

云环境中面向可靠性约束的工作流调度策略研究

李金亮^{1,2} 林 兵^{2,3} 陈 星^{1,2}

1 福州大学计算机与大数据学院 福州 350108

2 福建省网络计算与智能信息处理重点实验室 福州 350108

3 福建师范大学物理与能源学院 福州 350117

(211027178@fzu.edu.cn)

摘 要 随着越来越多的计算密集型依赖应用被卸载到云环境中执行,工作流调度问题受到了广泛的关注。针对云环境多目标优化的工作流调度问题,考虑到任务执行过程中服务器可能会发生性能波动和宕机等问题,基于模糊理论,使用三角模糊数表示任务执行时间和数据传输时间,提出了一种基于遗传算法的自适应粒子群优化算法(Adaptive Particle Swarm Optimization based GA, APSOGA),目的是在工作流的可靠性约束下,综合优化工作流的完成时间和执行代价。该算法为了避免传统粒子群优化算法存在的过早收敛问题,引入了遗传算法的随机两点交叉操作和单点变异操作,有效地提升了算法的搜索性能。实验结果表明,与其他策略相比,基于 APSOGA 的调度策略能够有效地降低云环境中面向可靠性约束的科学工作流的模糊总代价。

关键词: 云计算;可靠性约束;不确定性;多目标优化;三角模糊数

中图法分类号 TP393

Reliability Constraint-oriented Workflow Scheduling Strategy in Cloud Environment

LI Jinliang^{1,2}, LIN Bing^{2,3} and CHEN Xing^{1,2}

1 College of Computer and Data Science, Fuzhou University, Fuzhou 350108, China

2 Fujian Provincial Key Laboratory of Networking Computing and Intelligent Information Processing, Fuzhou 350108, China

3 College of Physics and Energy, Fujian Normal University, Fuzhou 350117, China

Abstract As more and more computationally intensive dependent applications are offloaded to the cloud environment for execution, the problem of workflow scheduling has received extensive attention. Aiming at the workflow scheduling problem of multi-objective optimization in cloud environment, and considering that the server may experience performance fluctuations and downtime during task execution, based on fuzzy theory, a triangular fuzzy number is used to represent task execution time and data transmission time. A genetic algorithm-based adaptive particle swarm optimization based GA (APSOGA) is proposed. The purpose is to comprehensively optimize the completion time and execution cost of the workflow under the reliability constraints of the workflow. In order to avoid the premature convergence problem of the traditional particle swarm optimization algorithm, the proposed algorithm introduces the random two-point crossover operation and single-point mutation operation of the genetic algorithm, which effectively improves the search performance of the algorithm. Experimental results show that, compared with other strategies, APSOGA-based scheduling strategy can effectively reduce the time and cost of reliability-constrained scientific workflows in cloud environments.

Keywords Cloud computing, Reliability constraints, Uncertainty, Multi-objective optimization, Triangular fuzzy number

1 引言

由于云数据中心具有丰富的计算资源和储存资源,因此经常被用来为终端用户提供服务并解决用户终端设备计算资源不足的问题。云计算技术能够根据用户的需求,弹性地为用户按需提供资源,可以比以往更快地执行现实世界的应用

程序,这对天文、高能物理、生物信息学和地震科学等计算密集型应用至关重要。这些复杂的计算密集应用是由成百上千个相互依赖的任务组成,常被构建为工作流模型^[1]。事实上,工作流的调度问题至关重要,调度结果的好坏会直接影响工作流模型的完成时间和执行代价,尤其是在合理的时间以及预算范围内完成工作流仍然是一项严峻的挑战^[2]。

到稿日期:2022-08-03 返修日期:2022-12-24

基金项目:国家自然科学基金(62072108);福建省自然科学基金杰青项目(2020J06014);福建省高校产学研合作项目(2022H6024)

This work was supported by the National Natural Science Foundation of China(62072108), Natural Science Foundation of Fujian Province for Distinguished Young Scholars(2020J06014) and Fujian University Industry-University Cooperation Project(2022H6024).

通信作者:林兵(WheelLX@163.com)

workflow调度问题的目标是为工作流的每个任务选择合适的计算资源,同时满足用户的需求。现有工作流的调度研究大多都是基于截止日期约束下执行代价最优或预算约束下完成时间最短。首先,这些研究没有考虑到现实环境中服务器可能会发生宕机等,导致工作流无法在既定的时间内完成;其次,单一目标的优化往往不能很好地解决用户的需求,在优化完成时间时,为了使完成时间更短,服务器租用成本往往会更高;在优化执行代价时,为了使代价更低,往往会追求租用价格便宜但处理速度更慢的服务器。这是由服务器的特性导致的,通常性能更好的服务器价格更贵。

同时,现有的关于云环境中的工作流调度问题大多都基于这样一个假设^[3],即每个工作流在特定类型虚拟机上的任务执行时间是确定的,并且可以事先准确计算。然而,由于现实世界的服务器无法保持持久的性能,实际的任务执行时间可能会产生波动^[4],同时影响到其子任务。由于现实环境的服务器无法一直按照既定的状态执行任务,因此,需要有一个高效的调度策略,能够在考虑现实环境中服务器的性能和宕机等可能出现的问题的同时优化多个 Qos。

基于上述考虑,本文提出了在工作流的总体可靠性约束下同时优化工作流的完成时间和执行时间的工作流调度模型。受文献^[5-6]的影响,本文使用三角模糊数来表示不确定性云环境中任务的计算时间和传输时间,同时提出了一种基于遗传算法的自适应粒子群算法,在满足工作流的总体可靠性约束下,优化工作流的模糊完成时间和模糊执行代价。

2 相关工作

近年来,由于云计算资源能够根据用户的需求弹性地为用户提供按需计算资源^[7-8],在云环境中工作流的调度问题得到了广泛研究。工作流调度问题是一个著名的 NP-hard 问题,因此通常用启发式策略来解决这些问题。经典的启发式解决方案是由 Wang 等^[9]提出的结合 PSO 和空闲时隙感知的工作流调度方法,该方法期望在截止日期约束下最小化工作流的执行成本。Li 等^[10]考虑到云服务提供商需要为多个具有不同 Qos 需求的用户提供服务,提出了一种基于群体协同进化的混合智能优化算法,实验结果表明该算法优于现有的调度算法。Faragardi 等^[11]考虑到不同类型的资源实例可能具有不同的性价比,提出了性价比优先的资源配置方法,在满足预算的约束下,提出了改进的 HEFT 算法 (GRP-HEFT),实验结果表明,GRP-HEFT 在不同规格的多个著名科学工作流应用中的平均性能都优于 PSO 算法和 GA 算法。Wang 等^[12]研究截止日期约束下优化工作流的完成时间,提出了一种免疫粒子群优化算法 (IMPSO),实验结果表明该方法能够有效地提高算法优化的质量和速度,解决了 PSO 存在的过早收敛问题。

