Journal of System Simulation

Volume 31 | Issue 11 Article 16

12-13-2019

A Cloud Service Composition Optimization Based on HNN

Huili Zhang
1. Linfen Vocational and Technical College, Linfen 041000, China; ;2. School of Educational Science, Shanxi Normal University, Linfen 041000, China;

Zhihe Li
2. School of Educational Science, Shanxi Normal University, Linfen 041000, China;

A Cloud Service Composition Optimization Based on HNN

Abstract

Abstract: With the rapid development of Cloud service application, how to effectively optimize the composition of Cloud services on cloud platform and improve the overall performance of cloud platform system have become an urgent research issue. In order to improve the efficiency of Cloud services, *a combined optimization model based on Hopfield neural network is proposed*. The problem of Cloud services is modeled. *The problem is expressed as Hopfield Neural Network energy model for optimization, and a PSO group algorithm with Cauchy disturbance is designed to improve the Hopfield model.* The experimental comparison shows that the method can improve the efficiency of Cloud service composition optimization more effectively than other typical algorithms.

Keywords

Hopfield neural network, Composition optimization, Web service, Resource constraints, Load balancing, Cloud computing

Recommended Citation

Zhang Huili, Li Zhihe. A Cloud Service Composition Optimization Based on HNN[J]. Journal of System Simulation, 2019, 31(11): 2335-2343.

第 31 卷第 11 期 系统仿真学报© Vol. 31 No. 11
2019 年 11 月 Journal of System Simulation Nov., 2019

一种基于 HNN 的云服务组合优化

张会丽[1,2]，李志河[2*]
(1. 临汾职业技术学院，山西 临汾 041000；2. 山西师范大学教育科学学院，山西 临汾 041000)

摘要：随着云服务应用开发的日新月异，如何有效地在云平台上实现优化服务的组合，提升云平台系统的整体性能是一个亟待解决的研究问题。为提升云服务的效率，*提出一种基于霍普菲尔德神经网络的组合优化模型。*该方法针对云服务问题建模；*设计一种带有柯西扰动技术的 PSO 算法来改进霍普菲尔德模型，将该云服务问题表达为霍普菲尔德神经网络能量模型进行优化。*通过实验比较证明，该方法比其它典型算法可以更加有效地提升云服务组合优化执行的效率。

关键词：霍普菲尔德神经网络；组合优化；Web 服务；资源约束；负载均衡；云计算
中图分类号：TP393；TP181 文献标识码：A 文章编号：1004-731X (2019) 11-2335-09
DOI: 10.16182/j.issn1004731x.joss.19-FZ0341

A Cloud Service Composition Optimization Based on HNN

Zhang Huili[1,2], *Li Zhihe*[2*]
(1. Linfen Vocational and Technical College, Linfen 041000, China;
2. School of Educational Science, Shanxi Normal University, Linfen 041000, China)

Abstract: With the rapid development of Cloud service application, how to effectively optimize the composition of Cloud services on cloud platform and improve the overall performance of cloud platform system have become an urgent research issue. In order to improve the efficiency of Cloud services, *a combined optimization model based on Hopfield neural network is proposed*. The problem of Cloud services is modeled. *The problem is expressed as Hopfield Neural Network energy model for optimization, and a PSO group algorithm with Cauchy disturbance is designed to improve the Hopfield model.* The experimental comparison shows that the method can improve the efficiency of Cloud service composition optimization more effectively than other typical algorithms.

Keywords: Hopfield neural network; Composition optimization; Web service; Resource constraints; Load balancing; Cloud computing

引言

Cloud服务的组合优化是一个典型的NP问题，随着 Cloud 服务的广泛开发和使用，如何提高服务的整体执行效率已经成为一个研究热点。因为运行过程中的 Cloud 服务的状态往往是随机的，且无法事先预知的，所以服务的质量也是无法确定的。如何保证服务群具备运行过程中的质量，又可以提升执行的效率将是服务调度算法发展的契机和创新点[1-2]。

Dahan F 等为提升 Web 服务的优化组合，提出一种增强型飞行蚁群优化算法。该方法中的飞行蚂蚁不仅在其路径上的节点上注入信息素，而且在相邻的节点上注入信息素，增加了它们在未来迭代

