文章编号:1671-4598(2018)09-0195-05 DOI:10.16526/j.cnki.11—4762/tp.2018.09.042 中图分类号:TP274 文献标识码:A

云计算中基于 IABC 算法的负载预测的研究

史振华
(绍兴职业技术学院,浙江 绍兴 312000)

摘要:针对云计算中的资源负载预测的问题,采用改进的人工蜂群算法和 SVM 相互结合的方式构建预测模型,首先通过采用反向学习对种群进行初始化,差分进化对种群个体进行选择,吸引点策略构建算法的蜜源选择,通过反馈机制和森林法则降低算法陷入局部最优的缺点,其次,将 SVM 预测模型中参数通过改进的蜂群算法进行优化得到了最佳的参数,最后仿真实验中,提出的 IABC 算法预测精度优于 SVM,LSSVM 等预测算法,具有一定的推广价值。

关键词:云计算;资源负载;人工蜂群算法;SVM

Research on Load Prediction of Cloud Computing Based on IABC Algorithm

Shi Zhenhua
(Shaoxing Vocation & Technical College, Shaoxing 312000, China)

Abstract: Aiming at the problems about resources load prediction appear in the cloud computing, the IABC algorithm is used with the combination of SVM to construct model of prediction. Firstly, backward learning is used for population initialization, the individuals of the population should be selected by differential evolution, constructing the nectar selection of the algorithm by the strategy of attraction points, and decreasing the disadvantage of being trapped into partial bests by feedback mechanism and the law of the jungle. Secondly, refining the parameters in SVM prediction model by IABC algorithm to have best parameters. Lastly, in simulation experiences, the prediction of IABC algorithm is better than that of predicting algorithm like SVM, LSSVM and so on. So, the prediction of IABC algorithm is worth being promoted.

Keywords: cloud computing; resource load; IABC algorithm; SVM

0 引言

云计算中资源负载关系到云服务质量的高低,通过预测未来资源负载能够有效的提高云计算的资源调度效率[1]。由于云计算资源具有实时性,动态性等特点,导致了资源负载预测同样具有不确定性和非线性[2]等特点,因此,国外学者使用很多的方法对其进行研究,文献[3-6]主要是以静态的对象研究为主,但是无法达到很好的预测效果。国内学者在此基础上进行了相应的研究,对一些算法[7-12]从不同的角度提出了预测方法,文献[13-17]从不同的角度提出了云计算资源的预测方法,都取得了比较好的效果。

本文将人工蜂群算法与 SVM 相结合构建预测模型,通过对人工蜂群算法的种群进行初始化,采用差分进化算法对个体选择、吸引子策略构建蜜源选择、使用反馈机制和森林法则等措施提高人工蜂群算法的性能,并进一步优化 SVM 中参数,提高了预测精度。

1 人工蜂群算法简述及不足

人工蜂群算法(Artificial Bee Colony Algorithm,简称 ABC)是一种根据蜜蜂采蜜行为而受到启发提出的一种群智能优化算法,其工作的过程是每一个蜂群个体之间进行分工与信息分享、信息交流,从而相互合作完成采蜜工作,ABC 算法具有操作简单、参数控制少、鲁棒性强的优点。特别是 ABC 算法用于处理某个最优问题的时候,其算法中对应的蜜源的位置就是优化问题中的可行解,而蜜源中花粉的数量对应优化问题可行解的质量高低,蜜蜂寻找花蜜速度对应可行解的优化速度,获得最大的花蜜量对应着优化问题的最优解。首先,算法随机产生 N 个初始解,采蜜蜂的数量为 N,将蜜源与采蜜蜂进行一对一对应,其次,当采蜜蜂进行搜索蜜源的时候会移动到新位置,通过将新位置的花蜜与上一个位置的花蜜的数量进行对比,选择优质的蜜源,在采蜜蜂完成全部的搜索后,将搜索结果对比花蜜容量,从中选择优质蜜源,当所有的采蜜蜂完成搜索之后,观察蜂会根据采蜜蜂提供的蜜源信息以一定的概率选择蜜源,最后,判断蜜源花蜜量经过有限次的搜索之后得到最优解即为最优质的蜜源。在 ABC 算法中,蜜蜂被分为三类,分别是雇佣蜂、跟随蜂和侦察蜂。这 3 种蜜蜂的搜索行为如下:

