

# RAQPPO 算法的云计算资源调度策略

赵 昱, 惠晓滨, 许建虹, 钟季龙

(空军工程大学装备管理与安全工程学院,西安,710051)

**摘要** 在“互联网+”时代,云计算代表了一种新的商业模式,而云系统中用户任务与计算节点的调度问题极大地影响着系统的性能和云竞争力。为此,提出了一种改进的量子粒子群算法——反向自适应量子粒子群算法(RAQPPO),通过对惯性权值参数的调整和加入反向学习算子来提高算法的全局搜索能力,并将其应用于云计算资源调度中,仿真验证了算法的有效性。建立了云计算资源调度问题的模型;采用自适应机制,将适应度函数的变化程度作为惯性权值的更新因子,避免了单纯地根据迭代次数的线性函数来取值,从而使粒子不易陷入局部最优;随后加入粒子反向学习算子,加强了粒子全局搜索能力。实验结果表明,RAQPPO 算法大大节约了任务完成时间,并且保持了良好的计算节点负载平衡。

**关键词** 云计算;资源调度;量子粒子群算法;惯性权值;反向学习

**DOI** 10.3969/j.issn.1009-3516.2016.06.013

**中图分类号** TP393 **文献标志码** A **文章编号** 1009-3516(2016)05-0070-06

## Cloud Computing Resource Scheduling Strategy Based on Improved RAQPPO Algorithm Research

ZHAO Yu, HUI Xiaobin, XU Jianhong, ZHONG Jilong

(Air Force Engineering University Material Management and Safety Engineering College, Xi'an 710051, China)

**Abstract:** In 'INTERNET + TIME', cloud computing represents a novel business model. However, the cloud user tasks in the system and compute node scheduling problem significantly affects the system performance and competitiveness of cloud. An improved algorithm of quantum particles-adaptive quantum particle swarm optimization (RAQPPO), based on the inertia weight adjustment of parameters and reverse learning to improve the global search ability of the algorithm, and applied to cloud computing resource scheduling problem to verify the effectiveness of the algorithm. With cloud computing resource scheduling model is established. And then uses the adaptive mechanism, the change of the fitness function as update of inertia weight factor, avoids simply value according to the linear function of the number of iterations. Add the particle reverse learning operator, to strengthen the global search ability particles. The experimental results show that the RAQPPO algorithm greatly save the task completion time, and keep a good computing nodes load balancing.

**Key words:** Cloud computing; Resource scheduling; Quantum particle swarm optimization; The inertia

**收稿日期:** 2016-03-24

**基金项目:** 国家自然科学基金(71501184)

**作者简介:** 赵昱(1993-),男,山西运城人,硕士生,主要从事云计算和智能算法研究.E-mail: 1803192468@qq.com

**引用格式:** 赵昱,惠晓滨,许建虹,等. RAQPPO 算法的云计算资源调度策略[J]. 空军工程大学学报:自然科学版,2016,17(6):70-75. ZHAO Yu, HUI Xiaobin, XU Jianhong, et al. Cloud Computing Resource Scheduling Strategy Based on Improved RAQPPO Algorithm Research [J]. Journal of Air Force Engineering University: Natural Science Edition, 2016, 17(6): 70-75.

weight; Reverse learning

当微处理器、微芯片等“微终端”计算机技术出现时,没有人能想到这些部署在卡条的晶体管为人类带来了“爆炸”式的信息增长。每一部手机、平板电脑甚至是一块小小的智能电子表,都为这个信息时代提供了前所未有的海量数据<sup>[1]</sup>。据IBM公司的统计研究,整个人类文明所获得的全部数据中,有90%是过去2年内产生的。到2020年,全世界产生的数据规模将达到现在的44倍,每一天全世界上传5亿张图片,每分钟有20h时长的视频被分享。人类已经进入“大数据时代”,传统的数据处理方式已不能满足人们对计算服务质量的要求,在此背景下云计算应运而生。

云计算集成了分布式计算和网格计算等多种技术,可以将分布于不同物理域的服务器通过逻辑连接成一个巨大的资源池,通过虚拟技术等将实现用户任务的快速完成<sup>[2]</sup>。在海量数据面前,如何使这些资源高效、稳定地运转,是云计算研究领域的一项重要课题<sup>[3]</sup>。