然而,现有的大多数研究工作都是假设任务在服务器上的执行时间是确定的,现实环境中,这种任务在服务器上的执行时间和传输时间预先定义的情况并不符合真实情况。由于服务器既无法保持长久的性能也不能保证服务过程中不出现故障等现象,任务在服务器上的执行时间和传输时间必然会受到影

响。执行时间是不确定的,因此提出了两种考虑任务执行时间不确定性和能量感知的调度方法。Pham 等^[14]考虑到云资源性能的波动性,使用实现率和中断率进行资源建模,并提出了一种新的多目标工作流调度方法,有效地减少了工作流的执行时间和成本。Cao 等^[15]提出了一种不确定性感知资源供应方法,用于在基于软件定义网络的边缘计算环境中对工作流进行调度。这类研究工作考虑到了服务器资源的不确定性,但没有考虑到这种不确定性如何对工作流的执行时间和传输时间造成影响。现有的不确定计算的环境下的模糊调度研究主要面向智能制造系统。Sun 等^[6]使用三角模糊数表示处理时间,同时提出了一种模糊化方法,将数据集中的处理时间模糊化为三角模糊数,研究模糊作业车间调度问题。然而,模糊云环境中工作流的调度研究问题仍然是一个待解决的问题。

本文考虑的是不确定的执行时间和传输时间下的工作流调度问题,基于模糊理论,利用三角模糊数将确定的执行时间和传输时间转换为不确定的执行时间和传输时间;同时,考虑到服务器宕机等可能发生的状况,设置了工作流的总体可靠性,在可靠性约束下,对完成时间及执行代价进行优化^[16]。本文的研究更具有现实意义。

3 问题定义

工作流调度模型框架主要由 3 部分组成:云环境资源、带可靠性约束的工作流,以及调度器^[17]。如图 1 所示。

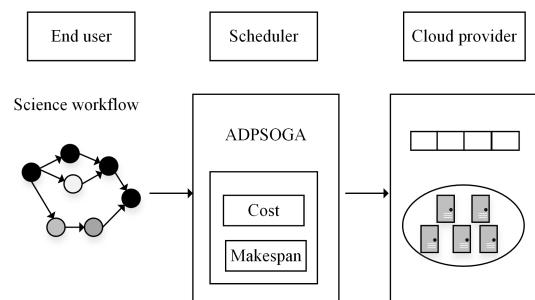


图 1 调度模型框架

Fig. 1 Scheduling model framework

3.1 确定性云环境中的工作流调度

工作流用有向无环图 DAG 的形式来表示^[18],即 $G = \langle T, E, D \rangle$ 。其中, T 表示一组节点, $T = \{t_1, t_2, \dots, t_n\}$, 每个节点都是一个任务; E 表示任务之间的一组边, $E = \{e_{1,2}, e_{1,3}, \dots, e_{i,j}\}$, 表示任务间的控制或数据依赖关系, 任务 t_i 和任务 t_j 之间传输的数据大小用 $e_{i,j} = (t_i, t_j)$ 表示; $D = \{d(t_1), d(t_2), \dots, d(t_n)\}$ 表示任务的计算工作量。由上述定义可以得出任务 t_i 的直接前驱任务 $P(t_i) = \{t_k | e_{k,i}\}$ 。每个工作流都会有一个设定的可靠性 Rel 值, 当调度策略的可靠性满足给定的可靠性约束时, 则认为该调度策略是可行的。

云平台通常以虚拟机的形式向用户提供计算资源, 云平台的虚拟机存在几个特性。1) 任务一旦被分配到虚拟机上执行, 则将完整地在此虚拟机上执行。2) 虚拟机一次只能执行一个任务, 无法同时执行多个任务。本文提出的云环境的资源由 m 个不同的虚拟机实例类型组成, 用 $R = \{r_1, r_2, \dots, r_m\}$ 表示。对于资源 r_i 可用 $r_i = \{t_{ist}^i, t_{ft}^i, u_i, c_i, \epsilon_i\}$ 表示, 其中 t_{ist}^i

表示资源 r_i 的开启时间, t_{if}^i 表示资源 r_i 的关闭时间, u_i 表示资源 r_i 的计算能力, c_i 表示资源 r_i 的单位时间价格, ϵ_i 表示资源 r_i 的故障率。不同实例的计算能力不同。

假设将任务 t_i 部署到资源 r_j 上, 则任务 t_i 的执行时间为:

$$ET(t_i, r_j) = \frac{d(t_i)}{u_j} \quad (1)$$

对于任务 t_i 的父任务 $P(t_i) = \{t_p | e_{p,i}\}$, 父任务 t_p 到任务 t_i 的传输时间 $trans(t_p, t_i)$ 为:

$$trans(t_p, t_i) = \begin{cases} \frac{e_{p,i}}{\beta}, & r(t_p) \neq r(t_i) \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

其中, β 表示 $r(t_p)$ 与 $r(t_i)$ 之间的带宽。考虑到任务间的数据依赖关系, 即子任务只有在父任务全部完成后才可以开始, 任务 t_i 的开始时间定义如下:

$$ST(t_i) = \max\{\max(FT(t_p) + trans(t_p, t_i)), Ava(r(t_i))\} \quad (3)$$

其中, $Ava(r(t_i))$ 表示虚拟机 $r(t_i)$ 准备执行任务 t_i 的最早时间, $FT(t_p)$ 表示任务 t_p 的完成时间, 则任务 t_i 的完成时间为:

$$FT(t_i) = ST(t_i) + ET(t_i, r(t_i)) \quad (4)$$

因此, 工作流的总体执行时间为:

$$T_{total} = \max\{FT(t_i) | t_i \in T\} \quad (5)$$

工作流执行代价包括计算代价和数据传输代价, 则执行代价为:

$$C_{total} = \sum_{j=1}^m c_{r_j} \times \left\lceil \frac{t_{if}^j - t_{st}^j}{\lambda_{r_j}} \right\rceil + \sum_{j=1}^n \sum_{k=j+1}^n c_{j,k} \times e_{i,j} \times s_{i,j} \quad (6)$$