(1) 雇佣蜂搜索。算法中的雇佣蜂主要是为了能够围绕在特定的蜜源周围随时进行搜索,当需要在另一个蜜源附近进行开采花蜜的时候,雇佣蜂在该蜜源附近进行新的搜索,其搜索公式如(1)。当寻找到新的蜜源之后,进行

收稿日期:2018-01-29; **修回日期**:2018-03-08。

作者简介:史振华(1980-),男,硕士,副教授,主要从事云计算,算法处理方向的研究。

适应度（公式 2）方面的评价，并与上一个蜜源的适应度对比，如果高于上一个蜜源，则进行依靠。

$$v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \tag{1}$$

$$v'_{ij} = \begin{cases} v_{ij} & if(fit(v_{ij}) > fit(x_{ij})) \\ x_{ij} & else \end{cases} \tag{2}$$

式中，φ_{ij} 表示在［－1，1］中的随机数，$fit(v)$ 是适应度评价函数。

（2）跟随蜂搜索。雇佣蜂主要将蜂蜜送到蜂巢之后，会邀请跟随蜂一起飞向蜜源。跟随蜂是否接受邀请主要与雇佣蜂依靠的蜜源质量具有一定的关系，通常来说当蜜源的质量越高，就能够吸引到更多的跟随蜂前往，反之，则会失去更多的跟随蜂，导致该蜜源被放弃，跟随蜂选择的概率如公式（3）所示。

$$P_i = \frac{fit(x_i)}{\sum_{i=1}^{D} fit(x_i)} \tag{3}$$

（3）侦察蜂搜索。当蜜源采摘完毕之后，在蜜源附近的雇佣蜂就变成了侦察蜂，从而进行全局搜索。其搜索的公式如（4）所示。

$$X_i = X_{\min} + R \cdot (X_{\max} - X_{\min}) \tag{4}$$

式中，R 为［－1，1］之间的随机数。

虽然该算法在很多实际问题中被广泛的应用，但还是存在一定的不足，比如种群的初始化是随机的，没有考虑对整个种群的共享信息的有效利用，导致算法在进行局部搜索的过程中能力差且陷入局部最优，收敛速度还需要进一步提高等问题。

2 人工蜂群算法的改进

2.1 种群初始化及其个体选择

基本的 ABC 算法的种群是没有初始化的即位置是随机的，这样会导致种群在分布的时候不均匀，因此整个算法的性能也会受到一定的影响，为了能够有效的避免这种情况的发生，有效的提高算法的搜索效率。本节采用反向学习的策略对种群进行初始化，用来增加种群多样性，以便更好的产生最优解。其解决的策略是在反向学习策略中，将一个可行解计算出对应的反向解，然后将这两个解进行合并排序，按照设定的条件选出一个个体作为下一代个体。其步骤如下：

步骤 1：初始化种群规模 N。

步骤 2：随机初始化阶段。

$$for\ i = 1{:}N\ do$$
$$for\ j = 1{:}D\ do$$
$$X_i^j = X_{\min}^j + rand(0,1)(X_{\max}^j - X_{\min}^j)$$
$$end\ for$$
$$end\ for$$

步骤 3：执行反向学习阶段。

$$for\ i = 1{:}N\ do$$
$$for\ j = 1{:}D\ do$$
$$oX_i^j = X_{\min}^j + X_{\max}^j - X_i^j$$
$$end\ for$$
$$end\ for$$

步骤 4：从 $\{X(N) \cup OX(N)\}$ 中选择适应值最好的前 N 个值作为种群初始解。

在种群的初始解初始阶段，首先通过对种群中个体进行差、变异等操作重新得到一个新的种群，其次与父代个体进行交叉操作，计算种群中的个体适应度值进行排列，从中选择出优秀的种群个体，得到新一代群体。最后进行包括变异、交叉和选择 3 个操作的进化过程.