云计算资源调度实际上是一种多约束、多目标优化问题,是一类NP难问题。而众多智能算法的出现为解决此类问题提供了思路。如文献[3]提出采用遗传算法求解云计算资源调度问题,与传统算法相比,获得了较好的服务质量;文献[4]将蚁群寻找食物的路径选择作为寻找最优调度方案的途径,兼顾了任务完成时间和任务吞吐量两方面目标,取得了不错的调度效果。还有诸如蛙跳、蜂群、海豚群等多种仿生算法来完成最优方案的搜索。在这些算法中,粒子群算法通过较少的参数设置,就能够提高全局的搜索能力,因此成为了云计算资源调度研究中的一个重要方向。然而易陷入局部最优和出现“早熟”的现象是其在计算诸多优化问题中难以克服的缺点。量子粒子群算法是将量子力学概念与粒子群算法相结合,通过解薛定谔方程求解和蒙特卡罗估计等方法,以波函数的形式描述粒子在空间内的运动状态,通过波函数塌缩来确定粒子的位置。实验表明,与标准粒子群算法相比,量子粒子群算法具有更好的计算性能,在寻找最优解和避免陷入局部最优等方面更加适合求解优化问题。

本文提出了一种改进的量子粒子算法——反向自适应量子粒子群算法(RAQPSO)。通过采用自适应机制,将适应度函数的变化程度作为惯性权值的更新因子,避免了单纯地根据迭代次数的线性函数来取值,加入反向学习算子,扩大了粒子的搜索范围。实验和仿真结果表明,RAQPSO算法能够有

效减少任务完成时间,同时能够保持各计算节点的负载均衡。

## 1 云计算资源调度模型

云计算系统在保证满足用户高质量服务的同时,尽可能地减少服务时间,使用户可以在较短的时间内获得计算结果或数据。从服务提供者角度来看,云计算的系统资源是有限的,资源竞争是不可避免的,如何减少资源竞争,同时将负载均衡的分配给各个计算节点,是存在于整个服务过程中的重要问题。通过资源调度策略,使各个计算节点的任务均衡合理的分配,是解决此类组合优化问题的有效途径。在资源调度的过程中,模型可由以下几个要素组成:

1)任务集合  $S = \{s_1, s_2, \dots, s_n\}$ ,表示  $n$  个相互独立且不可再分的任务组成的集合。

2)虚拟资源集合  $V = \{v_1, v_2, \dots, v_m\}$ ,表示云计算资源池中所包含的虚拟计算资源,其中每个计算资源  $v_i$  采用如下形式描述:

$$v_i = \{\text{CPU}, \text{MEMORY}, \text{DISK}\}$$

其中,CPU表示内核数,MEMORY表示内存大小,DISK表示磁盘空间大小。

3)执行时间矩阵  $T = (t_{ij})$ , ( $i = 1, 2, \dots, n; j = 1, 2, \dots, m$ ),  $t_{ij}$  表示第  $i$  个任务在第  $j$  个计算资源上完成所需要的时间。

4)决策矩阵  $D = (d_{ij})$ ,  $d_{ij} \in \{0, 1\}$ ,其中,  $d_{ij} = 1$  表示第  $i$  个任务由第  $j$  个计算资源执行;  $d_{ij} = 0$  表示第  $i$  个任务不在第  $j$  个计算资源执行。

基于以上要素,如何能够确定决策矩阵,即找到一种合理的调度方案,使用户任务能够快速完成,是资源调度的一个主要目标。为简化模型,本文对模型进行如下约定:

- 1)虚拟机的性能满足任一任务的要求;
- 2)每个任务只能由一个虚拟计算资源执行;
- 3)不考虑任务传输时间对模型优化的影响。

由于云计算系统的计算并行性,多个任务可以同时数据中心的各个节点上执行,对于一个调度方案  $D$ ,完成  $n$  个任务的时间为执行时间最长的虚拟机,其数学表达式为:

$$T(D) = \max_{j=1}^m \sum_{i=1}^n d_{ij} \text{time}_{ij}$$

式中: $n$  为任务数量; $m$  为云计算资源的数量。

## 2 量子粒子群算法(QPSO)

量子粒子群算法具有参数少、易于理解等诸多优点,但在实际应用中却出现了容易早熟、全局寻优能力差等特点。在2004年,Sun等从量子力学的角度提出了一种新的粒子演化模型。该模型结合了标准粒子群的优点,同时粒子以概率波的形式出现在空间,省去了粒子的运动速度,使算法迭代的更加精炼有序。在量子空间中,该算法可以在整个可行区域内搜索最佳解,大大提高了全局搜索能力。

