

DOI: 10.7652/xjtuxb201710011

采用改进二进制蝙蝠算法的任务调度算法

黄兴旺^{1,2}, 曾学文¹, 郭志川¹, 韩锐¹

(1. 中国科学院声学研究所国家网络新媒体工程技术研究中心, 100190, 北京;

2. 中国科学院大学, 100049, 北京)

摘要: 针对海服务系统环境中任务完成时间过长的问题, 提出一种基于改进二进制蝙蝠算法 (IBBA) 的任务调度算法。该算法通过在二进制蝙蝠算法 (BBA) 寻优过程中引入非线性惯性权重因子, 平衡全局搜索和局部搜索能力; 利用 2 个不同的邻居蝙蝠算子构成扰动项, 避免算法陷入局部最优; 利用自适应学习因子调整全局最优算子和邻居蝙蝠算子的权重, 控制整个寻优过程由全局搜索为主向局部搜索为主过渡。实验结果表明: IBBA 可以稳定地搜索到全局最优值, 与现有的基于二进制粒子群算法 (BPSO) 和二进制蝙蝠算法的任务调度算法相比, 在任务数较多时, 基于 IBBA 的任务调度算法能避免过早收敛, 使任务完成时间显著减少。在海服务网络中, 将该算法应用于任务调度, 可提高处理大粒度服务的效率。

关键词: 任务调度; 二进制蝙蝠算法; 海服务

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 0253-987X(2017)10-0065-06

A Task Scheduling Algorithm Based on an Improved Binary Bat Algorithm

HUANG Xingwang^{1,2}, ZENG Xuwen¹, GUO Zhichuan¹, HAN Rui¹

(1. National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China; 2. University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: A task scheduling algorithm based on an improved binary bat algorithm (IBBA-TA) is proposed to solve the problem of long completion time of tasks in sea service environments. The algorithm introduces nonlinear inertia weight factors in the optimization process of the binary bat algorithm (BBA) to balance capabilities of global and local searches. A perturbation term is constructed by using two mutually exclusive neighbor bats to avoid local optimums. Weights of both the global optimal operator and the neighbor bat operator are adjusted using an adaptive learning factor, and control the transition of the optimization process from global searches to local searches. Experimental results show that IBBA-TA stably obtains the global optimal value. Comparisons with the existing task scheduling algorithms based on the binary particle swarm optimization algorithm (BPSO) and the binary bat algorithm show that when the number of tasks is large, IBBA-TA avoids premature convergence and significantly reduces completion time of tasks. It is concluded that the algorithm can be used for task scheduling to improve the efficiency of processing large granularity services in sea service networks.

Keywords: task scheduling; binary bat algorithm; sea service

收稿日期: 2017-01-17。 作者简介: 黄兴旺(1990—), 男, 博士生; 韩锐(通信作者), 男, 副研究员。 基金项目: 国家高技术研究发展计划资助项目(2015AA015802)。

网络出版时间: 2017-08-19

网络出版地址: <http://www.cnki.net/kcms/detail/61.1069.T.20170819.1125.014.html>

近年来,出现了诸如大规模的媒体任务处理和大数据分析等大粒度服务并得到迅速发展,这些服务通常需要耗费大量的计算资源,云计算^[1]、雾计算^[2]、海计算^[3]等分布式服务架构提供了利用大量分布的虚拟机为这类应用提供服务的可能。为了保证实时性,研究人员提出了利用靠近用户的设备提供服务的计算模式,从而保证带宽可用和低延迟性能。海服务^[4]是中科院提出的基于网络边缘设备就近、自治地协作提供有限的计算、存储和网络服务的新架构,其基本目标是能够为时间敏感的应用提供低延迟的服务。大粒度服务可经过拆分,然后通过海服务将子任务分配到可用资源节点上进行处理,从而充分利用海服务系统中的可用资源,减少任务完成时间。在该任务分配过程中,获得令人满意的任务完成时间是最常见的用户需求,也是海服务网络满足在线实时、准实时的服务请求的关键,成为本文的研究重点。

