain execution time of tasks is represented by the triangular fuzzy number, and the target is to minimize the average value and standard deviation of the evaluation function. An improved chaotic ant colony algorithm was proposed to solve the model, the elitist strategy is introduced to optimize the pheromone updating, a chaotic mapping with infinite folding times is used for chaotic search, and the adaptive chaotic disturbance mechanism is designed to enhance the global searching ability. The model and algorithm were tested on the Cloudsim platform, the reliability of the model was proved, and the experimental results showed that the proposed algorithm had better performance in convergence speed, solution ability and load balance.

Keywords: cloud computing; resource scheduling; fuzzy programming; chaotic disturbance

收稿日期: 2018-05-16

基金项目: 国家自然科学基金(61772160); 黑龙江省教育厅科学技术研究项目(12541142)。

作者简介: 曹克翰(1993—), 男, 硕士研究生;

冯世祥(1995—), 男, 硕士研究生。

通信作者: 李成严(1972—), 男, 教授, 硕士研究生导师, E-mail: chengyan@hrbust.edu.cn.

0 引言

云计算系统应用虚拟化技术,以较低的成本提供高质量的资源服务,用户根据自身需求选择并付费使用相应的服务^[1]。云计算资源调度的研究目标是合理的将多个任务分配到虚拟资源节点上执行,并满足执行时间短、所需成本低、虚拟资源集群负载均衡等条件,文[2]证明了该调度问题是一个NP完全问题。

当前云计算资源调度问题的研究主要集中于确定环境下,文[3]提出了云计算任务调度中一种基于权重的混合启发式方法。文[4]以最小化完工时间为目标,建立了云计算任务调度模型。文[5]建立了云计算环境下的多目标工作流调度模型,在降低完工时间和成本的同时,减少能耗。在以上这些研究中,均假设任务在虚拟机上的执行时间是已知的确定值,然而在实际系统中,任务在完成之前其执行时间是不确定的,称为不确定性或异步性,这是由当前的计算机任务执行特性所决定的。因此造成云资源调度的初始条件是不确定的,所以在研究资源调度问题时必须予以考虑。在引入不确定性后,对云计算资源调度问题的求解变得更加复杂,对不确定量的描述有随机变量、模糊变量、灰色变量等,模糊变量是在当概念的外延不清晰时使用的一种方法。在资源调度问题中,任务的执行时间存在不清晰的特征,因此模糊变量和模糊优化技术是一种有效的手段。文[6]研究了云计算中基于模糊的资源重调度问题,文[7]建立了以最小化作业完成时间为目标的模糊调度模型。本文使用三角模糊数表示不确定的任务执行时间,建立了模糊云资源调度问题的期望值模型。

为求解云计算资源调度问题,智能优化算法得到了较多的应用,如遗传算法^[8](GA)、粒子群算法^[9](PSO)和蚁群算法^[10](ACO),但以上算法皆有不足,因此国内外学者对它们进行了改进。文[11]在PSO中引入了贪心算法,通过贪心算法快速求解PSO的初始粒子值。文[12]将混沌系统和蚁群算法结合起来,提出了一种混沌蚁群算法(Chaotic Ant Colony Optimization Algorithm, CACO),通过混沌扰动使算法跳出局部极值区间。文[13]在蚁群系统中使用了精英策略,加快了算法收敛速度的同时提高了解的质量。为进一步优化算法的性能,本文在

CACO的基础上,引入精英策略,并设计了自适应混沌扰动机制,提出了一种基于精英策略的改进算法(chaotic ant colony optimization algorithm based on elitist strategy, ECACO)。

本文使用模糊规划的方法,在云计算资源调度问题中引入不确定的任务执行时间,建立模糊云资源调度问题的期望值模型,并提出一种改进算法对模型进行求解。本文结构如下,第1部分建立了模糊云资源调度问题的期望值模型,并对模型的求解进行了分析,第2部分介绍了算法的设计过程,第3部分通过仿真实验对模型和算法进行验证,最后作出结论。