首先,假定系统是一个量子粒子系统,假设粒子的运动空间为 $N$ 维,种群数量为 $m$ ,迭代次数为 $t$ 。在搜索空间中,第 $i$ 个粒子的位置是 $x_i = (x_{i1}, x_{i1}, \dots, x_{iN})$ ,其中 $i = 1, 2, \dots, m$ 。将 $x_i$ 代入目标函数即可算出其适应值。记第 $i$ 个粒子在第 $t$ 次迭代中搜索到的局部最优位置为 $p_i(t) = (p_{i1}, p_{i1}, \dots, p_{iN})$ ,整个粒子群搜索到的最优位置为 $p_g(t) = (p_{g1}, p_{g1}, \dots, p_{gN})$ ,其中 $g$ 代表处于全局最好位置粒子的下标。

在量子空间中,利用波函数 $\psi(x, t)$ 描述粒子的状态,通过求解薛定谔方程得到粒子的概率密度函数和概率分布函数,而粒子在空间中的具体位置则由蒙特卡罗反变换得到,有:

$$x(t) = \alpha p_i(t) + \beta p_g(t) \pm \frac{L}{2} \ln\left(\frac{1}{u}\right) \quad (1)$$

式中: $\alpha + \beta = 1$ ;  $u$ 为 $[0, 1]$ 上服从均匀分布的随机数; $L$ 由下式确定:

$$L(t+1) = 2\xi |x_a(t) - x(t)| \quad (2)$$

最后,得到量子粒子群算法的进化方程为:

$$x(t+1) = \alpha p_i(t) + \beta p_g(t) \pm \xi |x_a(t) - x(t)| \ln\left(\frac{1}{u}\right) \quad (3)$$

式(3)中引入了当前粒子平均最好位置 $x_a(t)$ ,表示粒子群中所有粒子在第 $t$ 次迭代时个体的平均最佳位置,其表达式为:

$$x_a(t) = \frac{1}{N} \sum_{i=1}^N x_i(t) \quad (4)$$

$\xi$ 为惯性权值,也是QPSO算法中唯一的一个收敛参数,其取值可以固定不变,也可引入自适应机制动态调节,一般按照下式取值:

$$\xi = 0.5 + \frac{0.5(t_{\max} - t)}{t_{\max}} \quad (5)$$

式中: $t_{\max}$ 为最大迭代次数; $t$ 表示当前迭代次数。

因此标准的量子粒子群算法流程如下:

**步骤1** 初始化群体规模、最大迭代次数、解空间维数以及粒子位置;

**步骤2** 分配初始化粒子局部最优以及全局最优值;

**步骤3** 优化过程,按照式(3)、(4)和(5)更新QPSO算法中的所有粒子;

**步骤4** 计算群体当前的全局最优位置,即更新 $p_i$ 和 $p_g$ ;

**步骤5** 根据约束条件进行判断,当前所求的解是否满足,若满足输出最优解,否则跳到步骤2继续搜索直到满足条件或达到最大迭代次数为止。

## 3 基于RAQPSO算法的云计算资源调度策略

考虑云计算调度策略的任务完成时间,本文定义所有任务完成时间为适应度函数,其数学表达式如下:

$$T(X) = \max_{j=1}^m \sum_{i=1}^n x_{ij} \text{time}_{ij} \quad (6)$$

针对云计算任务调度离散性的特点,本文采取任务→计算节点的向量式编码方式。将 $m$ 个原子任务分配给 $n$ 个计算节点,粒子的编码方式为 $\mathbf{X} = (x_1, x_2, \dots, x_m)$ , $x_i$ 为整数,且 $x_i \in [0, n-1]$ ,表示计算节点标号。如 $\mathbf{X} = (2, 0, 3, 1, 4)$ 表示的含义为将5个任务分配给标号为0、1、2、3、4的5个计算节点,其中,第1个任务分配给2号,第2个任务分配给0号...,依次类推。