收稿日期：2019-05-17 修回日期：2019-07-18；
基金项目：国家社会科学基金(BIA180202)，教育部信息化教学研究课题(2018LXB0179)；
作者简介：张会丽(1974-)，女，霍州，硕士，副教授，研究方向为人工智能；李志河(通讯作者 1974-)，男，宁夏，博士，教授，研究方向为大数据。

中被探索的机会，增强了空间优化能力来实现组合优化[3]。Shahrokh P 等提出一种改进的遗传算法实现基于 QoS 的 Web 服务组合优化。该方法将启发式和遗传算法结合，提出一种半启发式遗传算法。这种启发式方法根据不满足的约束改变染色体，增强算法的性能[4]。WANG Peng-weid 等认为以往的自动服务组合方法主要依赖于序列结构，使得在服务组合过程中难以考虑不确定性的影响。所以采用并扩展了图形局域网技术来解决这一问题，实现服务工作流的过程优化[5]。LIU Zhi-zhong 等提出了一种特定的 SLO 来解决 QoS 感知云服务组合问题。SLO 是一种优化算法模型，它可以通过体现 SLO 的 3 个演化空间(微空间、学习空间和信仰空间)生成具体的算法[6]。

Wang H 等将用户的信用和偏好集成到 Web 的服务组合，以提升服务的全面性[7]。Wu Y 等研究人员，利用键有效地消除常用的反向索引的冗余，提出一种大型服务库中加速 Web 服务发现和组合的多级索引模型[8]。Rodriguez-Mier P 提出了一个集成语义的 Web 服务发现和组合框架，可以以图形化的方式方便用户发现和组合服务[9]。

韩敏等通过可信任评价模型融合到 Web 服务框架中，建立了一种新的基于 QoS 感知服务组合优化方法[10]。文献[11]通过重新定义离散粒子群中的操作算子和适应度函数，设计出一种改进的粒子群优化算法实现 Web 服务的组合优化。张以文等人，通过增强算法的高斯变异概率，提出一种基于烟花算法的 Web 服务组合优化模型[12]。郭星等通过引入烟花爆炸机制和粒子反向学习理论，增强种群粒子的搜索能力来改进烟花算法实现 Web 服务的组合优化[13]。Huo Y 等在服务组合模型中加入时间衰减函数，将服务组合形式化为非线性整数规划问题。该文还提出了离散的 g_{best} 引导人工蜂群算法，通过对蜜蜂的食物搜索，模拟了对最优服务组合解的搜索，使用该改进的蜂群算法实现云服务的组合优化[14]。

此外，还有叶恒舟等通过建立候选服务集，减少搜素空间，并用遗传算法来优化服务的组合[15]。倪志伟等通过增强蚁群算法过程中信息素的有效积累，提出了一种改进的蚁群算法优化服务的组合方法[16]。

陆湘鹏和叶恒舟通过在初始例子群中增加局部优选的粒子策略，并调整认知系统和社会系数，加大种群的多样性，提出一种可以在早期进行多样性修补的粒子群算法来实现 Web 服务的组合[17]。文献[18]中，王亮和郭星通过柯西变异代替高斯变异来改进烟花算法，实现优化算法的提升。吴黎兵等为避免粒子群算法过度早熟并尽力发现 Pareto 前端的组合服务，提出了一种受群体多样性影响的速度更新方法和惯性权重模型，改善了基于粒子群的 Web 服务优化组合算法[19]。此外，还有尹浩等将粒子变异策略来抑制群体的早熟收敛，并增强群体的全局搜索能力的新方法，提出一种基于约束支配关系的局部搜索策略并将其结合到 MDPSO 算法，并成功应用到服务组合设计中[20]。文献[21]中，温涛等通过采用区间数的形式描述用户的需求得到 QoS 指标，提出一种带有动态边界的改进粒子群算法，搜寻满足 QoS 要求的服务组合方案。

在组合优化算法当中，除群体智能算法外，神经网络也是一只独秀，其中霍普菲尔德神经网络将描述的数学问题转变成为求解一种网络系统的稳定状态的方法，得到了广泛的应用。如车间作业的优化生产调度[22]、多处理机作业调度[23]和变电站负载自动均衡的设计[24]等。