对于任务分配问题,当任务数和节点数增加到一定程度时,传统的寻优方法存在算法复杂、收敛速度慢等问题。启发式算法由于具备良好的鲁棒性和可行性,在任务调度寻优过程中具备明显的优势^[5]。粒子群算法(particle swarm optimization, PSO)是近年来得到广泛使用的一种启发式算法,其二进制版本 BPSO 继承了 PSO 的优点,在任务调度问题中得到了重点应用。文献[6]提出多种 PSO 算法改进的云计算任务调度算法,但在任务数较多时,性能提升能力下降。文献[7]中根据问题的离散特性,重新定义粒子群算法的位置、速度等操作,达到优化调度长度并降低租赁成本的目标。

蝙蝠算法(bat algorithm, BA)^[8]是一种通过模拟蝙蝠觅食定位行为的启发式算法,汲取了粒子群算法和模拟退火算法的优点,具有媲美粒子群算法的简单、可实施性和优于粒子群算法的性能^[9]。该算法利用脉冲响度和频率控制全局搜索和局部搜索能力的平衡,在逐次迭代过程中根据当前最优解调整速度和位置向量,在发现更优蝙蝠位置时才更新脉冲响度和频数,保证蝙蝠向目标靠近^[9]。BA 及其二进制版本 BBA^[9]已在特征选取^[10]、子集问题^[11]、工作流调度^[12]等多个重要工程问题中得到应用。本文针对基本二进制蝙蝠算法过早收敛的问题^[13]进行改进,提出一种改进的二进制蝙蝠算法(improved binary bat algorithm, IBBA),并将其应用于海服务任务调度中。

1 任务调度模型

1.1 问题描述

在传统的将任务集中进行处理的模式中,随着任务的数据量及复杂度增加,任务处理所需时间急剧增加,分布式的计算资源不能得到充分利用,造成严重的负载不均。在海服务系统架构下,任务分解技术可以有效提升任务处理效率,均衡网络负载。目前的任务分解技术主要可分为两类:一类是将任务分解为彼此相互独立的子任务;另一类是将其分解为彼此相互依赖的子任务。本文提出的任务调度算法建立在子任务相互独立的基础上。

任务调度问题可以定义如下:在满足约束条件的前提下,将 n 个相互独立的任务集 $\{t_1, t_2, \dots, t_n\}$ 分配到 $m(m < n)$ 个异构可用资源集 $\{v_1, v_2, \dots, v_m\}$ 上,使得任务完成总时间最少,资源得到充分利用。本文基于海服务为时间敏感的应用提供低延迟服务的设计思想,选取任务完成总时间最少为优化目标,从而使得海服务网络能够提供在线实时或准实时的服务。

1.2 任务完成时间最小化模型

下面给出具有 n 个相互独立的任务、 m 个节点的海服务分布式系统任务完成时间最小化模型

$$\min T = \max\{C_i\}, \quad i \in \{1, 2, \dots, m\} \quad (1)$$

令 $C_{i,j}(i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, n\})$ 表示任务 t_j 在资源节点 v_i 上执行所需的时间,则 C_i 表示完成分配给资源节点 v_i 执行的任务所需的总时间。本文假定每个任务只能分配给一个资源节点,每个资源节点处理完一个任务后,才能接着处理下一个任务。

2 改进二进制蝙蝠算法

通过分析基本二进制蝙蝠算法存在的问题,提出一种改进二进制蝙蝠算法 IBBA。

2.1 现有蝙蝠算法及二进制蝙蝠算法

BA 算法及其二进制版本 BBA 作为一种启发式算法,具有简单、快速等特点,但是在很多工程领域也存在着高维时容易陷入局部最优、过早收敛以及全局搜索和局部搜索不能很好的平衡等问题。

目前已有研究人员对 BA 的不足进行了改进研究。文献[14]提出混沌蝙蝠算法,使用多种映射替代 BA 的均匀分布本地搜索操作;文献[15]提出混合 BA 和差分进化算法(differential evolution, DE)的混合蝙蝠算法,利用 DE 本地搜索策略提升 BA