通过上节粒子更新计算公式可知,惯性权值 $\xi$ 的选择关系到整个算法的收敛性和搜索能力,因此如何选择 $\xi$ 成为QPSO算法的关键。若一个确定的粒子,其当前位置的适应度函数值与上一次迭代位置的适应度函数值相差不大,说明粒子搜索的范围较小,此时应增大搜索范围,增加惯性权值使粒子“跳出”上一代局部最优和全局最优区间,达到发散状态;同理,当两者相差较大时,减小惯性权值,使粒子收敛。因此当粒子陷入局部最优解时,粒子的运动范围适当增大,即实现粒子运动的小变异操作。

基于以上考量,本文提出了一种自适应机制的 $\xi$ 选取方法,即:

$$\xi(t+1) = 1 - 0.5f(t) \quad (7)$$

$f(t)$ 由下式确定:

$$f(t) = \frac{F(x_g(t))}{F(x_g(t-1))} \quad (8)$$

式中: $F(x_g(t))$ 为当前粒子全局最优点的适应度函数值。可以看出更新因子 $f(t)$ 的值在 $[0, 1]$ 之间,当其趋于0时表示远离最优点, $\xi$ 值减小;当其值趋于1时表示接近最优点, $\xi$ 值增大。一般取初始

状态下的  $\xi$  值为 1。

为了进一步扩大粒子搜索范围,考虑向粒子加入反向学习算子。基于本文粒子标号方法,定义反向学习算子如下:  $y_i = n - x_i$ 。其中  $n$  为计算节点总数,则粒子通过反向学习获得的运动位置为:  $Y = (y_1, y_2, \dots, y_m)$ 。当粒子陷入局部最优解时,通过一种反向学习机制,实现粒子位置的“突变”型操作。

因此,基于 RAQPSO 算法的云计算资源调度策略流程为:

**步骤 1** 初始化群体规模、最大迭代次数、解空间维数以及粒子位置;

**步骤 2** 计算各个粒子的适应度函数值,并且确定初始位置各个粒子的局部最优值以及它们全局最优值;

**步骤 3** 优化过程,按照式(6)确定迭代惯性权值的大小,并且,通过式(3)、(4)和(5)更新算法中的粒子;

**步骤 4** 计算粒子反向学习位置,通过比较反向学习前后粒子适应度值进一步更新粒子位置;

**步骤 5** 计算群体当前的全局最优位置和各个粒子局部最优位置的适应度函数值,与前一次迭代的适应度函数比较,根据比较结果更新  $p_g$  和  $p_i$  值;

**步骤 6** 根据约束条件进行判断,当前所求的解是否满足,若满足输出最优解,否则跳到步骤 2 继续搜索直到满足条件或达到最大迭代次数为止。

综上所述,RAQPSO 算法的云计算资源调度流程见图 1。

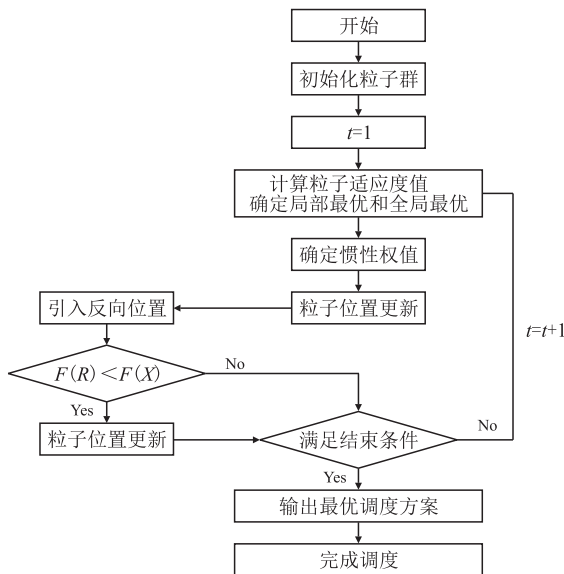


图 1 RAQPSO 算法的云计算资源调度模型求解流程  
Fig.1 The algorithm solving process of cloud computing resource scheduling model