其中, $c_{j,k}$ 表示资源 r_j 传输 1 GB 数据到 r_k 的所需要的价格, λ_{r_j} 表示资源 r_j 的要价单元时间。当任务 i 与任务 j 调度在不同的虚拟机实例上时, $s_{i,j} = 1$, 否则 $s_{i,j} = 0$ 。

对于用户来说, 工作流能否按照用户的需求按时完成是用户首要关注的问题。由于虚拟机崩溃、软件缺陷等因素的存在, 任务在虚拟机上可能无法在预定的时间内完成。本文的研究应考虑到故障导致的任务执行失败。通常, 瞬时故障遵循泊松分布, 因资源 r_i 的故障率为 ϵ_i , 故任务任务 t_i 在资源 r_j 上执行的可靠性为:

$$rel(t_i, r_j) = e^{-\epsilon_{r_j} \times ET(t_i, r_j)} \quad (7)$$

从式(7)中可以知道单个任务的可靠性由任务在虚拟机上的执行时间和虚拟机的故障率决定, 任务的执行时间越长, 机器故障率越高, 则可靠性越低。

由泊松分布的可加性可以得出, 工作流的总体可靠性为:

$$Rel = e^{\sum_{i=1}^n -\epsilon_{r(t_i)} \times ET(t_i, r(t_i))} = \prod_{i=1}^n rel(t_i, r(t_i)) \quad (8)$$

基于以上定义, 本文关注的是工作流总体可靠性约束下的完成时间和执行代价的优化调度问题, 可形式化表示为:

$$\min \begin{cases} (T_{total}, C_{total}) \\ \text{s. t. } Rel > \sigma_{rel} \end{cases} \quad (9)$$

其中, Rel 为当前调度方案的工作流总体可靠性数值, σ_{rel} 为预定义的可靠性约束阈值。

3.2 不确定性云环境中的工作流调度

在确定性云环境下的工作流调度问题中, 我们总是假定服务器的性能不会受到外界因素的影响, 任务在服务器上的执行时间和传输时间事先可以确定。然而, 现实环境中服务器

性能波动等因素可能会导致任务的执行时间不等于事先确定的时间。

受文献[19]的启发, 本文引入模糊理论, 使用三角模糊数来表示任务的计算时间和传输时间。三角模糊数 $\tilde{t} = (t^l, t^m, t^u)$ 的隶属函数 $u(x)$ 如图 2 所示。其中, t^m 为任务预定义的执行时间, 左右端点 t^l 和 t^u 表示任务执行时间的变化范围。

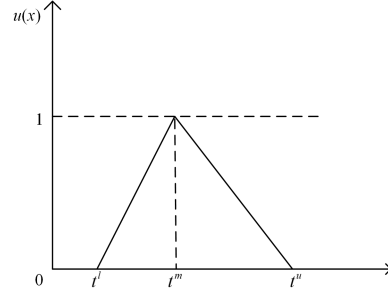


图 2 三角模糊数隶属函数图

Fig. 2 Diagram of triangular fuzzy number membership function

本文统一使用 $\tilde{\tau}$ 表示标量 τ 的三角模糊数。基于运行时间和传输时间不确定性的概念, 工作流的完成时间和执行成本均为三角模糊数, 表示为 $\widetilde{T_{total}}$ 和 $\widetilde{C_{total}}$, 则本文的优化问题可形式化表示为:

$$\min \begin{cases} (\widetilde{T_{total}}, \widetilde{C_{total}}) \\ \text{s. t. } Rel > \sigma_{rel} \end{cases} \quad (10)$$

对于优化目标 $\widetilde{T_{total}}$, 其值是一个三角模糊数, 由均值 $mean(\widetilde{T_{total}})$ 和方差 $std(\widetilde{T_{total}})$ 共同决定。Palacios 等^[20] 提出了一种比较准则来最小化两个目标值的线性组合, 称为 Palacios 准则。因此, 优化目标 $\widetilde{T_{total}}$ 的计算方式如式(11)所示:

$$\widetilde{T_{total}} = mean(\widetilde{T_{total}}) + \eta \times std(\widetilde{T_{total}}) \quad (11)$$

对于均值 $mean(\widetilde{T_{total}})$ 和方差 $std(\widetilde{T_{total}})$, Lee 等^[21] 定义了模糊集分别在均匀分布和比例分布下的均值和标准差, 本文的三角模糊数 $\widetilde{T_{total}}$ 是一种基于比例分布的情形, 故均值 $mean(\widetilde{T_{total}})$ 和方差 $std(\widetilde{T_{total}})$ 由下列公式计算得出:

$$mean(\widetilde{T_{total}}) = \frac{T_{total}^l + 2 \times T_{total}^m + T_{total}^u}{4} \quad (12)$$

$$std(\widetilde{T_{total}}) = \left[\frac{2 \times (T_{total}^l - T_{total}^m)^2 + (T_{total}^l - T_{total}^u)^2 + 4 \times (T_{total}^m - T_{total}^u)^2}{80} \right]^{\frac{1}{2}} \quad (13)$$

其中, η 为标准差 $std(\widetilde{T_{total}})$ 的权重。

对于三角模糊数 $\widetilde{C_{total}}$, 其处理方式同 $\widetilde{T_{total}}$ 。

3.3 工作流调度中的模糊理论

基于 Sun 等^[6] 所采用的方法, 本文提出了一种更加符合实际的模糊化方法来刻画任务的执行时间和传输时间。对于预估时间 t , 其对应的三角模糊数 $\tilde{t} = (t^l, t^m, t^u)$, t^m 为任务最可能的执行时间, 即给定的任务在服务器上的执行时间, t^l 和 t^u 的值分别从区间 $[\delta_1 \times t^m, t^m]$ 和 $[t^m, \delta_2 \times t^m]$ 内随机选取, 其中 $\delta_2 > 1 > \delta_1$ 。