的局部搜索能力。上述研究主要在于改进 BA 的本地搜索策略,从而提升局部搜索能力,但是对于二进制离散情况并不适用。文献[13]针对连续域的优化问题,提出了 3 种改进方法,其中第 2 个改进综合考虑了全局搜索和局部搜索的平衡,但是邻居蝙蝠的随机选取策略过于简单且只有一维变化,在维数较高时不利于跳出局部最优。在分布式系统任务调度环境中,优化目标函数是高维二进制离散函数,并且存在多个约束,因此需要一种能够避免陷入局部最优、更高精度的二进制优化算法。

2.2 改进二进制蝙蝠算法

由于 BBA 算法与 BPSO 算法具有相似的速度更新公式,因而 BBA 算法也同样具有类似于 BPSO 的不足。分析 BBA 的速度更新公式(1)可知:由于其第 1 项速度项在整个搜索过程中都占据比较大的权重,容易导致后期寻优步长太大,局部搜索能力快速下降;第 2 项由于只有当前最优解信息可供参考,导致容易陷入局部最优。

文献[13]表明,利用邻居蝙蝠信息可以提升跳出局部最优的能力,获得全局最优解。受文献[16]启发,增加由两个邻居蝙蝠构成的扰动项可使搜索朝多个方向进行,能够提升全局搜索能力。受差分进化算法变异算子^[16]启发,本文提出改进二进制蝙蝠算法新的速度更新方程,表示如下

$$\mathbf{V}_i(t+1) = w(\mathbf{V}_i(t)) + (\mathbf{X}_i(t) - \mathbf{X}_g)F_i\delta_1 + (\mathbf{X}_{k_1}(t) - \mathbf{X}_{k_2}(t))F_i\delta_2 \quad (2)$$

$$\delta_1 + \delta_2 = 1 \quad (3)$$

式中: w 表示惯性权重因子,通过控制速度的大小,平衡第 i 只蝙蝠的局部搜索能力和全局搜索能力; $\mathbf{X}_{k_1}$ 、 $\mathbf{X}_{k_2}$ 表示从当前蝙蝠群中随机选取的 2 个互斥邻居蝙蝠($i \neq k_1 \neq k_2$),其差 $(\mathbf{X}_{k_1}(t) - \mathbf{X}_{k_2}(t))$ 构成随机扰动项,用于增强全局寻优能力; δ_1 是当前全局最优解 $\mathbf{X}_g$ 的自适应学习因子,取值范围为 $0 \sim 1$; δ_2 是随机扰动项 $(\mathbf{X}_{k_1}(t) - \mathbf{X}_{k_2}(t))$ 的自适应学习因子,取值范围为 $1 \sim 0$ 。当 δ_1 增大时, δ_2 相应减小,当前全局最优解 $\mathbf{X}_g$ 的影响就会逐渐大于随机扰动项 $\mathbf{X}_{k_1}(t) - \mathbf{X}_{k_2}(t)$;反之亦然。 δ_1 随迭代次数的自适应更新方程如下

$$\delta_1 = 1 + (\delta_0 - 1)((t_{\max} - t)/t_{\max})^n \quad (4)$$

式中: δ_0 表示 δ_1 的初始值; $t_{\max}$ 表示最大迭代次数; t 表示当前迭代次数; n 表示非线性调制指数。当 t 增加时, δ_1 将从 δ_0 增大到 1, δ_2 相应从 $(1 - \delta_0)$ 减小到 0,从而控制搜索过程由全局搜索逐渐转向局部搜索。

文献[13]表明,采用非线性递减的惯性权重策略控制速度大小能够获得更优解。本文采用如下非线性惯性权重策略

$$w = ((t_{\max} - t)/t_{\max})^n(w_0 - w_f) + w_f \quad (5)$$

式中: w_0 和 w_f 分别表示初始和最终的惯性权重。

蝙蝠位置仍然采用文献[9]提出的策略,即通过 V-Shape 转换函数进行离散化来表示蝙蝠位置

$$X_{in}(t+1) = \begin{cases} (X_{in}(t))^{-1}, & r \leq f(V_{in}(t+1)) \\ X_{in}(t), & r > f(V_{in}(t+1)) \end{cases} \quad (6)$$

$$f(V_{in}(t+1)) = |(2/\pi)\arctan((\pi/2)V_{in}(t))| \quad (7)$$

式中: $V_{in}(t)$ 、 $X_{in}(t)$ 分别表示在第 t 次迭代中第 i 只蝙蝠在第 n 维的取值, $(X_{in}(t))^{-1}$ 则是 $X_{in}(t)$ 的二进制取反, r 为 $[0, 1]$ 之间的随机数。