### 4 实验与仿真

为了验证 RAQPSO 算法在云计算资源调度中应用的有效性,本文采用 Cloudsim 作为实验模拟平台,在处理器为 Inter(R)Core(TM) i7-4710MQ,内存为 8 G,操作系统为 Windows 8 的计算机上进行仿真实验。

Cloudsim 其核心类包括 DataCenter、DataCenterBroker、Host、VirtualMachine 等云计算平台及模拟资源。本文通过修扩展 DataCenterBroker 类,重载了 bindCloudletToVm() 方法,将算法加入到云计算资源调度中。在调用主程序 CloudSim.startSimulation() 方法之前,完成云任务向虚拟机的绑定。

实验中,将 PSO 算法、QPSO 算法和 RAQPSO 算法在云计算资源调度方案上的性能进行对比。在相同实验条件下,3 种算法的参数设置为:种群规模为 10,PSO 算法中  $c_1$  和  $c_2$  都为 2,所有算法迭代次数为 500 次。

当云计算资源数量为 20 个时,将 10~200 个不同的任务分配到资源节点进行计算,其中,虚拟机参数为:ram=512 MB,imagesSize=10 000 MB,不同虚拟机 MIPS 为 250~2 000 之间的随机整数。云任务由 Length 数值不同的输入参数来设定。采用 PSO、QPSO 和 RAQPSO 算法对资源调度方案进行求解,3 种算法所得调度方案的任务完成时间见图 2。

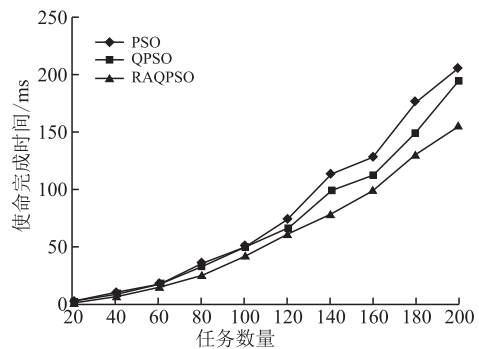


图 2 任务数量增加的任务完成时间  
Fig.2 The task completion time of the task increase number

通过对图 2 任务完成时间的分析,可以得到如下结论:当任务数量较少时,3 种算法完成任务的时间差距很小;随着任务数量的增加,3 种算法得到的最优调度方案完成时间差距逐步增大。当任务数量达到 200 时,QPSO 算法与 RAQPSO 算法相差 39 ms,而 PSO 算法与 RAQPSO 相差 50 ms。

当云系统中任务数量较大时,本文对比了 3 种

粒子群算法的性能。取虚拟机数量为30,任务数量为2 000~20 000,测试数据见图3。

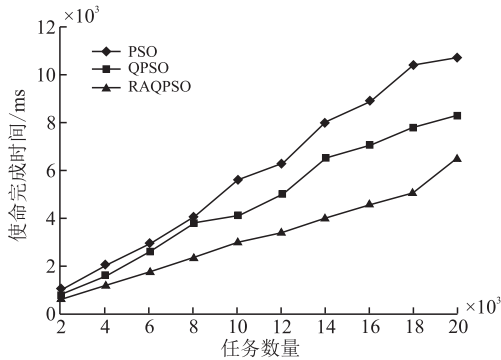


图3 任务数量增加的任务完成时间

Fig.3 The task completion time of the task increase number

大规模任务中,3种算法的性能差距较大。当任务数量达到20 000时,RAQP SO算法调度任务要比PSO算法节省近一半的时间,这对云系统来说具有非常明显的性能提升效果,因此本文提出的RAQP SO算法对于解决云系统中任务调度问题有非常好的效果。

随后,我们比较了给定任务数量,通过增加虚拟机数量来测试3种算法的性能。设定虚拟机参数为:ram=512 MB,imagesSize=10 000 MB,不同虚拟机MIPS为100~600之间的随机整数。图4显示了当任务数量为10 000时,虚拟机数量在10~100之间变化的任务完成时间变化情况。