一般的服务组合优化方法只是通过某种算法尽可能的满足服务质量的需求而进行调度的优化，并没有考虑实际系统资源的实际状况。因此，这些方法在实际应用中既不能发挥资源的优势，也不能提升系统的执行效率。

云服务组合优化其本质就是在有限的资源基础上实现满足用户 QoS 的任务调度。本文将建立一种带粒子群优化的离散霍普菲尔德神经网络模型(Hopfield Neural Network，HNN)来实现云服务

https://dc-china-simulation.researchcommons.org/journal/vol31/iss11/16
DOI: 10.16182/j.issn1004731x.joss.19-FZ0341

的组合优化，并综合考虑系统资源实时情况进一步提升云服务并行执行的效率。

1 云服务问题建模

1.1 云系统资源描述

经过对文献[23-24]的研究，这里在实验中将采用平均处理器使用率、平均内存使用率、平均 I/O 使用率和平均网络使用率 4 个指标作为计算节点分配的依据，如表 1 所示。

表 1 系统资源特征
Tab. 1 System resources characteristics /%

指标	分类
平均处理器使用率	任务
平均内存使用率	任务
平均 I/O 使用率	任务
平均网络使用率	任务

1.2 Web 服务问题描述

霍普菲尔德 Neural Network 求解问题的实质是将组合优化问题转化为对应的能量函数的极值过程。这里首先构造目标函数和约束条件，然后再将其转换为对应的能量函数[25]。

定义 1 一个任务由 n 个子任务构成，如公式(1)所示：

$$T=\{T_1,T_1,\cdots,T_n\} \tag{1}$$

定义 2 每个子任务可以由 q 个服务构成可以候选的集合 W，如公式(2)所示：

$$T_k=\{W_1^k,W_2^k,\cdots,W_q^k\} \tag{2}$$

每个任务和服务的关系描述如图 1 所示。

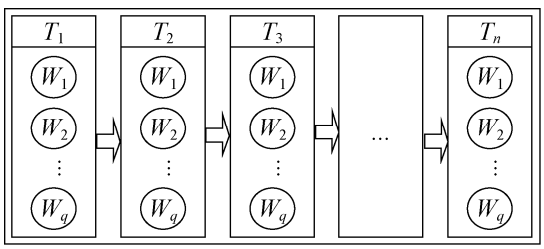

图 1 任务和服务
Fig. 1 Task & Service

定义 3 每个任务包含的子任务存在一个先后逻辑关系，可以按关系 R 表示，如式(3)所示：

$$R=\{<T_1,T_2>,<T_2,T_5>,<T_5,T_3>,\cdots,<T_i,T_n>\} \tag{3}$$

根据关系 R 定义每个子任务的依赖关系可以如图 2 所示。

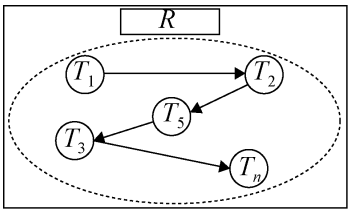

图 2 任务依赖关系描述
Fig. 2 Description of task dependence relationship

定义 4 Web 服务质量属性：是指该服务提供的非功能性属性，包括：性能、响应时间、安全性和费用等，定义如公式(4)所示：

$$Q^{\tau}=\left\{Q_1^{\tau},Q_2^{\tau},\cdots,Q_m^{\tau}\right\} \tag{4}$$

式中：Q^{τ} 为一个 QoS 属性集合，共有 m 个属性元素；Q_k^{τ} 为第 τ 个服务的第 k 个 QoS 属性。为了简化每个属性，这里将每个属性归一化处理到{0-1}空间中。