在进行 workflow 调度时,需要对三角模糊数进行一些运算操作,因此,我们要对模糊数的一些运算进行重新定义。

1) 三角模糊数的加法运算。对于两个三角模糊数 $\tilde{t} = (t^l, t^m, t^u)$ 和 $\tilde{r} = (r^l, r^m, r^u)$, 根据模糊理论中定义的模糊数加法原则可以得出模糊数的加法规则如式(14)所示:

$$\tilde{t} + \tilde{r} = (t^l + r^l, t^m + r^m, t^u + r^u) \quad (14)$$

2) 三角模糊数的比较运算。受 Sakawa 等^[22]提出的比较准则对两个模糊数进行大小比较的启发,本文采用的比较原则如下:若 $\frac{t^l + 2 \times t^m + t^u}{4} > \frac{r^l + 2 \times r^m + r^u}{4}$, 则 $\tilde{t} > \tilde{r}$ 。

3) 三角模糊数的数乘运算。三角模糊数的数乘运算由式(15)给出。

$$\lambda \times \tilde{t} = (\lambda \times t^l, \lambda \times t^m, \lambda \times t^u) \quad \forall \lambda \in R \quad (15)$$

4 APSOGA 调度策略

本章将首先介绍基础 PSO, 然后具体介绍 APSOGA 调度策略。

4.1 PSO

PSO(粒子群优化)是一种协作的、基于种群的元启发式算法,由 Masdari 等^[23]于 1995 年提出。PSO 通过设计一个无质量的粒子模仿鸟群中的鸟来寻求问题的最佳解。粒子在 PSO 中极为重要,每个粒子代表问题的一个候选解,每个粒子都有两个属性,分别是速度 V 和位置 X , 粒子通过更新位置和速度在整个搜索空间内移动。粒子的速度受粒子自身情况、粒子自身最佳位置以及全局最佳位置 3 方面因素影响。为了评判每个粒子在问题空间的优越性,引入了适应度函数来评判粒子的好坏。每个粒子的好坏由自身的位置和速度决定,它们根据周围粒子和自身的经验在问题空间中不断迭代更新自身的位置和速度,粒子的速度依据式(16)更新,位置依据式(17)更新。

$$V_i^{t+1} = \omega \times V_i^t + c_1 \times r_1 \times (pBest_i^t - X_i^t) + c_2 \times r_2 \times (gBest^t - X_i^t) \quad (16)$$

$$X_i^{t+1} = X_i^t + V_i^t \quad (17)$$

其中, t 表示当前的迭代次数; V_i^t 和 X_i^t 表示第 i 个粒子在第 t 次迭代时的速度和位置; $pBest^t$ 和 $gBest^t$ 分别表示经过 t 次迭代后粒子自身最优位置和整个种群的最优位置; ω 表示惯性因子,它决定了算法的收敛能力; c_1 和 c_2 被称为加速常数,分别体现粒子对自身最优值和全局最优值的学习能力; r_1 和 r_2 是迭代过程中范围在 $(0, 1)$ 的随机数,用于加强算法的随机性。

4.2 APSOGA

本文将从以下 5 个方面具体阐述 APSOGA。

4.2.1 问题编码

由于云计算资源规模超大,因此资源池的构造会极大地影响算法的搜索效率。为了提高算法的搜索效率,需要设计一种合理的编码方式来使算法能够更好地解决 workflow 调度这一离散型的优化问题。本文使用云计算资源和任务组成的二维离散粒子的编码方式,一个粒子代表问题空间中的一个解,粒子 i 在 t 时刻的位置公式如式(18)所示:

$$X_i^t = \{t_{i1}^t, t_{i2}^t, \dots, t_{im}^t\} \quad (18)$$

其中, t_{i1}^t 表示第 i 个粒子的第 1 个任务在 t 时刻所处的虚拟机编号。图 3 给出了包含 5 个子任务的一个 workflow 调度策略对应的编码粒子。以任务 1 为例,任务 1 对应的虚拟机编号为 4,表示任务 1 将分配到资源池中编号为 4 的虚拟机上执行。

Task	1	2	3	4	5
Particle	4	3	1	0	2

图 3 编码粒子映射关系

Fig. 3 Mapping relationship of encoded particles

4.2.2 模糊适应度函数

本文的目标是在满足 workflow 的总体可靠性约束下优化 workflow 的总代价 f , 总代价 f 包含 workflow 的模糊执行代价 $\widetilde{C_{total}}$ 和模糊执行时间 $\widetilde{T_{total}}$ 。本文的调度目标有两个,属于多目标规划问题,故将适应度函数设为:

$$f = k_1 \times (\frac{\widetilde{T_{total}}}{T_{one}}) + k_2 \times (\frac{\widetilde{C_{total}}}{C_{one}}) \quad (19)$$

其中, k_1 和 k_2 分别表示完成时间和执行代价的权重系数, T_{one} 和 C_{one} 分别表示所有任务仅在一台服务器上执行所花费的完成时间和执行代价。由于在迭代过程中存在不满足可靠性约束的粒子,因此候选解中可能存在不满足可靠性约束的不可行解。故在算法中需要比较两个粒子的总代价时,必须要考虑粒子是否是可行解。本文将待比较的两个粒子的具体情况分成以下 3 种情况进行讨论。

1) 对于待比较的两个粒子,若两个粒子都满足可靠性约束,则选择总代价较小的粒子。

$$F(P_i^t) = f(P_i^t) \quad (20)$$

2) 若一个粒子满足可靠性约束,而另一个粒子不满足可靠性约束,则选择满足可靠性约束的粒子。

$$F(P_i^t) = \begin{cases} f(P_i^t), & rel > \sigma_{rel} \\ \infty, & rel < \sigma_{rel} \end{cases} \quad (21)$$

3) 若两个粒子都不满足可靠性约束,则本文选择可靠性大的粒子。因为该粒子经过迭代后更可能成为可行解,则:

$$F(P_i^t) = Rel(P_i^t) \quad (22)$$

4.2.3 粒子的更新策略

在算法搜索最优解时,粒子需要不断迭代更新自身的速度和位置。传统 PSO 更新自身位置的方式如式(16)所示,但传统 PSO 存在过早收敛的缺陷。为了避免算法过早收敛,APSOGA 引入了遗传算法的交叉算子和变异算子^[24],对式(16)的相应部分进行更新。粒子 i 在第 t 次迭代时,粒子的更新方式如式(23)所示,其中 \otimes 和 \odot 分别表示交叉算子和变异算子。

$$P_i^t = c_1 \otimes (c_2 \otimes (\omega \odot P_i^{t-1}, pBest_i^{t-1}), gBest^{t-1}) \quad (23)$$