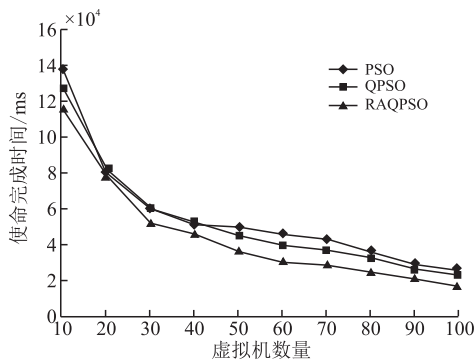


图4 资源数量增加的任务完成时间

Fig.4 The task completion time of the increase in the number of resources

由图4可知,当任务数量相同时,通过3种算法的资源调度来完成的任务,随着计算资源的增加,3种算法调度的任务策略使完成时间逐步递减,RAQP SO算法的调度策略优于PSO和QPSO算法,较为适用于任务数量固定而虚拟机数量增加的调度策略问题。

最后,我们统计了2 000个任务在分配给8个计算资源时的节点负载,3种算法分配任务的负载见图5,该图表明,PSO算法在进行资源分配时节点

负载最不平衡,其次是QPSO,RAQP SO算法相对于前两种算法性能更优;对比算法均不同程度地出现了处理能力强的资源分配到较少的任务,而处理能力弱的资源分配到了较多任务的现象。本文算法具有易于实现、不易陷入局部最优的特点,因此获得了更为理想的调度方案。

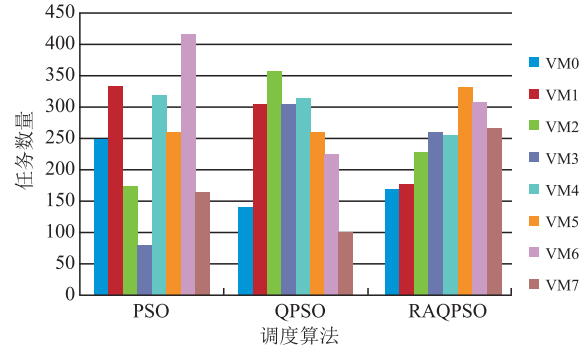


图5 计算资源负载情况

Fig.5 The load of computing resources

## 5 结语

针对云计算环境下资源调度的NP难题,提出了一种改进的QPSO算法——RAQP SO算法。通过采用自适应机制,将适应度函数的变化程度作为惯性权值的更新因子,加快了收敛速度;加入粒子反向学习算子,加强了粒子全局搜索能力。通过实验和仿真分析,验证了RAQP SO算法能够有效减少任务完成时间,同时能够保持各计算节点的负载均衡,为云计算资源调度策略提供了一个新的方法。

## 参考文献(References):

- [1] BUYYA R, YEO C S, VENUGOPAL S, et al. Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility [J]. Future Generation Computer Systems, 2009, 25(6): 599-616.
- [2] ARMBRUST M, FOX A, GRIFFITH R, et al. Above the Clouds: A Berkeley View of Cloud Computing [J]. Eecs Department University of California Berkeley, 2009, 53(4): 50-58.
- [3] 林伟伟, 齐德昱. 云计算资源调度研究综述[J]. 计算机科学, 2012, 39(10): 1-6.  
LIN Weiwei, QI Deyu. Survey of Resource Scheduling in Cloud Computing [J]. Computer Science, 2012, 39(10): 1-6. (in Chinese)
- [4] 周文俊, 曹健. 基于预测及蚁群算法的云计算资源调度策略[J]. 计算机仿真, 2012, 29(9): 239-242, 246.  
ZHOU Wenjun, CAO Jian. Cloud Computing Resource Scheduling Strategy Based on Prediction and