通过上述改进,IBBA 能够提升可行解在二进制搜索空间中的分布,增强蝙蝠跳出局部最优的能力以获得更优解。

3 采用 IBBA 的任务调度算法

本节将前面提出的 IBBA 算法运用于海服务任务调度问题中,目标函数和约束条件已由前面提出的任务完成总时间最小化模型给出。

3.1 蝙蝠编码

为了表示算法迭代过程中的任务调度结果,使用由 0 和 1 组成的矩阵 $\mathbf{X}_i$ 表示每只蝙蝠的位置,进行编码。矩阵 $\mathbf{X}_i$ 表示如下

$$\mathbf{X}_i = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ x_{31} & x_{32} & \cdots & x_{3n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \quad (8)$$

式中:如果子任务 t_j 被分配到资源节点 v_i 上执行,则 $x_{ij} = 1, i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, n\}$; 否则, $x_{ij} = 0$ 。

3.2 适应度值

蝙蝠算法的蝙蝠位置优劣由适应度函数衡量,而适应度函数通常由求解问题的目标函数决定。由前面所述的任务调度模型可知,本文的任务调度目标为最小化任务完成总时间,适应度函数为

$$F(x) = T = C_{\max}(s) \quad (9)$$

式中: $s = \{x_{11}, \dots, x_{ij}, \dots, x_{mn}\}$ 表示一个可行解;令 C_i 表示资源节点 v_i 完成其分配到的任务所需要的时间,则 $C_{\max}(s) = \max\{C_i\}$ 表示任务完成的总时间。

3.3 IBBA 算法流程

- IBBA 算法流程如下。
- 步骤 1** 初始化蝙蝠种群和相关参数。
- 步骤 2** 根据算法迭代次数,依据原 BBA 算法的频率更新方程、本文改进的速度更新方程以及原 BBA 算法的位置更新方程,进行频率、速度和位置更新。
- 步骤 3** 如果满足蝙蝠算法变异操作条件,则利用当前全局最优解更新蝙蝠位置进行变异操作。
- 步骤 4** 计算当前蝙蝠的适应度,若比当前全局最优解更优并且满足声波响度条件,则更新当前全局最优解,并更新声波响度和频率值;否则,转向步骤 5。
- 步骤 5** 判断蝙蝠群的迭代次数是否达到最大迭代次数,如果达到则退出调度寻优过程;否则,转向步骤 2。

4 实验与结果分析

为了验证 IBBA 算法的有效性,选用 CloudSim 仿真工具^[17]进行仿真,并在 CloudSim 的 Data-centerBroker 中实现 IBBA 算法。为了确定 IBBA 算法的性能,选取具有参考性的基于 BPSO 和 BBA 的调度算法以及基于文献[13]第 2 个改进的二进制版本 BEBA 的调度算法进行比较。在 CloudSim 中的仿真过程如下:①创建 1 个采用默认设置的数据中心;②在该数据中心创建 2 个采用主流配置的主机模拟网络边缘设备,2 台主机都具有 2 GB 的随机存储器、1 TB 的存储、1 Gb/s 的带宽,并且采用时间共享策略将虚拟机调度到主机中,此外,第 1 台主机配备双核处理器 Intel 2 Extreme X6800,第 2 台处理器配备六核处理器 Core i7 Extreme Edition 3960X;③主机中的虚拟机个数主要影响寻优过程耗时,不会对本文算法和所对比算法的性能产生影响,不失一般性,在 2 台主机上分别创建 2 个和 3 个虚拟机,模拟网络边缘设备中的虚拟机节点,在任务数不同的情况下保持虚拟机的个数和配置一致,每个虚拟机采用 Intel Pentium 4 Extreme Edition 处理器,配备 10 GB 大小的镜像,0.5 GB 的内存,100 Mb/s 的带宽,并且采用时间共享调度策略将任务分发给虚拟机;④采用来自瑞典高性能计算中心 HPC2N 的数据集 HPC2N-2002-2.1-cln.swf^[18]生成待分配任务;⑤分别使用实现了 IBBA、BBA、BP-SO 和 BEBA 的 DatacenterBroker 进行任务调度。