APSOGA 在传统 PSO 更新公式的惯性部分引入了变异算子,其更新方式如下:

$$A_i^t = \omega \odot P_i^{t-1} = \begin{cases} Mu(A_i^{t-1}), & r_1 < p_m \\ A_i^{t-1}, & otherwise \end{cases} \quad (24)$$

其中, r_1 是 $(0, 1)$ 的随机数, p_m 为给定的变异概率,当 $r_1 < p_m$, A_i^{t-1} 发生变异操作, $Mu(A_i^{t-1})$ 会随机更改 A_i^{t-1} 的一个

分位的服务器编码,若 $r_1 \geq p_m$ 则不发生变异操作。图 4 为对图 3 编码粒子的变异操作。

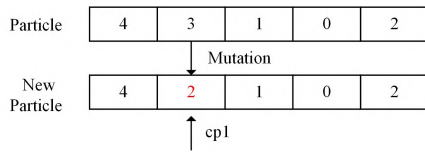


图 4 粒子的变异操作

Fig. 4 Mutation operation of particle

对于个人认知部分和社会认知部分,引入了交叉算子对式(16)的相应部分进行更新,其更新方式如下:

$$B_i' = c_1 \otimes (A_i', pBest_i'^{-1}) = \begin{cases} C_p(A_i', pBest_i'^{-1}), & r_2 < p_c \\ A_i', & \text{otherwise} \end{cases} \quad (25)$$

$$C_i' = c_2 \otimes (B_i', gBest'^{-1}) = \begin{cases} C_g(B_i', gBest'^{-1}), & r_2 < p_c \\ B_i', & \text{otherwise} \end{cases} \quad (26)$$

其中,式(25)和式(26)分别对个人认知部分和社会认知部分进行更新, r_2 和 r_3 是(0,1)内的随机数, p_c 为给定的交叉概率,当 r_2 (或 r_3) $< p_c$ 时,则 A_i' 发生变异操作, C_p (或 C_g)会随机选择粒子的两个分位,将粒子分位之间的服务器编码与 $pBest_i'^{-1}$ (或 $gBest'^{-1}$)对应的分位之间的服务器编码进行交叉。图 5 为对图 3 的个人认知部分的交叉操作,图 6 为对图 3 的社会认知部分的交叉操作。

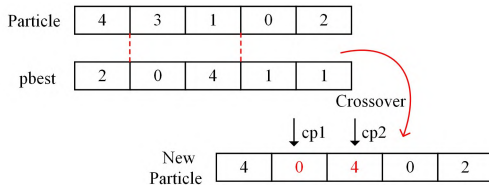


图 5 个人认知部分交叉操作

Fig. 5 Intersection of personal cognitive part

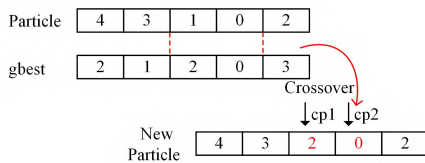


图 6 社会认知部分交叉操作

Fig. 6 Cross-operation of social cognition part

4.2.4 参数调整

式(16)的惯性权重因子 ω 能够决定 PSO 的收敛能力和搜索能力。对于惯性因子 ω ,当 ω 较小时,算法具有较强的局部搜索能力;当 ω 较大时,算法具有较强的全局搜索能力。式(27)是传统的 PSO 算法的惯性调整策略。

$$\omega = \omega_{\max} - \text{iter}_{\text{cur}} \times \frac{\omega_{\max} - \omega_{\min}}{\text{iter}_{\max}} \quad (27)$$

其中, ω_{\max} 和 ω_{\min} 分别为初始化时设定的 ω 的最大值和最小值, iter_{cur} 和 iter_{\max} 为当前迭代次数和算法的最大迭代次数。

传统 PSO 的惯性因子 ω 仅和迭代次数有关,它不能很好地满足实际问题的复杂性。本文提出一种新的惯性因子 ω 的调整策略,该策略能够根据当前粒子 X_i^{t-1} 与历史全局最佳粒子 $gBest^{t-1}$ 的差异自适应调整 ω 的值,从而增强 APSOGA

算法的搜索能力。 ω 的更新方式如下。

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times \exp\left(\frac{d(X_i^{t-1})}{d(X_i^{t-1}) - 1.01}\right) \quad (28)$$

$$d(X_i^{t-1}) = \frac{\text{div}(X_i^{t-1}, gBest^{t-1})}{|T|} \quad (29)$$

其中, $\text{div}(X_i^{t-1}, gBest^{t-1})$ 表示粒子 X_i^{t-1} 与历史全局最佳粒子 $gBest^{t-1}$ 之间的不同分位的数量, $|T|$ 表示工作流中子任务的数量。当 $\text{div}(X_i^{t-1}, gBest^{t-1})$ 的值较小时,表示当前粒子与历史全局最佳粒子的分位不同的数量较少,此时应该减小 ω 的值,增强算法的局部搜索能力,提高算法的收敛效果,找到最优解;反之,则应该增大 ω 的值,增强算法的全局搜索能力,扩大算法的搜索空间。

此外,算法的个人认知因子和社会认知因子的更新方式采用线性增减策略^[25]。其更新方式如式(30)和式(31)所示:

$$c_1 = c_1^{\text{start}} - \text{iter}_{\text{cur}} \times \frac{c_1^{\text{start}} - c_1^{\text{end}}}{\text{iter}_{\max}} \quad (30)$$

$$c_2 = c_2^{\text{start}} - \text{iter}_{\text{cur}} \times \frac{c_2^{\text{start}} - c_2^{\text{end}}}{\text{iter}_{\max}} \quad (31)$$

其中, c_1^{start} 和 c_2^{start} 分别为参数 c_1 和参数 c_2 设定的初始值, c_1^{end} 和 c_2^{end} 为 c_1 和 c_2 的最终值。

4.2.5 粒子到调度结果的映射

首先,对于编码粒子 i ,给出编码粒子 i 到调度结果的映射,具体如算法 1 所示。另外,基于 APSOGA 的工作流模糊调度策略的主要流程包含以下 6 个步骤。