- ACO Algorithm[J]. Computer Simulation, 2012, 29(9):239-242,246.(in Chinese)
- [5] FU Xiong, CANG Yeliang. Task Scheduling and Virtual Machine Allocation Policy in Cloud Computing Environment[J]. Journal of Systems Engineering and Electronics, 2015, (4):847-856.
- [6] ZHOU Zhou, HU Zhigang, SONG Tie. A Novel Virtual Machine Deployment Algorithm with Energy Efficiency in Cloud Computing[J]. Journal of Central South University, 2015, (3):974-983.
- [7] DHALE V D, MAHAJAN A R. Review of Cloud Computing Architecture for Social Computing[C]// International Journal of Computer Applications. 2012;15-19.
- [8] XU Xiaolong, WU Jiaying, YANG Geng. Low-Power Task Scheduling Algorithm for Large-Scale Cloud Data Centers[J]. Journal of Systems Engineering and Electronics, 2013(5):870-878.
- [9] QAMAR S, LAI Niranjana, SINGH Mrityunjay. Internet Ware Cloud Computing: Challenges[J]. IJCSIS, 2010, 7(3):206-210.
- [10] SUN Dawei, CHANG Sungui, GAO Shang. Modeling a Dynamic Data Replication Strategy to Increase System Availability in Cloud Computing Environments[J]. Journal of Computer Science and Technology, 2012, 27(2):256-272.
- [11] 袁浩, 李昌兵. 基于社会力群智能优化算法的云计算资源调度[J]. 计算机科学, 2015, 42(4):206-208, 243.  
YUAN Hao, LI Changbing. Resource Scheduling Algorithm Based on Social Force Swarm Optimization Algorithm in Cloud Computing [J]. Computer Science, 2015, 42(4):206-208, 243.(in Chinese)
- [12] 肖明清, 杨召, 薛辉辉, 等. 云计算及其在测试领域的应用探索[J]. 空军工程大学学报:自然科学版, 2015, 16(1):50-55.  
XIAO Mingqing, YANG Zhao, XUE Huihui, et al. Cloud Computing and Exploration of Its Application to Test Realm[J]. Journal of Air Force Engineering University: Natural Science Edition, 2015, 16(1):50-55.(in Chinese)
- [13] 刘静, 须文波, 孙俊, 等. 基于量子粒子群算法求解整数规划[J]. 计算机应用研究, 2007, 24(3):79-81, 105.  
LIU Jing, XU Wenbo, SUN Jun. Quantum-Behaved Particle Swarm Optimization for Integer Programming[J]. A Plication Research of Computers, 2007, 24(3):79-81, 105.
- [14] 苑帅, 沈西挺, 邵娜娜, 等. 引入人工蜂群搜索算子的QPSO算法的改进实现[J]. 计算机工程与应用, 2016, 52(15):29-33.  
YUAN Shuai, SHEN Xiting, SHAO Nana, et al. Realization of Improved Quantum Particle Swarm Optimization Algorithm Based on Search Operator of Artificial Bee Colony[J]. Computer Engineering and Applications, 2016, 52(15):29-33.(in Chinese)
- [15] 黄宇, 韩璞, 刘长良, 等. 改进量子粒子群算法及其在系统辨识中的应用[J]. 中国电机工程学报, 2011, 31(20):114-120.  
HUANG Yu, HAN Pu, LIU Changliang, et al. An Improved Quantum Particle Swarm Optimization and Its Application in System Identification[J]. Proceedings of the Chinese Society for Electrical Engineering, 2011, 31(20):114-120.(in Chinese)
- [16] 徐文忠, 彭志平, 左敬龙, 等. 基于遗传算法的云计算资源调度策略研究[J]. 计算机测量与控制, 2015, 23(5):1653-1656.  
XU Wenzhong, PENG Zhiping, ZUO Jinglong, et al. Research on Cloud Computing Resource Scheduling Strategy Based on Genetic Algorithm[J]. Computer Measurement & Control, 2015, 23(5):1653-1656.(in Chinese)
- [17] 刘卫宁, 靳洪兵, 刘波, 等. 基于改进量子遗传算法的云计算资源调度[J]. 计算机应用, 2013, 33(8):2151-2153.  
LIU Weining, JIN Hongbing, LIU Bo, et al. Cloud Computing Resource Scheduling Based on Improved Quantum Genetic Algorithm[J]. Journal of Computer Applications, 2013, 33(8):2151-2153.(in Chinese)
- [18] 杨单, 李超锋, 杨健, 等. 基于改进混沌萤火虫算法的云计算资源调度[J]. 计算机工程, 2015, (2):17-20, 25.  
YANG Dan, LI Chaofeng, YANG Jian, et al. Cloud Computing Resource Scheduling Based on Improving Chaos Firefly Algorithm[J]. Computer Engineering, 2015, (2):17-20, 25.(in Chinese)

(编辑:徐楠楠)