（1）变异。对于个体 x_i，按照如下生成变异个体：

$$u_i = x_{r1} + F(x_{r2} - x_{r3}) \tag{5}$$

式中，x_{r1}，x_{r2} 和 x_{r3} 从进化种群中随机选取 3 个个体，设定 F 为缩放比例因子用以控制向量产生的影响。

（2）交叉。为了进一步增加种群多样性，引入差分进化算法：

$$v_{i,j} = \begin{cases} u_{i,j}, rand(0,1) \leqslant CR \\ x_{i,j}, rand(0,1) > CR \end{cases} \tag{6}$$

式中，$j = 1,2,\cdots D$，D 为空间维数，CR 为 0 到 1 之间的概率。

（3）选择。使用贪婪策略，将交叉后的个体 v_i 和父代个体 x_i 按照公式（7）生成子代个体：

$$x_{i+1} = \begin{cases} v_i, f(v_i) \leqslant f(x_i) \\ x_i, f(v_i) > f(x_i) \end{cases} \tag{7}$$

对个体进行差分进化算法步骤如下。

步骤 1：初始化种群，对种群进行评价。

步骤 2：对其中的个体按公式执行（5）产生变异。

步骤 3：对个体和变体执行（6）和（7）生成新的个体。

步骤 4：对新的个体组成的新的种群进行总体评价，从而确定下一代种群。

2.2 跟随蜂的蜜源选择

在 ABC 算法中，跟随蜂是通过轮盘赌策略来选择蜜源的，虽然能够保证优秀的蜜源能被选中，但有的时候从整体上也会错失更加优秀的蜜源，显然这样会延长耗费计算的资源，导致迭代时间延长，本文算法在遵守这种理念初衷的前提下，提出一种“吸引点”策略，通过引入 cr 来改变跟随蜂的搜索方式。所有的跟随蜂全部以吸引点为中心的周围等比例的缩放来共同开发，从而从整体上有效的提高算法的开发能力。而吸引点 cr 作为蜂群中的“蜂王”，按照一定的比例范围吸引所有跟随蜂靠近它，搜索示意如图 1 所示。白色框表示吸引子，黑色球表示不同的种群的初始位置，三角形表示围绕吸引点后的种群所在的新位置。

本文对于吸引点的获得按照两种方式来进行。第一种情况是当种群的个体获得全局最优解的时候，吸引点 cr 的值通过公式（8）得到，第二种情况就是没有获得全局最优解的时候，吸引点按（9）得到：

$$cr = v_{best} * r_1 + r_2 \tag{8}$$

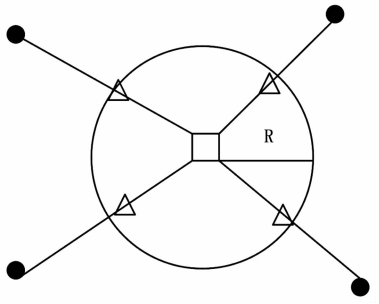

图 1　搜索示意图

$$cr = v_i + (v_{best} - v_i) * r_3 \tag{9}$$

式中，v_i 表示当前可行解的蜜源，r_1，r_2，r_3 是一组随机数目，且 $r_1+r_2+r_3=1$，吸引点 cr 作为整个蜂群的中心，而每一个侦察蜂都以它为中心，从远到近根据一定的比例进行缩放，这样能够有效的降低个体飞越边界的可能性，同时个体的整体算法的寻优能力得到了加强，整体算法开发能力也逐步增强。因此对种群个体进行位置更新为：

$$x'_j = \begin{cases} (x_j - cr_j)\%((cr_j - x_{\min}) * r)/r \; if \; x_j < cr_j \\ (x_j - cr_j)\%((x_{\max} - cr_j) * r)/r \; else \end{cases}$$

式中，$x_{\max}$ 和 $x_{\min}$ 是种群中的上限和下限，在算法的每一步进化过程中，r 的值恒定，保证种群个体不丢失原先的社会信息。

2.3　反馈机制和森林法则

为了进一步解决 ABC 算法陷入局部最优，容易产生收敛速度慢的问题，本文将反馈机制和森林法则进行整体融合，其主要思想是在 ABC 算法的全局搜索的过程中引入反馈机制，这样可以扩大算法的感知的范围，因此整体上提高算法的开发能力和探索能力，通过线性微分递增来平衡算法中的开发能力和探索能力，模拟森林法则中的优胜劣汰的方式，随机的选择较差的个体进行初始化。