1)初始化 APSOGA 的相关参数,如种群大小 PN 、最大迭代次数 Max_{iter} 、惯性因子 ω 等,并随机生成种群。

2)根据式(19)计算适应度,每个粒子的初始状态为个体最佳粒子,将初始种群中适应度值最小的粒子设为全局最佳粒子。

3)根据式(23),引入遗传算法的变异和交叉算子更新粒子自身的位置,计算更新后的粒子的适应度。

4)若更新后的粒子的适应度小于个体最佳粒子的适应度,则更新个体最佳粒子,将当前粒子设置为个体最佳粒子。

5)同时,将更新后的粒子的适应度与全局最佳粒子的适应度进行比较,若当前粒子的适应度小于全局最佳粒子的适应度,则更新全局最佳粒子,将当前粒子设置为全局最佳粒子并更新最佳适应度。

6)检查是否满足算法迭代结束条件,若满足,则算法结束,否则,返回步骤 3)。

算法 1 编码粒子到调度结果的映射

输入:(W,R,X)

输出:F

1. 初始化: $T_{\text{total}} \leftarrow 0, C_{\text{total}} \leftarrow 0$
2. 计算 $ET[|R| \times |T|]$ 和 $E[|T| \times |T|]$
3. 计算三角模糊数 V_{ET}
4. 根据式(8)计算 Rel
5. if $Rel < \sigma_{\text{rel}}$
6. 粒子不满足可靠性约束,将粒子设为不可行粒子
7. for t_i in W
8. if t_i 为入任务 then
9. $ST(t_i) = (0, 0, 0)$
10. $t_{\text{lat}}^{X(t_i)} = (0, 0, 0)$
11. $t_{\text{let}}^{X(t_i)} += V_{ET}(t_i, X(t_i))$

```

12. else
13.   maxT=(0,0,0)
14.   for  $t_p$  in  $P(t_i)$ 
15.     maxT=max(maxT,FT( $t_p$ )+ trans( $t_p, t_i$ ))
16.   end for
17.   ST( $t_i$ )=max(maxT,  $t_{let}^{X(t_i)}$ )
18.    $t_{let}^{X(t_i)} = V_{ET}(t_i, X(t_i))$ 
19. end if
20. FT( $t_i$ )=ST( $t_i$ )+  $V_{ET}(t_i, X(t_i))$ 
21. end for
22. 基于式(19)计算适应度值 F
23. 返回 F

```

5 实验仿真与结果

本文的模拟实验环境选用 64 位 Windows 10 系统,并配置 16GB 内存和 2.30GHz 的 i7 处理器。基于文献[23],AP-SOGA 算法的参数设置如下:种群大小设置为 50,最大迭代次数为 500, $c_1^{start}=0.9$, $c_1^{end}=0.4$, $c_2^{start}=0.4$, $c_2^{end}=0.9$ 。

5.1 实验设置

本文使用的工作流测试模型采用 Bharathi 等^[26]研究的 5 个不同领域的工作流:地震科学 CyberShake、生物信息学 Sipht、天文学 Montage、重力物理学 LIGO 以及生物基因组学 Epigenomics。这 5 种工作流各有不同的结构,关于工作流的详细信息可以参考文献[27]。图 7 给出了这 5 种工作流样例的结构。这些工作流的详细信息都存储在 xml 格式的文件中。

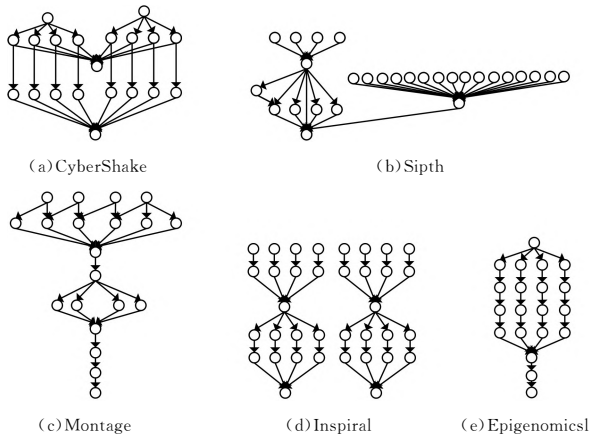


图 7 5 种科学工作流结构图

Fig. 7 5 Scientific workflow structure diagrams

针对不同的工作流,本文各选取了 3 种不同规格的工作流:约 30 个任务的微型工作流、约 50 个任务的小型工作流,以及约 100 个任务的中型工作流。本文云资源池有 6 个云服务器,如表 1 所列,假定服务器 m4.16xlarge 的计算能力最强,工作流的各个任务在 m4.16xlarge 的计算时间直接由相应的 xml 文件得到,根据其他服务器与 m4.16xlarge 服务器的性能比较,可以得出各任务在其他服务器上的执行时间。

本文从 Amazon EC2 云平台选择了 6 台虚拟机进行仿真实验,虚拟机资源配置信息如表 1 所列。Amazon EC2 通常以 60s 或 1h 为要价区间 λ_i ,本文选取 60s 作为要价区间,传输 1GB 数据的价格为 0.2。

表 1 虚拟机配置信息

Table 1 Virtual machine configuration information

Name of VM	Computing ability	Bandwidth	Price	Fault rate
m4. large	1	20	0.16	0.100
m4. xlarge	2	20	0.26	0.070
m4. 2xlarge	4	20	0.53	0.030
m4. 4xlarge	6	20	0.93	0.020
m4. 10xlarge	8	20	2.13	0.008
m4. 16xlarge	10	20	3.33	0.006

工作流的可靠性 σ_{rel} 表示工作流按照给定时间在服务器 m4.xlarge 上运行的可靠性, T_{one} 和 C_{one} 表示工作流在 m4.2xlarge 一台服务器上运行所需的完成时间和执行代价。参考文献[19],将任务的执行时间和传输时间转化为三角模糊数,参数 δ_1 和参数 δ_2 分别设置为 0.85 和 1.2,标准差的权重系数 η 取 1,适应度函数权重系数 k_1 和 k_2 分别为 0.2 和 0.3。

5.2 对比算法

在本小节中,为了评估 APSOGA 算法的有效性,基于文献[17,19,24],本文将 APSOGA 策略与传统的 PSO 策略、随机策略进行了实验效果的对比。在现有的工作流调度问题中,这些算法通常被用作对比算法。

在传统 PSO 策略中,采用的是与 APSOGA 相同的编码方式,更新方式采用传统的更新方式,传统 PSO 算法参数设置参考文献[28],种群大小设置为 50,最大迭代次数为 500, $c_1=1$, $c_2=1$, $w=1$ 。