为了评估 4 种方法在不同维度下的性能表现,分别选取 HPC2N 数据集的前 100、200、300、400、500 条任务记录生成 5 组待分配任务进行实验。采

用各种算法常用的参数进行初始化,种群规模都设为 50,IBBA 和 BEBA 的非线性调制指数取值均为 2,每种方法实验 30 次,最大迭代 500 次,取平均值。从任务提交到任务完成所需平均时间如图 1 所示。表 1 给出了任务完成时间的均值和方差。由图 1 和表 1 可知,本文提出的 IBBA 算法在任务数较多(任务数为 300~500)的情况下具有非常明显的优势,能够以更短的时间完成任务。

为了确定 IBBA 算法得到的任务完成时间与其他 3 种算法相比是否有明显减少,本文采用统计学 student *t* 检验^[13]。*t* 检验值计算公式为

$$t = (M_1 - M_2) / (S_1^2 / (n_1 - 1) + S_2^2 / (n_2 - 1))^{1/2} \quad (10)$$

式中: M_1 、 S_1 和 n_1 分别表示第 1 个样本的均值、方差以及样本大小; M_2 、 S_2 和 n_2 分别表示第 2 个样本的均值、方差以及样本大小。选取 BPSO、BBA 和 BEBA 算法所得结果作为第 1 个样本,IBBA 算法所得结果作为第 2 个样本,分别进行 student *t* 检验计算,样本大小为虚拟机数与任务数的乘积。

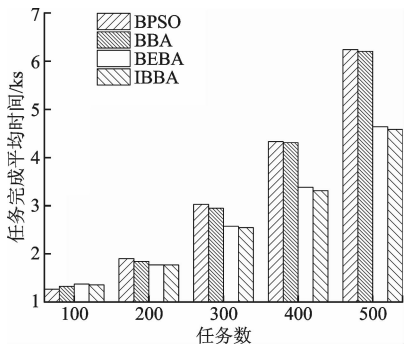


图 1 任务完成时间的对比

置信区间设为 95%,即 $t_{0.05} = 1.96$,则当 $t > 1.96$ 时,第 2 个样本明显优于第 1 个样本;当 $t < -1.96$ 时,第 1 个样本明显优于第 2 个样本;当 t 值在 $[-1.96, 1.96]$ 之间时,2 个样本在统计意义上没有明显区别。表 2 给出了 IBBA 与另外 3 种算法根据表 1 结果进行 *t* 值检测计算的结果(保留 2 位小数)。由表 2 可知,当任务数为 100 时,BPSO 和 BBA 的性能优于 IBBA,而 BEBA 与 IBBA 无明显区别;当任务数为 200 时,IBBA 所得结果仍然与 BEBA 持平,但已经明显优于 BPSO 和 BBA 所得结果;当任务数再增加(任务数为 300~500)时,IBBA 算法所得结果明显优于第 1 个样本。

本文还给出了每种方法在任务数为 300 时的 1 000 次迭代过程的适应度的变化,如图 2 所示。由图 2 可知,BBA 算法在 150 次迭代后就无法继续提

表 1 任务完成时间的均值和方差

任务数	任务完成时间的均值/s				任务完成时间的方差/s			
	BPSO	BBA	BEBA	IBBA	BPSO	BBA	BEBA	IBBA
100	1 260.6	1 320.9	1 368.6	1 351.6	153.07	174.85	232.4	240.20
200	1 895.7	1 837.8	1 766.4	1 764.5	192.38	246.99	262.2	217.47
300	3 022.9	2 941.3	2 568.1	2 541.3	244.08	388.12	321.2	299.57
400	4 324.3	4 301.5	3 375.5	3 307.0	382.18	455.53	352.0	237.89
500	6 232.3	6 197.2	4 632.4	4 577.6	465.37	511.63	455.1	337.05