（1）反馈机制：

从雇佣蜂搜索公式中可以发现，ABC 算法主要用于搜索而不是进行开发，随着算法的不断深入，如何能够更好的开发是算法后期中薄弱环节，特别是从平衡算法的角度出发，其探索能力和开发能力上如何解决算法的收敛能力是非常重要的，本文根据粒子群算法的中的全局最优解的概念启发，在搜索公式中引入线性微分递增策略和全局最优解的思路来解决以上问题，采用公式如（10），（11）：

$$v_{ij} = W(t) * x_{ij} + r_{ij} \mid x_{kj} - x_{ij} \mid + \varphi_{ij}(x_{gj} - x_{ij})$$

$$s.t.\, r_{ij} = \begin{cases} +m * rand(0,1), d = 1 \\ -m * rand(0,1), d = 0 \end{cases} \tag{10}$$

$$\frac{dw(t)}{dt} = 2\,\frac{w_{\max} - w_{\min}}{T^2} \times t \tag{11}$$

式中，x_{gj} 表示全局最优解，x_{ij} 代表当前解，x_{kj} 是当前不同于 x_{ij} 的一个随机解，r_{ij} 是一个属于（−1，1）之间的随机数，$w_{\max}$ 和 $w_{\min}$ 分别表示自适应因子中的最大值和最小值，设 T 为最大迭代次数，t 为当前迭代次数。在公式（10）的描述中，r_{ij} 是不确定的随机数，通过相应的反馈机制使得最优解的蜜源的质量能够在上一代蜜源中得到更新，因此说明目前的方向是正确的，因此可以继续在该方向上搜索，反之，则向相反的方法搜索。当上一代蜜源得到更新的时候，$d=1$，否则 $d=0$，通过反馈机制，可以直接搜索可能存在最优解的区域。

（2）树林法则。

自然界中存在这样一种优胜劣汰的现象。速度慢的动物往往会被凶猛的动物吃掉，本文中的适应度差的解就好比速度慢动物，容易被重新初始化。在算法中，适应度差的解个体周围的其他解的适应度也比较差，对这些个体按照随机原则重新初始化，设定全局适应度最差的个体 g_w 的领域半径范围如公式（12）。

$$Range = \sqrt{\sum_{i=1}^{D}(x_{gj} - g_{wj})^2} \tag{12}$$

式中，$Range$ 表示全局适应度最差个体的领域半径，本文将全局最优解与最差解之间的欧式距离作为领域范围的半径。显然，根据公式（12）所确定的领域范围，从适应度差的个体中随机抽取个体需要满足如（13）条件：

$$\sqrt{\sum_{i=1}^{D}(x_{ij} - g_{wj})^2} \leqslant a * Range \tag{13}$$

式中，a 表示为一个范围系数，只有满足公式（13）的种群个体被才能被重新初始化。

3　云计算流量预测模型构建

云计算的资源负载预测可以通过 SVM 来建立预测模型，在模型中使用非线性映射函数 $\varphi(x)$ 将云计算中的短时间中的负载数列 $x_1, x_2, \cdots x_n$ 作为样本从低维空间投影到高维的特征空间中，并进行线性回归，SVM 在高维特征空间中的回归函数为：

$$f(x) = \omega \times \varphi(x) + b \tag{14}$$

ω 为权向量，b 为偏置向量。根据风险最小化原则，公式（14）转换为如下优化问题表达：

$$\min J = \frac{1}{2}\parallel \omega \parallel^2 + C\sum_{i=1}^{n}(\xi_i^* + \xi_i)$$

$$s.t \begin{cases} y_i - \omega \times \varphi(x) - b \leqslant \varepsilon + \xi_i \\ \omega \times \varphi(x) + b - y_i \leqslant \varepsilon + \xi_i^* \\ \xi_i^*, \xi_i \geqslant 0, i = 1,2,\cdots,n \end{cases} \tag{15}$$