假设系统中可以提供的服务集合为 $W^t=\{W_1^t,W_2^t,\cdots,W_q^t\}$，$W_{\mathrm{q}}^t$ 表示第 t 个子任务的第 q 个服务。每个服务具备 m 个 QoS 属性，记为：$Q^{\mathrm{k}}=\{Q_1^k,Q_2^k,\cdots,Q_{\mathrm{m}}^k\}$，其中 Q_i^k 表示第 k 个服务的第 i 个质量属性，该属性可以是响应时间、安全性、价格、吞吐量、可靠性和依赖性等指标。其中依赖性表示两个服务之间存在相互调用或成员访问的关系。

定义 5 设每个用户对每个服务的 QoS 的质量属性的要求为 $P=\{P_1,P_2,\cdots,P_{\mathrm{m}}\}$，每个子任务选取的最佳配置服务的函数是公式(5)，该式描述了第 t 个任务中用户质量要求最近的服务。

$$\mathrm{g}^{\mathrm{t}}(W^t)=\min((Q_j^k-P_j)^2) \tag{5}$$

定义 6 (目标函数 1)适应度函数 f 被定义为公式(6)。

$$f_1(T)=\sum_{i=1}^{n}g^t(W^t) \tag{6}$$

定义 7 子任务依赖矩阵 $\boldsymbol{R^t}=\{r_{ij}\}$，若 r_{ij}=1 表示存在子任务 i 对子任务 j 有直接依赖性，否则没

有直接依赖性，如公式(7)表示：

$$\boldsymbol{R}^t=\begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{pmatrix}_{n\times n} \tag{7}$$

设状态变量 E_{ijk} 其值为 0 或 1，若 E_{ijk}=1 表示在第 k 个周期时，子任务 i 运行在第 j 个节点上。其中，i 的范围为 1 到 N；j 的范围为 1 到 M；k 的范围是 1 到 T。

约束 1 一个计算节点上在任意周期 k 内至多可以运行一个子任务，记为：

$$T_1=\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{T}\sum_{\substack{l=1\\ l\neq i}}^{N}E_{ijk}\times E_{ljk}=0 \tag{8}$$

约束 2 限制一个子任务只能在单个计算节点上行执行，记为：

$$T_2=\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{T}\sum_{\substack{h=1\\ h\neq i}}^{N}\sum_{l=1}^{T}E_{ijk}\cdot E_{ihl}=0 \tag{9}$$

约束 3 执行期内所有的节点没有空闲周期，可以表示为：

$$T_3=\sum_{j=1}^{M}\sum_{k=1}^{T}\left(\sum_{i=1}^{N}E_{ijk}-1\right)^2=0 \tag{10}$$

约束 4 单个子任务执行期限为：

$$\begin{aligned} &T_4=\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{T}U_{ijk}\times S_{ijk}^2H(\mathrm{S}_{ijk}),\\ &S_{ijk}=k-p_i,\quad \text{and}\\ &H(\mathrm{S}_{ijk})=\begin{cases}0 & S_{ijk}>0\\ 1 & S_{ijk}\le 0\end{cases}\end{aligned} \tag{11}$$

式(11)中，p_i 是子任务 i 的最长执行周期限制。函数 $H(S_{ijk})$是一个分段函数。该分段函数返回 0 或 1 的值。当子任务 i 的周期大于 p_i 时，该值返回 0；否则还会 1。

约束 5 子任务依赖，表示任意两个子任务之间存在依赖关系，则执行前后相互制约，不可竞争执行，记为：

$$\begin{aligned} T_5=&\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{T}\sum_{\substack{h=1\\ h\neq i}}^{N}\sum_{\substack{l=1\\ l\neq j}}^{M}\sum_{t=1}^{N}E_{ijk}\times r_{ih}^t\times E_{hlk}=\\ &\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{k=1}^{T}\sum_{\substack{h=1\\ h\neq i}}^{N}\sum_{\substack{l=1\\ l\neq j}}^{M}E_{ijk}\times r_{ih}^s\times E_{hlk}=0\end{aligned} \tag{12}$$

式中：变量 t 表示依赖矩阵的阶数；$\boldsymbol{R}^t$ 代表第 t 阶依赖矩阵；元素 r^t_{ih} 表示依赖关系中第 i 个子任务对第 h 个子任务存在的依赖状况。该式可以限制 2 个互相有依赖的服务不会同时执行，且依赖服务的顺序。