1 模糊云资源调度问题

1.1 模糊云资源调度模型

云计算中资源调度问题可以描述为将任务集合 $T = \{t_1, t_2, \dots, t_n\}$ 中的 n 个任务,分配到虚拟机集合 $VM = \{vm_1, vm_2, \dots, vm_m\}$ 中的 m 个虚拟机上执行($m < n$)。本文研究执行时间不确定情况下的云资源调度问题,将任务在虚拟机上的执行时间定义为 $\tilde{e}_{ij}$,它是一个不确定的模糊数。则执行时间矩阵ETC和虚拟机单位时间的运行成本RCU为

$$RCU = \{rcu_1, rcu_2, \dots, rcu_m\} \quad (1)$$

$$ETC = \begin{bmatrix} \tilde{e}_{11} & \tilde{e}_{12} & \dots & \tilde{e}_{1n} \\ \tilde{e}_{21} & \tilde{e}_{22} & \dots & \tilde{e}_{2n} \\ \vdots & \vdots & & \vdots \\ \tilde{e}_{m1} & \tilde{e}_{m2} & \dots & \tilde{e}_{mn} \end{bmatrix} \quad (2)$$

其中, rcu_i 为 vm_i 单位时间内执行任务消耗的资源成本。在一个完整的资源调度方案 P 中,虚拟机 vm_i 执行任务所需时间 $vmTime_i$ 和成本 $vmCost_i$ 分别为

$$vmTime_i = \sum_{j=1}^n d_j \times \tilde{e}_{ij}, d_j \in vmTask_i \quad (3)$$

$$vmCost_i = vmTime_i \times rcu_i \quad (4)$$

其中, $vmTask_i$ 为分配在 vm_i 上执行的任务集合。执行调度方案 P_i 需要的总时间为

$$Time(P_i) = \max \{vmTime_1, vmTime_2, \dots, vmTime_m\} \quad (5)$$

执行任务的总成本为

$$Cost(P_i) = \sum_{i=1}^m vmCost_i \quad (6)$$

在云环境中,资源的调度需要考虑多方面的因素,任务完成时间决定了客户的满意度,而资源提供

商希望尽可能的降低成本,因此提出一个时间 - 成本约束条件,将两者同时纳入考虑范围。其中,时间约束函数为

$$rTime(P_i) = \frac{Time(P_i) - Time_{MIN}}{Time_{MAX} - Time_{MIN}} \quad (7)$$

成本约束函数为

$$rCost(P_i) = \frac{Cost(P_i) - Cost_{MIN}}{Cost_{MAX} - Cost_{MIN}} \quad (8)$$

其中, $Time_{MAX}$ 、 $Time_{MIN}$ 为任务在性能最差、最好的机器上运行所需要的时间, $Cost_{MAX}$ 、 $Cost_{MIN}$ 为任务在执行时所需的最高、最低成本。 $rTime(P_i)$ 和 $rCost(P_i)$ 的值越小, 执行任务所需的时间和成本就越少。

在时间 - 成本选择约束条件下的云计算资源调度模型为

$$\min\{res(P_i)\} \quad (9)$$

其中, 评价函数

$$res(P_i) = t * rTime(P_i) + c * rCost(P_i) \quad (10)$$

由于任务在虚拟机上的执行时间是模糊数, 所以评价函数值也是模糊数。目标函数是使基于时间 - 成本约束条件的评价函数的期望值最小。其中: t 是时间因子, c 是成本因子, 且 $t + c = 1$ 。可以通过调节 t 和 c 的取值来影响资源选择时的倾向, 当 $t = 1$, $c = 0$ 时, 调度的目标为最小化任务完成时间; 当 $t = 0$, $c = 1$ 时, 调度的目标为最小化成本; 当 $t = 0.5$, $c = 0.5$ 时, 调度的目标为使任务完成时间较短的同时, 令成本也较低。

定义虚拟资源的负载均衡函数为

$$Load = \frac{\min_{1 \leq i \leq m} vmTime_i}{\max_{1 \leq i \leq m} vmTime_i} \quad (11)$$

式中, $\min_{1 \leq i \leq m} vmTime_i$ 表示调度方案中虚拟机的最短执行时间, $\max_{1 \leq i \leq m} vmTime_i$ 为虚拟机最长执行时间, 即调度方案总执行时间。由式(11)可知

1) 若 $\max_{1 \leq i \leq m} vmTime_i = 0$, 则任务还未被执行;

2) 若 $Load = 0$ 且 $\max_{1 \leq i \leq m} vmTime_i \neq 0$, 此时有空闲虚拟机;