式中，$\parallel \omega \parallel$ 是与函数 f 具有相关复杂度相关的项，ε 表示不敏感损失稀疏，ζ_i^*，ζ_i 和 C 表示松弛因子和惩罚因子。为了将问题（15）转换为凸二次优化问题，特引入朗格朗日乘子：

$$L(\omega, b, \xi_i, \xi_i^*, \alpha, \alpha^*, \gamma, \gamma^*) =$$

$$\frac{1}{2}\parallel \omega \parallel + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) - \sum_{i=1}^{n}\alpha_i(\xi_i + \varepsilon - y_i + f(x_i)) -$$

$$\sum_{i=1}^{n}\alpha_i^*(\xi_i^* + \varepsilon - y_i + f(x_i)) - \sum_{i=1}^{n}(\xi_i\gamma_i - \xi_i^*\gamma_i^*) \tag{16}$$

式中，α_i 和 α_i^* 表示拉格朗日乘子，γ_i 表示损失因子，将

公式（16）转换为对偶形式，如下：

$$\omega(\alpha,\alpha^*)=-\frac{1}{2}\sum_{i,j=1}^{n}(\alpha_i-\alpha_i^*)\times(\alpha_j-\alpha_j^*)(\varphi(x_i),\varphi(x_j))$$

$$\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)y_i-\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)$$

$$s.t\begin{cases}\omega=\sum\limits_{i,j}^{n}(\alpha_i-\alpha_i^*)x_i\\ \sum\limits_{i=1}^{n}(\alpha_i-\alpha_i^*)=0\\ 0\leqslant\alpha_i,\alpha_i^*\leqslant C\end{cases}\tag{17}$$

因此，SVM 回归函数为：

$$f(x)=\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)(\varphi(x_i),\varphi(x))+b\tag{18}$$

为了简化（18），将使用核函数 $K(x_i,x)$ 代替 $(\varphi(x_i),\varphi(x))$，因此 SVM 函数如下：

$$f(x)=\sum_{i=1}^{n}(\alpha_i-\alpha_i^*)K(x_i,x)+b$$

$$s.t.K(x_i,x)=\exp\left(-\frac{\|x_i-x_j\|^2}{2\sigma^2}\right)\tag{19}$$

从以上式中发现，建立基于 SVM 的云计算负载预测模型的目的就是为了找到最优支持向量参数 σ。即也就是云计算负载预测值。

4 构建 IABC－SVM 预测模型

步骤 1：输入云计算资源负载序列，产生 SVM 的训练集和验证集。

步骤 2：设定 SVM 的核函数参数 σ 的范围，并设置改进的 ABC 算法参数。

步骤 3：设置 ABC 算法的中蜜源对应每一个支持向量参数 σ。

步骤 4：SVM 通过初始的参数 σ 对训练集进行学习。

步骤 5：使用改进的方法对 ABC 算法中三类蜜蜂的初始解和相关搜索进行操作。

步骤 6：当迭代次数超过最大迭代次数的时候，训练结束，输出种群最优位置，即为 SVM 中的向量参数 σ 最优值，否则转向步骤 5 继续执行。

步骤 7：采用向量参数 σ 来建立预测模型 $f(x)$，对验证集合的预测结果进行分析。

5 实验与分析

5.1 数据来源

为了能够有效的说明本文算法在云计算资源负载中发挥的作用，选择 Clarknet[18] 的网络负载作为本文实验的研究对象，在时间的设定上选择连续 7 天的 100 个历史数据作为训练数据，数据间隔为 4 个小时，预测未来 2 天的的云计算下的负荷数据。本文使用平均绝对误差 MAE 来计算预测模型的有效性，公式如下，其中 y_t 为实际负载数据，$\hat{y}_t$ 为预测数据，T 为预测时间序列，如公式（20）所示，但由于 SVM 对于［0，1］之间的数据具有非常高的敏感度，因此需要对网络流量数据进行归一化处理，公式（21）所示。