定义 6 (目标函数 2) 限制每个子任务的运行时间需求为 P_i 个周期，所有服务的执行时间趋近于各自的需要执行的时间 P_i，整体运行目标函数记为：

$$f_2=\min\left(\sum_{i=1}^{N}\left(\sum_{j=1}^{M}\sum_{k=1}^{T}E_{ijk}-P_i\right)^2\right) \tag{13}$$

式中：第 i 个子任务的执行期望为 P_i。该式定义执行的目标函数满足执行时间最短。

当系统达到上述能量项时，各个能量项趋近于 0。联合以上 5 个条件和 2 个目标函数可以构造整个系统的能量函数，如式(14)所示。其中，B_y 是权重因子。

$$\begin{aligned} &B_y>0;\\ &E=\frac{1}{2}\left(\sum_{y=1}^{5}B_yT_y+f_1+f_2\right)\end{aligned} \tag{14}$$

1.3 能量函数的设计

通过设计的能量函数来求极小值实现多条件的组合优化。可以证明该式(14)满足 Lyapunov 控制系统稳定性条件，即系统存在稳定状态，该能量函数有解[25]。整理式(14)可以得到式(15)。

$$\begin{aligned} E=&-\frac{1}{2}\sum_{i}\sum_{j}\sum_{k}\sum_{x}\sum_{y}\sum_{z}E_{ijk}\times\phi_{ijkxyz}\times E_{xyz}+\\ &\sum_{x}\sum_{y}\sum_{z}\eta_{xyz}\times E_{xyz}\\ \phi_{ijkxyz}=&-T_1(1-\varphi(i,x))\varphi(j,y)\varphi(k,z)-\\ &T_2\varphi(i,x)(1-\varphi(j,y)-T_3\varphi(i,x)-T_4\varphi(j,x)-\\ &T_6(1-\varphi(i,x)(1-\varphi(j,x))\varphi(k,z)r_{ih}^s\\ \eta_{xyz}=&-T_3F_m-T_4+T_5S_{xyz}^2H(S_{xyz})\\ \varphi(u,v)=&\begin{cases}0 & \text{if}\quad u-v\neq 0\\ 1 & \text{if}\quad u-v=0\end{cases}\end{aligned} \tag{15}$$

式中：φ_{ijkxyz} 为神经元之间的连接强度；变量 η_{xyz} 为神经元阈值。这里采用二元变量的离散型神经元

表示系统的状态，该二元变量可由式(16)推导得：

$$E_{xyz}^{t+1}=\begin{cases}0 & \text{if} \quad S_{xyz}<0\\ 1 & \text{if} \quad S_{xyz}>0,\\ E_{xyz}^{t} & \text{if} \quad S_{xyz}=0\end{cases} \tag{16}$$

$$S_{xyz}=\sum_{x}\sum_{y}\sum_{z}\varphi_{ijkxyz}\times E_{ijk}-\eta_{xyz}$$

2 带柯西扰动的粒子群算法

2.1 粒子群优化算法(PSO)

在该算法中，每只捕食过程中的鸟被抽象成一个具有独立位置和运动速度的粒子。同时，每只鸟被所在的群体所影响。此外，该群体也同样具有统一的位置和运动速度。该运动速度可以被视作是该群体所有“粒子”速度的集成。同时，群体中存在一个“领袖”，它会影响者整体的运动方向。通过该“捕食”的群体行为过程，该鸟群将获得问题空间中的优化解。

在系统初始开始时，需要产生一群代表鸟群的粒子。在“捕食”的运动过程中，每个粒子将趋向两个目标。一个是整体粒子群的最佳位置，即群体的最佳解 p_g；另一个是每个独立粒子的最佳位置，记为 p_i。在整个算法的迭代过程中，每个粒子将按照公式(17)去更行自己的位置和运动参数。

$$\begin{aligned}v_{i,j}^{k}&=\rho\times v_{i,j}^{k-1}+\delta_1\beta_1(p_{i,j}^{k-1}-x_{i,j}^{k-1})+\\&\quad\delta_2\beta_2(p_{g,j}^{k-1}-x_{i,j}^{k-1})\\x_{i,j}^{k}&=x_{i,j}^{k-1}+\gamma\times v_{i,j}^{k}\end{aligned} \tag{17}$$