3) 若 $Load = 1$, 则各虚拟机的执行时间相同, 此时系统的负载均衡性达到最佳, $Load$ 的值越接近 1, 证明系统对资源的利用越充分。

1.2 模型转换

本文使用三角模糊数来表示任务在虚拟机上不确定的执行时间 $\tilde{e}_{ij}$, 三角模糊数 $\tilde{R} = (r^L, r^M, r^R)$ 的隶属度函数图如图 1 所示, 其中 r^L 、 r^M 、 r^R 代表了 3

个端点值。应用三角模糊数建立模糊模型后, 在求解模型时需要使用模糊运算。

在对模糊云资源调度模型的求解中, 涉及到了模糊加法、模糊乘法和模糊取大三种运算。例如将任务分配到一个虚拟机上执行时, 会累积该虚拟机的运行时间和成本, 此时需要进行模糊加法运算, 计算虚拟机的成本消耗时, 会进行模糊乘法运算, 而在求解任务总执行时间, 即计算式(5)时, 需要进行模糊取大运算。文[4]定义了这三种运算。

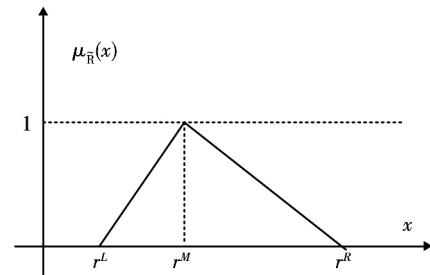


图 1 三角模糊数的隶属函数

Fig. 1 Membership function of triangular fuzzy number

取任意两个三角模糊数 $\tilde{X}$ 、 $\tilde{Y}$ 和任意实数 λ , 对它们执行模糊运算的结果为模糊加法:

$$\tilde{X} + \tilde{Y} = (x^L + y^L, x^M + y^M, x^R + y^R) \quad (12)$$

模糊乘法:

$$\lambda \tilde{X} = \begin{cases} (\lambda x^L, \lambda x^M, \lambda x^R), & \lambda \geq 0 \\ (\lambda x^R, \lambda x^M, \lambda x^L), & \lambda < 0 \end{cases} \quad (13)$$

模糊取大:

$$\tilde{X} \vee \tilde{Y} = (x^L \vee y^L, x^M \vee y^M, x^R \vee y^R) \quad (14)$$

记 $\tilde{Z} = res(P_i)$, 因模糊运算具有可分解性, 所以 $\tilde{Z}$ 的 3 个端点 Z^L, Z^M, Z^R 只和模糊执行时间 $\tilde{e}_{ij}$ 的 3 个端点 $\tilde{e}_{ij}^L, \tilde{e}_{ij}^M$ 和 $\tilde{e}_{ij}^R$ 相关, 可按照模型依次求解。则式(9)可转化为

$$\min\{res(P_i)\} = \min\{\tilde{Z}\} = \min\{Z^L, Z^M, Z^R\} \quad (15)$$

云计算资源调度问题是一个在众多调度方案中选取最优方案的决策问题, 本文的调度目标是最小化评价函数期望值。不同于确定模型中可以直接对确定值排序, 模糊数之间的排序较为复杂。文[5]提出了一种较好的三角模糊数排序方法, 使用式(16)和(17)计算三角模糊数的平均值和标准差, 若一个模糊数具有较高的平均值和较低的标准差, 则认为该模糊数排序更高。

$$\bar{x}_p(\tilde{X}) = \frac{1}{4}(X^L + 2X^M + X^R) \quad (16)$$

$$\sigma_p(\tilde{X}) = \frac{1}{80} [\tilde{B}(X^L)^2 + 4(X^M)^2 + 3(X^R)^2 - 4X^L X^M - 2X^L X^R - 4X^M X^R]^{\frac{1}{2}} \quad (17)$$

根据以上方法,可将模糊云资源调度模型转化为时间-成本约束条件下的单目标规划模型,调度的目标转化为最小化评价函数的平均值和标准差,式(9)可进一步转化为

$$\min\{res(P_i)\} = \min\{\tilde{Z}\} = \min\{Z^L, Z^M, Z^R\} = \min\{Z_\eta + \theta Z_\mu\}, \theta \in [0, 1] \quad (18)$$