$$MAE=\frac{1}{T}\sum_{t=1}^{t=T}|y_t-\hat{y}_t|\tag{20}$$

$$x'=\frac{x-\min(x)}{\max(x)-\min(x)}\tag{21}$$

式（21）中，x 表示原始网络流量，$\max(x)$ 和 $\min(x)$ 表示最大值和最小值。

5.2 仿真结果分析

5.2.1 本文算法的预测效果

本文选取了连续 9 天的云计算的网络流量简化结果，单位为 Gb/S，对应结果如表 1 所示。表 2 为采用本文算法来预测得到第 8～9 天的云计算下流量预测结果。从表 2 中发现，根据相同的时间序号下的云计算的云计算网络流量预测与表 1 种的网络流量的仿真结果的误差基本上在 5％之内，图 2 显示了 Clarknet 平台中的实际负载和预测负载的效果，图中的两条线大部分都重合或者十分相近，从以上的分析中可以说明本文算法具有良好的预测效果。

表 1 连续 9 天的云计算中网络流量仿真实际结果

时间序号	网络流量	时间序号	网络流量	时间序号	网络流量
1	2.8972	12	7.8932	22	7.6731
2	5.5920	13	2.3894	23	8.3291
3	13.9822	14	1.8923	24	5.6732
4	15.8921	15	9.0823	25	9.912
5	10.2871	16	3.8932	26	8.2389
6	7.9832	17	4.3902	27	7.2891
7	9.9023	18	6.3902	28	8.2903
9	15.8922	19	4.9871	29	8.2392
10	8.9230	20	9.3782	31	9.2891
11	5.9832	21	4.8921	31	5.9832

表 2 第 8－9 天的云计算中的网络流量仿真结果

时间序号	网络流量	时间序号	网络流量	时间序号	网络流量
1	2.9172	12	7.9132	22	7.2931
2	5.6120	13	2.4194	23	7.9391
3	12.1724	14	1.9133	24	5.7332
4	14.2931	15	8.8821	25	9.732
5	11.0821	16	3.9132	26	7.9319
6	8.1831	17	4.5102	27	7.1191
7	9.8027	18	6.4202	28	8.1203
9	16.1923	19	5.2881	29	8.1272
10	9.1431	20	9.7182	31	9.3121
11	6.1332	21	4.9121	31	6.1832