式中：因数 ρ 记录了先前运动的能力。变量 x_k 和 v_k 分别代表了第 k 次迭代过程中的位置和运动速度。此外，δ_1 和 δ_2 是 2 个 0 到 1 之间的随机数。β_1 和 β_2 是 2 个需要人工设置的加速因子。γ 是控制运动速度的权重参数。

2.2 柯西扰动

基础的粒子群算法提供了一种群体智能的优秀方法去解决搜索离散空间的解。但是该算法在搜索问题的过程中很容易落入局部空间的最优解，而不是整个问题空间中的最优解。为了避免此类情况，我们将通过柯西扰动技术来改进基本的粒子群算法(PSO)。

$$\begin{aligned}v_{i,j}^{k}&=w\times v_{i,j}^{k-1}+c_1\beta_1(p_{i,j}^{k-1}+\beta_3 g^{k}-x_{i,j}^{k-1})+\\&\quad c_2\beta_2(p_{g,j}^{k-1}-x_{i,j}^{k-1})\end{aligned} \tag{18}$$

$$x_{i,j}^{k}=x_{i,j}^{k-1}+\gamma\cdot v_{i,j}^{k}$$

$$g^{k}=\beta_4\text{cauchy}(\cdot) \tag{19}$$

柯西分布具有比高斯分布更好的扰动能力[18]。使用柯西函数 cauchy(·) 更新个体粒子的运动速度。在公式(18)中针对粒子的运动速度加入公式(19)的所定义的柯西扰动，其中 β_4 是柯西公式调整系数，β_3 是柯西扰动速度比例因数。

2.3 适应度函数

在带柯西扰动的粒子群算法迭代过程中，将采用公式(20)为最佳适应度评价函数 $f(\cdot)$。其中，$y_i(\cdot)$ 是该算法需要满足的目标，且 $y_i'(\cdot)$ 是计算值。$f(\cdot)$ 被用来评价整个模型的配置，该值越小，模型质量越高。当该函数满足一定的精度时，可以作为算法迭代过程的结束。

$$f(\cdot)=\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i(\cdot)-y_i'(\cdot))^{2}} \tag{20}$$

2.4 算法步骤

该算法的计算步骤如图 3 所示。

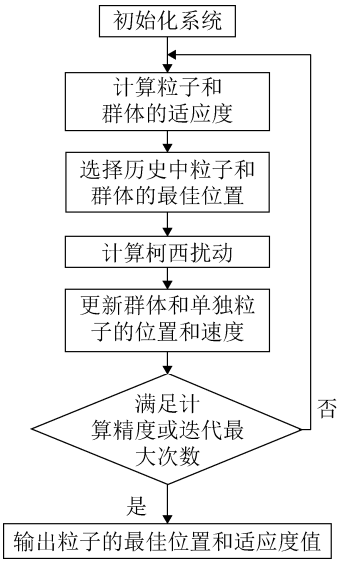

图 3 算法步骤描述

Fig. 3 Description of algorithm step

step 1：设置系统参数，并初始化种子。

step 2：通过适应度函数评价每个单独粒子的适应度。

step 3：选择迭代过程中每个粒子的最佳位置和群体的最佳位置。

step 4：生成柯西扰动。

step 5：更新速度和位置。

step 6：判断是否满足结束条件。如果满足，则输出粒子的最佳位置和适应度值；否则，回到 step 2 继续执行。

3 实验与分析

3.1 实验参数与环境

为了验证本算法在云服务组合优化方面的优势，该实验搭建在 5 台计算节点上完成。这 5 台计算机的具有相同的配置(i3 处理器 CPU3.1Ghz，Memory 8GB，操作系统 linux，开发环境 Pycharm 2017，开发语言 Python3.7)。

该实验从 WSDream 数据集[26]中抽取 1 500 个测试用例，并通过实验与算法烟花算法、粒子群算法和蚁群算法 3 个典型算法进行比较。

3.2 实验参数

在本实验中需要给带有柯西扰动的粒子群算法配置参数去自动调优 HNN 的 5 个因数，并设置最大迭代次数为 500 次，误差精度 10^{-6}，其参数配置如表 2 所示。