其中: Z_η 为模糊数 $\tilde{Z}$ 的平均值; Z_μ 为标准差; θ 是对不确定度的加权系数。

2 算法设计与分析

2.1 精英策略

在蚁群算法中引入精英策略,可以提高算法的局部搜索能力和收敛速度,使蚁群可以在较少的迭代后找到更优的解。但如果精英蚂蚁过多,解元素的差异就会变小,这会直接影响蚂蚁对路径的选择,导致算法停滞。对此,本文基于排序机制提出一种改进策略:在当前循环结束后,根据生成方案的总执行时间,对所有蚂蚁进行排序,选取一部分执行时间较短的蚂蚁作为精英蚂蚁,对它们留下的信息素增量进行一次更新。引入精英策略后,蚁群的信息素更新公式如下:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \quad (19)$$

$$\Delta\tau_{ij} = \begin{cases} \sum_{k=1}^m [\Delta\tau_{ij}^k \cdot (1-\rho) + \Delta\tau_{ij}^*], & \text{case1} \\ \sum_{k=1}^m \Delta\tau_{ij}^k, & \text{case2} \\ 0, & \text{otherwise} \end{cases} \quad (20)$$

case 1: k 是精英蚂蚁且将 t_j 分配给 vm_i

case 2: k 不是精英蚂蚁且将 t_j 分配给 vm_i

$$\Delta\tau_{ij}^* = \frac{Time(P_{ave}) - Time(P_k)}{Time(P_{ave}) - Time(P^*)} \cdot \frac{Q}{Time(P_k)} \quad (21)$$

其中: ρ 为信息素挥发系数; $\tau_{ij}(t)$ 表示在 t 时刻路径 (vm_i, t_j) 上的信息素浓度; $\Delta\tau_{ij}^k$ 表示第 k 只蚂蚁在本次迭代中留在路径 (vm_i, t_j) 上的信息素增量; $\Delta\tau_{ij}^*$ 为精英蚂蚁的信息素增量; $Time(P_{ave})$ 为当前循环中所有蚂蚁生成方案的平均执行时间。

2.2 自适应混沌扰动

混沌系统是指在一个确定性系统中,存在貌似随机的不规则运动,其行为表现为不确定、不可重复

和不可预测,称此现象为混沌现象。混沌运动具有随机性、遍历性和对初始条件敏感的特性,利用这些特性,可以优化搜索。Lyapunov 指数是用于衡量一个映射混沌性的重要参照,在某种意义上,它可以看作是映射平均折叠次数的一种表示。常用的 Logistic 映射和 Tent 映射产生的混沌序列虽然具有较好的混沌性,但它们的映射折叠次数都是有限的。本文采用的是由式(22)定义的无限折叠映射:

$$x_{n+1} = \sin\left(\frac{2}{x_n}\right), n=0, 1, 2, \dots, x_0 \in [-1, 1] \quad (22)$$

当迭代初值 x_0 不为 0 时,系统处于混沌状态(实际上初值也不可以为不动点,但上式中的不动点皆为超越数,因此不作考虑)。无限折叠映射的 Lyapunov 指数表达式为

$$\begin{aligned} \lambda &= \lim_{n \rightarrow \infty} \left(\frac{1}{n} \sum_{i=0}^{n-1} \ln |f'(x_i)| \right) \\ &= \lim_{n \rightarrow \infty} \left(\frac{1}{n} \sum_{i=0}^{n-1} \ln \left| \frac{2}{x_i^2} \cos\left(\frac{2}{x_i}\right) \right| \right) \end{aligned} \quad (23)$$

与 Logistic 映射和 Tent 映射相比,折叠次数为无穷大的无限折叠映射显然具有更好的混沌性。在混沌搜索过程中,使用无限折叠映射得到的解会更均匀,以此生成的启发式信息质量更高,将可以更好的指导蚁群的搜索。

蚁群算法容易陷入局部最优解,而精英策略的引入一定程度上放大了这个缺点,因此引入混沌搜索和混沌扰动对其进行优化。在算法初始化过程中,使用混沌搜索来达到全局搜索的目的,然后选择较优路径生成后发性信息,以此初始化蚁群信息素矩阵,指引后续搜索。当蚁群经过一定次数的迭代后,搜索到的最优解变化量不大于一个判定量时,认为算法陷入停滞状态,对当前最优解路径上的信息素量进行混沌扰动。设判定蚁群停滞的迭代次数和判定量分别为 γ 和 δ (δ 为一极小常数),当满足 $|f_t - f_{t+\gamma}| \leq \delta$ (f_t 和 $f_{t+\gamma}$ 为第 t 次迭代和第 $t+\gamma$ 次迭代的最优解)时,蚁群的信息素更新公式调整为