随机策略:该方法基于随机搜索策略,采用与 APSOGA 相同的编码方式,使用随机的方式更新粒子编码,每次迭代互不影响,在问题的解空间随机搜索,并计算每个粒子的适应度,记录搜索过程中的最优解。

5.3 实验结果

为了测试 APSOGA 策略、PSO 策略以及随机策略在云环境中受到网络波动、服务器故障等因素影响下的工作流调度性能,本文使用 3 种策略对不同规模的 5 种工作流分别进行 30 组重复实验,表 2—表 4 分别记录了 3 种策略的工作流调度实验的最优适应度值和平均适应度值,适应度值如 4.2.2 节所述,代表了完成时间和执行代价的加权和。

表 2 微型工作流实验效果对比

Table 2 Comparison of micro-workflow experiment effects

Workflow	Algorithm	Optimal fitness	Mean fitness
CyberShake	APSOGA	0.316274673	0.342816100
	PSO	0.348376492	0.373336762
	RA	0.430570045	0.478607649
Sipht	APSOGA	0.274703948	0.280768193
	PSO	0.279794983	0.282450197
	RA	0.288961944	0.294797072
Montage	APSOGA	0.310632504	0.336793669
	PSO	0.327377613	0.349184289
	RA	0.360824555	0.490227673
Inspiral	APSOGA	0.236083304	0.246072118
	PSO	0.247453520	0.262809501
	RA	0.285602118	0.311960133
Epigenomics	APSOGA	0.255134902	0.261166488
	PSO	0.258131627	0.262362935
	RA	0.267709750	0.274282930

表 3 小型工作流实验效果对比

Table 3 Comparison of small-workflow experiment effects

Workflow	Algorithm	Optimal fitness	Mean fitness
CyberShake	APSOGA	0.280477047	0.311609159
	PSO	0.327026552	0.345424291
	RA	0.456115870	0.487787609
Sipth	APSOGA	0.256871661	0.263624583
	PSO	0.262714713	0.269183426
	RA	0.279992812	0.290416137
Montage	APSOGA	0.277671636	0.326138729
	PSO	0.308367552	0.325139067
	RA	0.478108312	0.556199821
Inspirial	APSOGA	0.230110491	0.243822453
	PSO	0.253372721	0.267809484
	RA	0.306582671	0.361179349
Epigenomics	APSOGA	0.230613807	0.241196914
	PSO	0.232376859	0.243518946
	RA	0.249907848	0.264843214

表 4 中型工作流实验效果对比

Table 4 Comparison of medium-workflow experiment effects

Workflow	Algorithm	Optimal fitness	Mean fitness
CyberShake	APSOGA	0.284462751	0.330523659
	PSO	0.337750275	0.356301833
	RA	0.533287540	0.571779220
Sipth	APSOGA	0.238000629	0.250372346
	PSO	0.242780921	0.263163606
	RA	0.282576112	0.313619981
Montage	APSOGA	0.262792479	0.312809500
	PSO	0.295509745	0.313218128
	RA	0.564526004	0.601694119
Inspirial	APSOGA	0.232744516	0.255352093
	PSO	0.260283850	0.273464895
	RA	0.409879356	0.452943266
Epigenomics	APSOGA	0.228919984	0.243915924
	PSO	0.232344934	0.247351706
	RA	0.292991561	0.338385882

APSOGA 策略、PSO 策略以及随机策略在 5 种微型工作流进行 30 次重复实验的调度结果如表 2 所列。从表中可以得出,对于微型工作流,无论是平均适应度值还是最优适应度值,APSOGA 策略都可以获得最优解,PSO 策略次之,随机策略最差,这是由于 APSOGA 策略在 PSO 策略的基础之上加入了遗传算法的变异操作和交叉操作,使得 APSOGA 算法比 PSO 算法拥有更好的全局寻优能力。而随机策略的效果较差是因为随机策略在搜索问题空间中的解时效率较低。

小型工作流 30 次重复的调度结果如表 3 所列,可以看出,APSOGA 策略在最优适应度值和平均适应度值上均得到了最优解。此外,APSOGA 策略的最优适应度值优于传统 PSO 策略最高达 4.7%,最高优于随机策略 20%。

中型工作流 30 次重复的调度结果如表 4 所列,APSOGA 策略在不同程度上都优于其他算法,并且观察随机策略在中型工作流和微型工作流的表现可以发现,随着任务数量的增多,问题空间的变大,随机策略性能越来越差,与 APSOGA 策略的实验效果的差距随着任务数量的增多而变大。综上所述,APSOGA 策略无论在规模小的工作流还是规模大的工作流,与其他调度策略相比,都具有更好的调度性能。

APSOGA 策略在传统 PSO 策略的基础上引入了遗传算法的变异算子和交叉算子,避免了传统 PSO 容易过早收敛的缺陷,也避免了粒子陷入局部最优,让粒子可以有更好的全局

寻优能力。其次,APSOGA 策略调整了对惯性因子 w 的更新方式,传统 PSO 策略采用线性方式更新惯性因子 w , w 仅和当前迭代次数有关,不能很好地解决复杂问题,而 APSOGA 策略采用自适应调整的方式,考虑了当前粒子与全局最佳粒子的分位不同的数量,提高了 w 对算法的搜索性能。实验结果表明,无论是相比传统 PSO 策略还是随机策略,APSOGA 策略都可以获得更好的调度方案。

结束语 本文针对虚拟机运行过程中可能会发生的性能波动和宕机等问题,设计了一种在工作流总体可靠性约束下使用三角模糊数表示任务的执行时间和传输时间的问题模型,同时提出了一种 APSOGA 算法以优化工作流的完成时间和执行代价。该算法基于传统 PSO 算法,并加入了遗传算法的变异操作和交叉操作避免算法陷入局部最优。通过对 3 种调度策略在 5 种不同规格下的科学工作流的调度测试表明,APSOGA 策略在云环境服务器可能发生性能波动和宕机等问题的情况下,对带可靠性约束的工作流调度具有更佳的适应性。

在未来的工作中,针对工作流调度模型框架的 3 个组成部分,计算资源将考虑加入边缘计算资源,带可靠性约束的单工作流将扩展为多工作流,调度器则将具体分成几个部分,每个部分实现不同的功能,以此进一步完善问题模型。