表 2 带由柯西扰动的 PSO 参数设置
Tab. 2 Parameters for PSO with Cauchy disturbance

parameters	δ_1	δ_2	ρ	γ	β_1	β_2	β_3	β_4
values	4.3	4.1	3.5	2.4	3.1	3.3	1.3	5.7

3.3 结果分析

3.3.1 系统参数配置

如图 4 所示，通过使用柯西扰动的粒子群算法的迭代计算，在迭代到 267 次时，网络开始收敛到精度要求。该 HNN 模型中的 5 个参数，可以配置为 $B_1=3.7, B_2=3.5, B_3=1.7, B_4=2.7, B_5=3.9$。

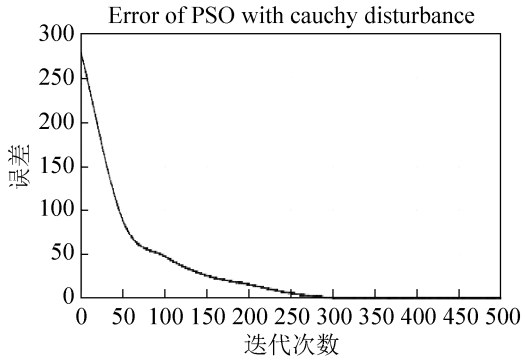

图 4 带柯西扰动的粒子群算法迭代过程
Fig. 4 Process of PSO Algorithm with Cauchy disturbance

3.3.2 算法性能分析

为了比较算法的承重负载情况，从服务集中抽取 1 500 个服务，并每次逐步增加 100 个服务进行试验。

从图 5 中，可以发现烟花算法在测试数量达到 1 000 个时，算法的执行时间开始有变化，再达到 1 300 个时，系统执行时间与测试数量变化开始显著变化。

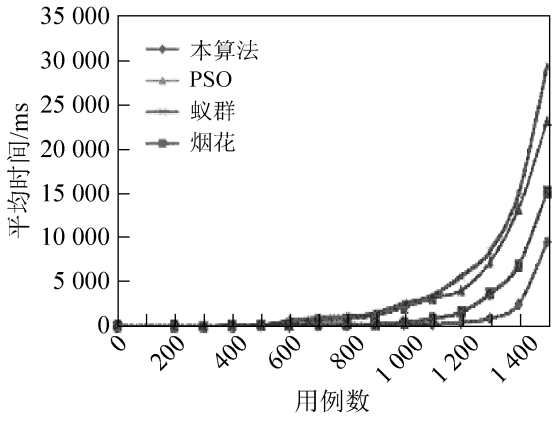

图 5 算法性能比较
Fig. 5 Performance comparison of algorithms

从该图中还可以发现，PSO 算法与蚁群算法的情况类似，当执行用例数达到 600 个时，执行时间均开始有变化。但是蚁群算法增加测试服务数量到 1 100 时，变化趋势明显，而 PSO 算法则滞后蚁群算法到 1 200 左右时，才呈现出该趋势。说明 PSO 算法和蚁群算法的执行效率和变化趋势类似。

如图 5 中所示，当系统负载数量小于 1 200 个时，本算法系统的运行时间小于 500 ms。当服务次数达到 1 200~1 400 个时，该算法执行时间与执行个数有明显变化变化。当超过 1 400 个时，算法执行时间有明显变化，说明 1 400 个左右时，系统的负载压力较大，此时系统对负载承接数量开始敏感。此外，可以进一步发现，本算法比较其它算法承载力更强，更满足云计算的弹性要求。

3.3.3 节点性能分析

通过图 6，可以观测到当系统运行到个服务数量增加到 1 500 个时，5 个计算节点的资源利用情况，包括：CPU、内存、IO 和网络 4 个性能指标的平均利用率。节点 1 是主节点负责服务任务的调度和算法的执行，剩余 4 个从节点负责计算任务。从图 6 中，可以观察到主节点 1 的负载比剩余的 4 个节点较大，各项指标都在 85%左右。剩余 4 个节点的 4 个指标基本类似，说明该系统可以很好地均匀分配任务和服务，系统负载均衡。