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} + \omega c_{ij}^{\gamma+1} \quad (24)$$

$$c_{ij}^{\gamma+1} = (1-\omega)c_{ij}^{\gamma} + x_{ij} \quad (25)$$

式中: ω 为混沌扰动系数; c_{ij}^{γ} 为蚁群停滞后第 γ 次迭代的混沌扰动量; x_{ij} 由混沌映射公式迭代得到。蚁群停滞的时间越长,混沌扰动的强度就越大,如果经过长时间的扰动后,最优解依然没有变化,则认为已找到最优解,算法结束。

ECACO 算法的流程图如图 2 所示。

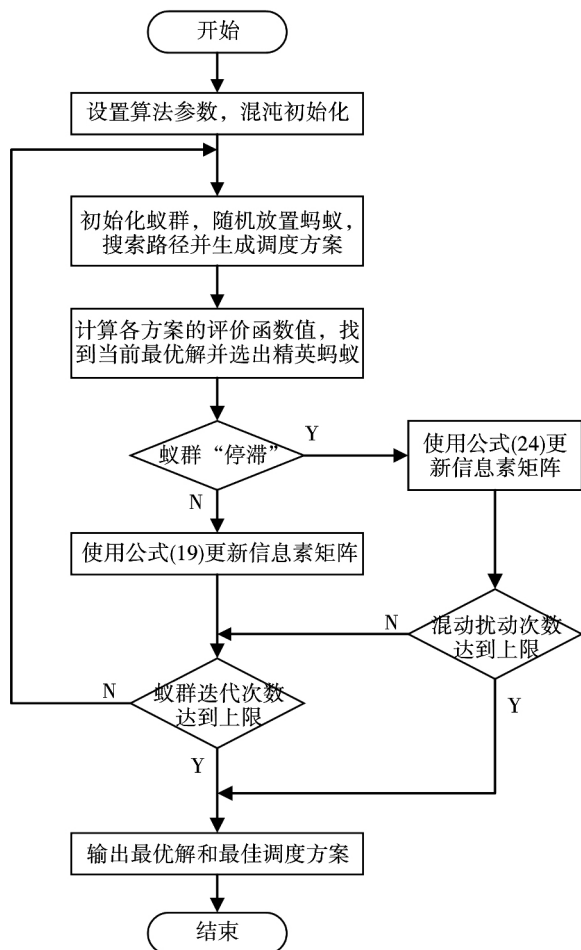


图 2 算法流程图

Fig. 2 Process of algorithm

3 仿真实验

3.1 算法参数设置

为验证本文提出的模型和算法,在云仿真平台 Cloudsim 上进行仿真实验。因目前没有适合云计算资源调度问题的标准数据集,本文采用文 [6] 的方法,在区间 $[50000, 150000]$ 和区间 $[500, 1500]$ 内随机生成任务的大小和虚拟机处理速度,以此作为仿真实验的数据。在实验中,要将每一个确定的执行时间 e_{ij} 转化为三角模糊数,其左端点在区间 $[\xi_1 e_{ij}, e_{ij}]$ 内随机生成 ($0 < \xi_1 < 1$),右端点在区间 $[e_{ij}, \xi_2 e_{ij}]$ 内随机生成 ($\xi_2 > 1$),在以下实验中令 $\xi_1 = 0.9, \xi_2 = 1.2$ 。经过多次重复实验得知,ACO、CACO 和 ECACO 都将在 300 次迭代中收敛,因此设置算法最大迭代次数为 300 次。对于 ECACO,在蚁群陷入停滞状态后,若进行 50 次混沌扰动后最优解依然不变,则直到算法迭代结束,当前最优解都将保持不变。即可将混沌扰动 50 次后依然不变的解视为最终解,因

此设置最大混沌扰动次数为 50 次。

加权系数 ∂ 是用来调整调度时对模糊评价函数的平均值和不确定度的侧重, ∂ 越大,调度对问题的不确定性越重视。 ∂ 的取值应由具体的应用环境来确定,若在网络传输稳定、虚拟机处理速度较慢这种时间代价较高的环境中,不确定性对调度的影响较小, ∂ 应取较小的值。而在不确定性大的环境中, ∂ 应取较大的值。在本文的对比实验中, ∂ 的取值为 0.5,虚拟机的数量为 8 个,算法的主要参数设置如表 1 所示。