参 考 文 献

- [1] LI H,WANG D,ZHOU M C,et al. Multi-Swarm Co-Evolution Based Hybrid Intelligent Optimization for Bi-Objective Multi-Workflow Scheduling in the Cloud[J]. IEEE Transactions on Parallel and Distributed Systems,2021,33(9):2183-2197.
- [2] RIZVI N,RAMESH D. HBDCWS: heuristic-based budget and deadline constrained workflow scheduling approach for heterogeneous clouds [J]. Soft Computing, 2020, 24 (24): 18971-18990.
- [3] LIU J,REN J,DAI W,et al. Online multi-workflow scheduling under uncertain task execution time in IaaS clouds[J]. IEEE Transactions on Cloud Computing,2019,9(3):1180-1194.
- [4] ZHANG X. Multi-objective optimization of workflow scheduling in uncertain cloud environment[J]. Computer Engineering and Design,2021,42(7):1948-1956.
- [5] GAO D,WANG G G,PEDRYCZ W. Solving fuzzy job-shop scheduling problem using DE algorithm improved by a selection mechanism[J]. IEEE Transactions on Fuzzy Systems, 2020, 28(12):3265-3275.
- [6] SUN L,LIN L,GEN M,et al. A hybrid cooperative coevolution algorithm for fuzzy flexible job shop scheduling [J]. IEEE Transactions on Fuzzy Systems,2019,27(5):1008-1022.
- [7] YE L,XIA Y,YANG L,et al. Dynamic Scheduling Stochastic Multiworkflows With Deadline Constraints in Clouds[J]. IEEE Transactions on Automation Science and Engineering,2022. doi: 10.1109/TASE.2022.3204313.
- [8] GHAMRANI M H,ZHOU M C,HON C T. Toward cloud computing QoS architecture: Analysis of cloud systems and cloud services[J]. IEEE/CAA Journal of Automatica Sinica, 2017,4(1):6-18.

- [9] WANG Y, ZUO X. An effective cloud workflow scheduling approach combining PSO and idle time slot-aware rules[J]. IEEE/CAA Journal of Automatica Sinica, 2021, 8(5): 1079-1094.
- [10] LI H, WANG D, ZHOU M C, et al. Multi-Swarm Co-Evolution Based Hybrid Intelligent Optimization for Bi-Objective Multi-Workflow Scheduling in the Cloud[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 33(9): 2183-2197.
- [11] FARAGARDI H R, SEDGHPUR M R S, FAZLIAHMADI S, et al. GRP-HEFT: A budget-constrained resource provisioning scheme for workflow scheduling in IaaS clouds[J]. IEEE Transactions on Parallel and Distributed Systems, 2019, 31(6): 1239-1254.
- [12] WANG P, LEI Y, AGBEDANU P R, et al. Makespan-driven workflow scheduling in clouds using immune-based PSO algorithm[J]. IEEE Access, 2020, 8: 29281-29290.
- [13] TAGHINEZHAD-NIAR A, PASHAZADEH S, TAHERI J. QoS-aware online scheduling of multiple workflows under task execution time uncertainty in clouds[J]. Cluster Computing, 2022, 25(6): 3767-3784.
- [14] PHAM T P, FAHRINGER T. Evolutionary multi-objective workflow scheduling for volatile resources in the cloud[J]. IEEE Transactions on Cloud Computing, 2020, 10(3): 1780-1791.
- [15] CAO H, XU X, LIU Q, et al. Uncertainty-aware resource provisioning for workflow scheduling in edge computing environment [C]// 2019 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/13th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE). IEEE, 2019: 734-739.
- [16] LI K, ZHANG G, ZHU Z. A decomposition-based multi-objective workflow scheduling algorithm in cloud environment[J]. Computer Engineering & Science, 2016, 38(8): 1588-1594.
- [17] LIN B, GUO W, CHEN G. Scheduling strategy for science workflow with deadline constraint on multi-cloud[J]. Journal on Communication, 2018, 39(1): 56-69.
- [18] MENG S, HUANG W, YIN X, et al. Security-aware dynamic scheduling for real-time optimization in cloud-based industrial applications[J]. IEEE Transactions on Industrial Informatics, 2020, 17(6): 4219-4228.
- [19] LIN C, LIN B, CHEN X. Study on Scientific Workflow Scheduling Based on Fuzzy Theory Under Edge Environment[J]. Computer Science, 2022, 49(2): 312-320.
- [20] PALACIOS J J, GONZALEZ-RODRIGUEZ I, VELA C R, et al. Coevolutionary makespan optimisation through different ranking methods for the fuzzy flexible job shop[J]. Fuzzy Sets and Systems, 2015, 278: 81-97.
- [21] LEE E S, LI R J. Comparison of fuzzy numbers based on the probability measure of fuzzy events[J]. Computers & Mathematics with Applications, 1988, 15(10): 887-896.
- [22] SAKAWA M, KUBOTA R. Fuzzy programming for multiobjective job shop scheduling with fuzzy processing time and fuzzy due date through genetic algorithms[J]. European Journal of Operational Research, 2000, 120(2): 393-407.
- [23] MASDARI M, SALEHI F, JALALI M, et al. A survey of PSO-based scheduling algorithms in cloud computing[J]. Journal of Network and Systems Management, 2017, 25(1): 122-158.
- [24] GUO W, LIN B, CHEN G, et al. Cost-driven scheduling for deadline-based workflow across multiple clouds [J]. IEEE Transactions on Network and Service Management, 2018, 15(4): 1571-1585.
- [25] SHI Y, EBERHART R. A modified particle swarm optimizer [C]// 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360). IEEE, 1998: 69-73.
- [26] BHARATHI S, CHERVENAK A, DEELMAN E, et al. Characterization of scientific workflows[C]// 2008 third Workshop on Workflows in support of Large-scale Science. IEEE, 2008: 1-10.
- [27] JUVE G, CHERVENAK A, DEELMAN E, et al. Characterizing and profiling scientific workflows[J]. Future Generation Computer Systems, 2013, 29(3): 682-692.
- [28] RODRIGUEZ M A, BUYYA R. Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds[J]. IEEE Transactions on Cloud Computing, 2014, 2(2): 222-235.



LI Jinliang, born in 1998, postgraduate, is a member of China Computer Federation. His main research interests include cloud computing and heuristic algorithms.



LIN Bing, born in 1986, Ph.D, associate professor, postgraduate supervisor, is a member of China Computer Federation. His main research interests include cloud computing and intelligent computing and its application.

(责任编辑:何杨)