5.2.2 Clarknet 平台下的几种预测方法的研究

在 Clarknet 平台中将本文算法、SVM 方法，IABC－LSSVM[19] 在 CPU 负载和网络负载的条件下进行对比，对比结果如 3～4 所示。

从图 4 得到本文提出的算法在不同的云计算 CPU 负载

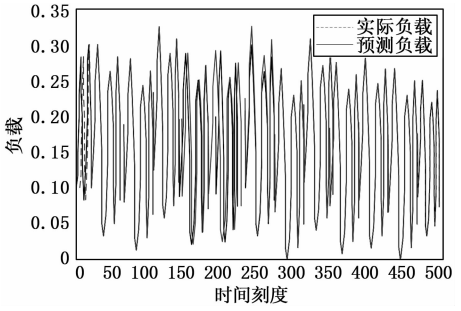

图 2　Clarknet 平台的网络负载预测结果

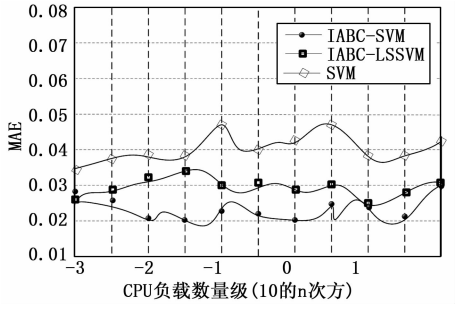

图 3　4 种算法下的 CPU 负载的 MAE 值比较

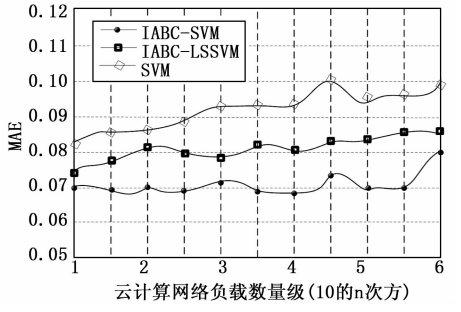

图 4　4 种算法下的云中网络负载的 MAE 值比较

条件下相比于其他两种预测方法明显具有良好的效果，预测精度更高，预测误差小，这说明本文预测方法能够适应 CPU 在不同负载变换下的预测结果。图 5 中发现本文算法的曲线相对于其他两种算法曲线的波动性小，相对平缓，这说明本文算法能够适应不同条件下的网络负载预测。从以上 2 个实验中发现本文算法能够有效的提高蜂群种群在解空间中的搜索能力，提高全局搜索效率，从而使得基于 IABC 的预测具有更好的效果。

6　结束语

针对云计算中的资源负载的预测效果，本文提出了云计算中的基于 ABC 优化的 SVM 预测模型。仿真实验说明本文算法能够在 CPU 负载和网络负载中具有比较好的预测精度，能够为云计算下的资源预测提供了一种参考。

参考文献：

[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] Reiss C，Tumanov A，Ganger G R，et al. Towards understanding heterogeneous clouds at scale：google trace analysis [A]. Proceedings of the 3rd ACM Symp on Cloud Computing [C]. 2012.

[3] Caron E，Desprez F，Muresan A. Forecasting for grid and cloud computing on－demand resources based on pattern matching [A]. Proceedings of the 2nd IEEE Int Conf on Cloud Computing Technology and Science [C]. 2010：456－463.

[4] Islam S. Empirical prediction models for adaptive resource provisioning in the cloud [J]. Future Generation Computer Systems，2012，28 (1)：155－162.

[5] Prevost J J. Prediction of cloud data center networks loads using stochastic and neural models [A]. Proceedings of the 6th International Conference on System of SystemsEngineering (SoSE) [C]. 2011：276－281.

[6] Khan A. Workload characterization and prediction in the cloud：A multiple time series approach [A]. Proceedings of the 3rd IEEE International Workshop on Cloud Management [C]. 2012：1287－1294.

[7] 李达港. 基于时间序列的 Openstack 云计算平台负载预测与弹性资源调度的研究 [J]，重庆邮电大学学报 (自然科学版)，2016，28 (4)：560－566.

[8] 聂清彬. 面向用户任务与云资源匹配度的调度算法研究 [J]. 微电子学与计算机，2016，33 (7)：93－97.

[9] 马华伟. 云计算环境下虚拟机负载均衡问题研究 [J]. 微电子学与计算机，2014，31 (4)：10－12.

[10] 海　江. 一种改进云计算虚拟资源负载均衡调度方案 [J]，控制工程，2017，24 (1)：100－105.

[11] 匡珍春. 基于猫群优化算法的云计算虚拟机资源负载均衡调度 [J]. 吉林大学学报 (理学版)，2016，54 (5)：1117－1122.

[12] 何丹丹. 云环境下基于节能和负载均衡的混沌粒子群资源优化调度 [J]. 计算机测量与控制，2014，22 (5)：1626－1628.

[13] 刘晓艳. 基于改进云模型的云计算负载预测 [J]. 计算机应用研究，2015，32 (10)：3124－3127.

[14] 孙兰芳，张曦煌. 基于蜜蜂采蜜机理的云计算负载均衡策略 [J]，计算机应用研究，2016，33 (4)：1179－1182.

[15] 徐爱萍. 实时多任务异构云计算平台负载均衡算法 [J]. 中国科技大学学报，2016，10 (3)：215－221.

[16] 陈榕利. 云计算环境下差异化资源的合理调度模型改进 [J]. 现代电子技术，2017，40 (12)：152－154.

[17] 马安香. 基于深度置信网络的云应用负载预测方法 [J]，东北大学学报 (自然科学版)，2017，38 (2)：209－213.

[18] Traces in the Internet traffic archive [EB/OL]. [2015－08－30]. http：//ita. ee. lbl. gov/html/traces. html.

[19] 许爱军. 改进 ABC 算法优化 LSSVM 的网络流量预测模型 [J]，计算机应用与软件，2015，32 (1)：323－326.