表 1 实验运行环境

Tab. 1 Experimental environment

参数	取值
∂	0.5
ρ	0.5
γ	20
ω	0.2
混沌扰动次数上限	50
蚂蚁数量	50
迭代次数上限	300

3.2 确定模型与模糊模型对比

为了对任务执行时间确定和不确定两种情况下的模型进行比较,在不同任务规模下,使用 ECACO 算法对两种模型进行求解。在模型侧重方向的选择中,令评价函数中时间的权重和成本的权重相同,以便同时比较时间和成本的变化,因此设置时间因子 $t = 0.5$,成本因子 $c = 0.5$ 。确定云资源调度模型和模糊云资源调度模型之间的对比结果如图 3 所示。可以看出,不确定因素的存在提高了评价函数的值,这意味着任务的执行时间和所需成本会增加,因此必须在调度时考虑不确定性因素的影响,否则会导致理论与实际的偏差过大而降低系统的效率。

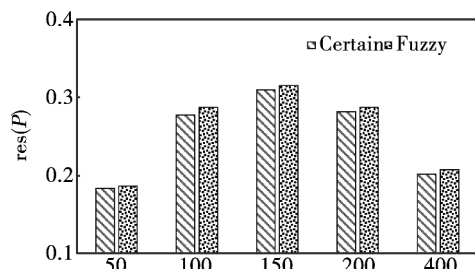


图 3 模型对比

Fig. 3 Comparison of models

3.3 算法性能对比

在相同参数下,对 ACO、CACO 和 ECACO 进行实验,图 4、5、6 分别为在任务数量为 100 的情况下,

当 $t=0.5, c=0.5$ 时, $t=1, c=0$ 时, 以及 $t=0, c=1$ 时算法的迭代过程。可以看出, 不管时间因子和成本因子如何取值, 当使用 ECACO 求解模型时, 得到的调度方案都能够在时间和成本的最小化上获得更好的结果。图 7 是任务数量为 400, $t=0.5, c=0.5$ 时算法的迭代过程, 可以看出, 在大规模的任务调度上, ECACO 依然表现出了良好的性能。观察 3 种算法的迭代曲线, 可知当蚁群陷入停滞状态后, ECACO 的扰动机制可以使蚁群更快的跳出局部极值区间, 找到更优解。相比较于 CACO, 本文提出的 ECACO 具有更快的收敛速度和更好的全局搜索能力。除此之外, ECACO 能够判断出蚁群是否获得稳定最优解, 并且在获得稳定解后提前停止迭代, 因此在一定程度上减少了算法迭代的时间, 提高了系统的整体效率。