表 2 3 种算法与 IBBA 算法比较的 student *t* 检验值

任务数	<i>t</i>		
	BPSO	BBA	BEBA
100	−7.14	−2.31	1.14
200	14.28	7.04	0.18
300	48.25	31.59	2.36
400	101.04	86.52	7.21
500	145.70	133.78	4.84

高,BPSO、BEBA 以及 IBBA 都在 600 次迭代后才无法继续搜索到更优值。与 BPSO、BBA 以及 BEBA 算法相比,IBBA 算法跳出局部最优能力更强,拥有更好的全局搜索能力。通过调整惯性权重和利用邻居蝙蝠信息,整个寻优过程能够很好地平衡全局搜索和局部搜索,避免陷入局部最优,从而获取更优解。

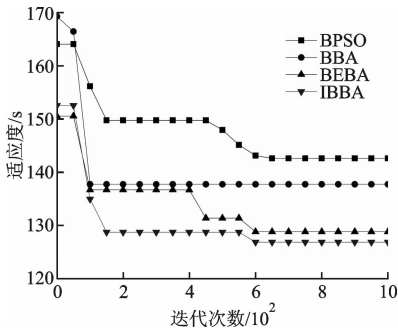


图 2 任务数为 300 时适应度的变化

5 结 论

本文主要研究了海服务系统环境下任务调度问题,针对其离散、复杂度高的特点,提出一种采用 IBBA 的任务调度算法。IBBA 具有基本二进制蝙蝠算法的特点,通过引入 2 个邻居蝙蝠构成的扰动项避免陷入局部最优;通过非线性权重以及自适应学习因子平衡寻优过程中的全局搜索和局部搜索。实验仿真结果表明,任务数较多时,与现有二进制蝙

蝠算法及二进制粒子群算法相比,IBBA 算法能够避免过早收敛于局部最优,在精度上也有明显提高,显著减少完成任务所需时间。该算法可应用于海服务网络任务调度中,提升处理大粒度服务的效率。

参考文献:

[1] ARMBRUST M, FOX A, GRIFFITH R, et al. A view of cloud computing [J]. Communications of the ACM, 2010, 53(4): 50-58.

[2] BONOMI F, MILITO R, ZHU J, et al. Fog computing and its role in the internet of things [C]// Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing. New York, USA: ACM, 2012: 13-16.

[3] 封松林, 宁德军, 张红广, 等. 面向海云协同的共性服务综述 [J]. 网络新媒体技术, 2014, 3(1): 1-7. FENG Songlin, NING Dejun, ZHANG Hongguang, et al. A survey on sea-cloud collaboration oriented common service [J]. Journal of Network New Media, 2014, 3(1): 1-7.

[4] 王劲林, 田静, 尤佳莉, 等. 一种现场、弹性、自治的网络服务系统: 海服务系统研究与设计 [J]. 中国科学: 信息科学, 2015, 45(10): 1237-1248. WANG Jinlin, TIAN Jing, YOU Jiali, et al. Research and design of an on-site, elastic, autonomous network: sea service system [J]. Scientia Sinica: Informationis, 2015, 45(10): 1237-1248.

[5] QIAO Nannan, YOU Jiali, SHENG Yiqiang, et al. An efficient algorithm of discrete particle swarm optimization for multi-objective task assignment [J]. IE-ICE Transactions on Information and Systems, 2016 (12): 2968-2977.

[6] KHALILI A, BABAMIR S M. Makespan improvement of PSO-based dynamic scheduling in cloud environment [C]// Proceedings of the 2015 23rd Iranian Conference on Electrical Engineering. Piscataway, NJ, USA: IEEE, 2015: 613-618.