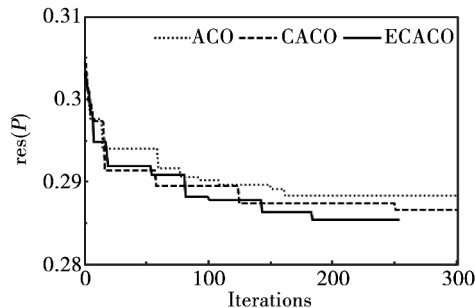


图 4 $t=0.5, c=0.5, 100\text{task}-8\text{vm}$

Fig. 4 $t=0.5, c=0.5, 100\text{task}-8\text{vm}$

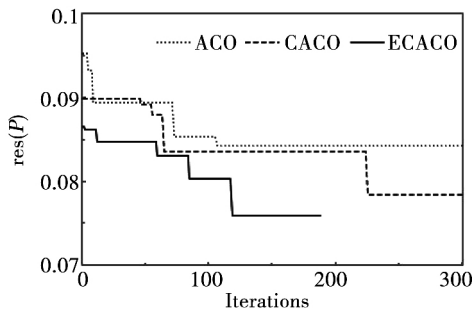


图 5 $t=1, c=0, 100\text{task}-8\text{vm}$

Fig. 5 $t=1, c=0, 100\text{task}-8\text{vm}$

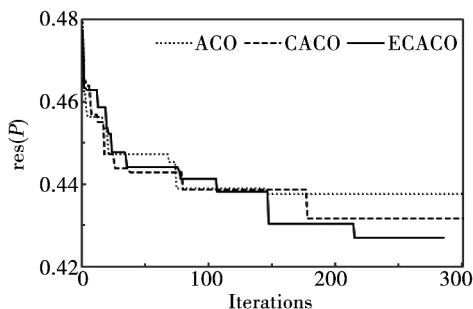


图 6 $t=0, c=1, 100\text{task}-8\text{vm}$

Fig. 6 $t=0, c=1, 100\text{task}-8\text{vm}$

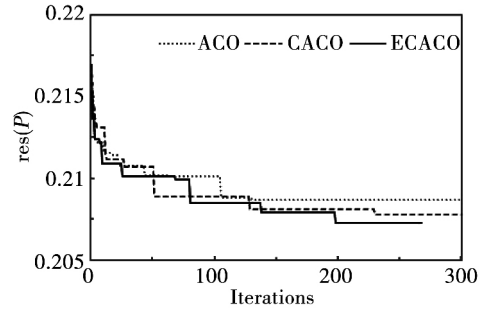


图 7 $t=0.5, c=0.5, 400\text{task}-8\text{vm}$

Fig. 7 $t=0.5, c=0.5, 400\text{task}-8\text{vm}$

当 $t=0.5, c=0.5$ 时, 使用 3 种算法对不同任务规模下的模型进行多次求解, 对得到的解取平均值, 结果如表 2 所示, 相对差值列为 CACO 的评价函数数值大于 ECACO 评价函数数值的相对平均值。表 2 表明, 无论是小规模的资源调度还是大规模的资源调度, 使用 ECACO 都能够获得更好的解。图 8 为 $t=0.5, c=0.5$ 时, 不同任务规模下 3 种算法的负载均衡程度对比。可以看出, ECACO 的负载均衡程度明显高于 ACO 和 CACO, 这证明 ECACO 对于资源的利用率更高, 可以有效避免虚拟机上的工作负载过重。

表 2 不同任务规模下的评价函数值

Tab. 2 Evaluation function values under different task numbers

任务规模	ACO	CACO	ECACO	相对差值 /%
50	0.188 214	0.187 141	0.186 652	0.261
100	0.288 158	0.286 958	0.286 517	0.154
150	0.316 821	0.316 081	0.315 64	0.140
200	0.288 200	0.287 082	0.286 796	0.100
400	0.208 680	0.207 768	0.207 487	0.135

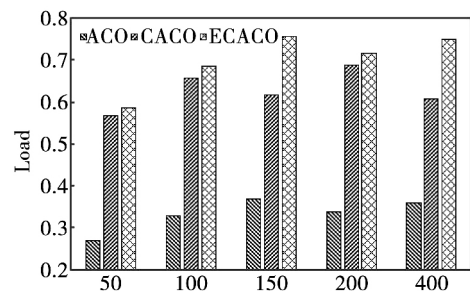


图 8 不同任务规模下的负载均衡程度

Fig. 8 Load balancing degree under different task numbers

通过以上实验可知,本文提出的 ECACO 收敛速度快、最优解搜索能力强,使用 ECACO 对模糊云计算资源调度模型进行求解,可以实现任务完成时间更短、成本消耗更低并且虚拟机负载更加均衡的目标,从而提高云计算系统的综合效率。

4 结 论

本文使用模糊变量表示不确定的任务执行时间,建立了云计算资源调度问题的模糊规划模型,并提出一种改进算法进行求解。ECACO 引入了精英策略增强蚁群的局部搜索能力,然后通过无限折叠映射生成启发式信息和混沌扰动量,并优化了混沌扰动机制,让蚁群在停滞后可以更快的跳出局部极值区间。实验证明 ECACO 具有更好的性能,能够进一步提高云计算系统的效率。不确定因素的存在会使任务的执行时间和成本增加,因此在调度时必须考虑。相对于以往研究,本文提出的模型将不确定因素纳入了研究范围,更符合实际环境中的应用。

参 考 文 献:

- [1] JULA A, SUNDARARAJAN E, OTHMAN Z. Cloud Computing Service Composition: A Systematic Literature Review [J]. Expert Systems with Applications, 2014, 41(8): 3809.
- [2] RIMAL B P, JUKAN A, KATSAROS D, et al. Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach [J]. Journal of Grid Computing, 2011, 9(1): 3.
- [3] ABDULLAHI M, NGADI M A, ABDULHAMID S M. Symbiotic Organism Search Optimization Based Task Scheduling in Cloud Computing Environment [J]. Future Generation Computer Systems-The International Journal of Esience, 2016, 56: 640.
- [4] YAO G S, DING Y S, JIN Y C, et al. Endocrine-based Coevolutionary Multi-swarm for Multi-objective Workflow Scheduling in a Cloud System [J]. Soft Computing, 2017, 21(15): 4309.
- [5] KAMALINIA A, GHAFARI A. Hybrid Task Scheduling Method for Cloud Computing by Genetic and DE Algorithms [J]. Wireless Personal Communications, 2017, 97(4): 6301.
- [6] KIM J, KIM T, PARK M, et al. Fuzzy-Based Resource Reallocation Scheduling Model in Cloud Computing [J]. Lecture Notes in Electrical Engineering, 2014, 301: 43.
- [7] SHOJAFAR M, JAVANMARDI S, ABOLFAZLI S. FUGE: A Joint Meta-heuristic Approach to Cloud Job Scheduling Algorithm Using Fuzzy Theory and a Genetic Method [J]. Cluster Computing-The Journal of Networks Software Tools and Applications, 2015, 18(2): 829.
- [8] HASSAN M A, KACEM I, MARTIN S, et al. Genetic Algorithms for Job Scheduling in Cloud Computing [J]. Studies in Informatics & Control, 2015, 24(4): 387.
- [9] SADHASIVAM N, THANGARAJ P. Design of an Improved PSO Algorithm for Workflow Scheduling in Cloud Computing Environment [J]. Intelligent Automation & Soft Computing, 2016, 31(8): 493.
- [10] HU X X, ZHOU X W. Improved Ant Colony Algorithm on Scheduling Optimization of Cloud Computing Resources [J]. Applied Mechanics & Materials, 2014, 678: 75.
- [11] ZHONG Z F, CHEN K, ZHAI X J, et al. Virtual Machine-Based Task Scheduling Algorithm in a Cloud Computing Environment [J]. Tsinghua Science and Technology, 2016, 21(6): 660.
- [12] MA Y, WANG Y. Grid Task Scheduling Based on Chaotic Ant Colony Optimization Algorithm [C] // International Conference on Computer Science and Network Technology. IEEE, 2013: 469.
- [13] YOUSEFIKHOSHBAKHT M, DIDEHVAR F, RAHMATI F. An Efficient Solution for the VRP by Using a Hybrid Elite Ant System [J]. International Journal of Computers Communications & Control, 2014, 9(3): 340.
- [14] BENTRACIA T, MOUSS L H, MOUSS N K, et al. Evaluation of Optimality in the Fuzzy Single Machine Scheduling Problem Including Discounted Costs [J]. International Journal of Advanced Manufacturing Technology, 2015, 80(5-8): 1369.
- [15] BALIN S. Non-identical Parallel Machine Scheduling with Fuzzy Processing Times Using Genetic Algorithm and Simulation [J]. International Journal of Advanced Manufacturing Technology, 2012, 61(9-12): 1115.
- [16] CALHEIROS R N, RANJAN R, BELOGLAZOV A, et al. CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms [J]. Software Practice & Experience, 2010, 41(1): 23.
- [17] PRIYA V, BABU C N K. Moving Average Fuzzy Resource Scheduling for Virtualized Cloud Data Services [J]. Computer Standards & Interfaces, 2016, 50: 251.
- [18] 罗智勇, 朱梓豪, 尤波, 等. 基于串归约的时间约束下 workflow 精确率优化算法 [J]. 哈尔滨理工大学学报, 2018, 23(5): 68.
- [19] LU D, MA J, SUN C, et al. Credit-based Scheme for Security-aware and Fairness-aware Resource Allocation in Cloud Computing [J]. Science China-Information Sciences, 2017, 60(5): 052103.
- [20] 赵辉, 吕青, 丁树业. 模糊综合评判在优化电机冷却系统中的应用 [J]. 哈尔滨理工大学学报, 2016, 21(6): 106.
- [21] ZHANG Y, SUN J. Novel Efficient Particle Swarm Optimization Algorithms for Solving QoS-demanded Bag-of-tasks Scheduling Problems with Profit Maximization on Hybrid Clouds [J]. Concurrency and Computation-Practice & Experience, 2017, 29(21): 4249.

(编辑: 温泽宇)