[7] 李健, 黄庆佳, 刘一阳, 等. 云计算环境下的大规模

- 图状数据处理任务调度算法 [J]. 西安交通大学学报, 2012, 46(12): 116-122.
- LI Jian, HUANG Qingjia, LIU Yiyang, et al. A task scheduling algorithm for large graph processing cloud in computing [J]. Journal of Xi'an Jiaotong University, 2012, 46(12): 116-122.
- [8] YANG X S. A new metaheuristic bat-inspired algorithm [J]. Computer Knowledge & Technology, 2010, 284: 65-74.
- [9] MIRJALILI S, MIRJALILI S M, YANG X S. Binary bat algorithm [J]. Neural Computing and Applications, 2014, 25(3): 663-681.
- [10] RODRIGUES D, PEREIRA L A M, NAKAMURA R Y M, et al. A wrapper approach for feature selection based on bat algorithm and optimum-path forest [J]. Expert Systems with Applications, 2014, 41(5): 2250-2258.
- [11] 刘艺, 刁兴春, 曹建军, 等. 求解子集问题的鲶鱼效应蝙蝠蚁群优化 [J]. 系统工程与电子技术, 2016, 38(10): 2441-2448.
- LIU Yi, DIAO Xingchun, CAO Jianjun, et al. Catfish bat algorithm-ant colony optimization for subset problems [J]. Systems Engineering and Electronics, 2016, 38(10): 2441-2448.
- [12] RAGHAVAN S, MARIMUTHU C, SARWESH P, et al. Bat algorithm for scheduling workflow applications in cloud [C] // Proceedings of the International Conference on Electronic Design, Computer Networks & Automated Verification. Piscataway, NJ, USA: IEEE, 2015: 139-144.
- [13] YILMAZ S, KÜÇÜKSİLİ E U. A new modification approach on bat algorithm for solving optimization problems [J]. Applied Soft Computing, 2014, 28(5): 259-275.
- [14] GANDOMI A H, YANG X S. Chaotic bat algorithm [J]. Journal of Computational Science, 2013, 5(2): 224-232.
- [15] FISTER I, FISTER D, YANG X S. A hybrid bat algorithm [J]. Electrotechnical Review, 2013, 80(1/2): 1-7.
- [16] GAO W F, LIU S Y. A modified artificial bee colony algorithm [J]. Computers and Operations Research, 2012, 39(3): 687-697.
- [17] 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 and Experience, 2011, 41(1): 23-50.
- [18] FEITELSON D G, DAN T, KRAKOV D. Experience with using the parallel workloads archive [J]. Journal of Parallel and Distributed Computing, 2014, 74(10): 2967-2982.

(编辑 武红江)

(上接第52页)

- [10] SCHMIT J M, RILEY M A, DALVI A, et al. Deterministic center of pressure patterns characterize postural instability in Parkinson's disease [J]. Experimental Brain Research, 2006, 168(3): 357-367.
- [11] MOHEBBI M, GHASSEMIAN H, ASL B M. Structures of the recurrence plot of heart rate variability signal as a tool for predicting the onset of paroxysmal atrial fibrillation [J]. Journal of Medical Signals and Sensors, 2011, 1(2): 113-121.
- [12] ACHARYA U R, FAUST O, SREE V, et al. Linear and nonlinear analysis of normal and CAD-affected heart rate signals [J]. Computer Methods and Programs in Biomedicine, 2014, 113(1): 55-68.
- [13] RANGAPRAKASH D, PRADHAN N. Study of phase synchronization in multichannel seizure EEG using nonlinear recurrence measure [J]. Biomedical Signal Processing and Control, 2014, 11: 114-122.
- [14] JOSINSKI H, MICHALCZUK A, MUCHA R, et al. Analysis of human motion data using recurrence plots and recurrence quantification measures [C] // Asian Lecture Notes in Computer Science: 9622. Berlin, Germany: Springer, 2016: 397-406.
- [15] WEBBER C L Jr, ZBILUT J P. Dynamical assessment of physiological systems and states using recurrence plot strategies [J]. Journal of Applied Physiology, 1994, 76(2): 965-973.
- [16] WOLF A, SWIFT J B, SWINNEY H L, et al. Determining Lyapunov exponents from a time series [J]. Physica: D Nonlinear Phenomena, 1985, 16(3): 285-317.

(编辑 杜秀杰)