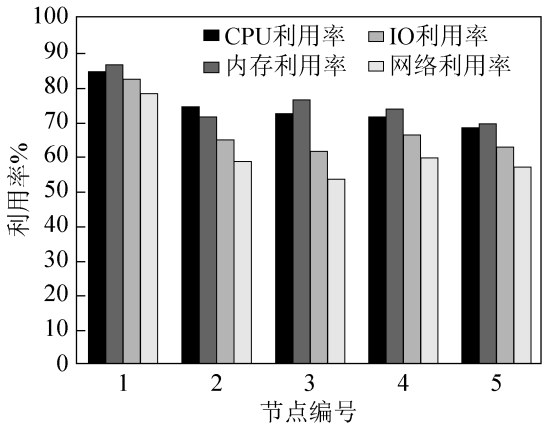

图 6 节点性能
Fig. 6 Node performance

经过上述观察，可以得出本算法无论在服务承载数量还是系统负载均衡方面都比烟花算法、PSO 算法、蚁群算法具有较好的优势。

4 结论

一般的服务组合设计算法仅仅是改变实现的方法，其基本思想都是尽力满足用户服务质量的要求而已。本文提出一种基于改进的霍普菲尔德神经网络的云服务组合模型，不仅满足了用户对服务的 QoS 需求，而且考虑了系统实际运行资源的分配问题，并且通过带柯西扰动的粒子群算法优化霍普菲尔德神经网络参数，使其可以自动按用户需求和按资源调度任务，最终实现系统整体负载平衡，达到系统整体性能的提升。

在实验中通过与 3 个典型算法的在性能、服务满意度和性能指标上进行比较，充分证明了该方法的有效性。此外，通过实验我们也发现：任务的大小和服务的粒度可能影响系统的执行效率，如何划分任务和服务的颗粒，将是我们新的研究方向。

参考文献：

[1] 丁志军, 周泽霞. Web 服务组合测试综述[J]. 软件学报, 2018, 29(2): 299-319.
Ding Zhijun, Zhou Zexia. Survey on Web Service Composition Testing[J]. Journal of Software, 2018, 29(2): 299-319.

[2] Jatoth C, Gangadharan G R, Buyya R. Computational Intelligence based QoS-aware Web Service Composition: A Systematic Literature Review[J]. IEEE Transactions on Services Computing (S1939-1374), 2017, 10(3): 475-492.

[3] Dahan F, Hindi K E, Ghoneim A, et al. An adapted ant-inspiredalgorithm for enhancing Web service composition[J]. International Journal on Semantic Web & Information Systems (S1552-6283), 2017, 13(4): 181-197.

[4] Shahrokh P, Safi-EsfahaniF. QoS-based Web service compositionapplying an Improved Genetic Algorithm (IGA) method[J]. International Journal of Enterprise Information Systems (S1548-1115), 2016, 12(3): 60-77.

[5] Wang P W, Ding Z J, Jiang C J, et al. Automatic web service composition based on uncertainty execution effects[J]. IEEE Transactions on Services Computing (S1939-1374), 2016, 9(4): 551-565.

[6] Liu Z Z, Chu D H, Song C, et al. Social learning optimization (SLO) algorithm paradigm and its application in QoS-aware cloud service composition[J]. Information Sciences (S0020-0255), 2016, 326: 315-333.

[7] Wang H, Zou B, Guo G, et al. Integrating Trust with User Preference for Effective Web Service

Composition[J]. IEEE Transactions on Services Computing(S1939-1374), 2017, 10(4): 574-588.

[8] Wu Y, Yan C, Ding Z, et al. A Multilevel Index Model to Expedite Web Service Discovery and Composition in Large-Scale Service Repositories[J]. IEEE Transactions on Services Computing (S1939-1374), 2016, 9(3): 330-342.

[9] Rodriguez-Mier P, Pedrinaci C, Lama M, et al. An Integrated Semantic Web Service Discovery and Composition Framework[J]. IEEE Transactions on Services Computing (S1939-1374), 2016, 9(4): 537-550.

[10] 韩敏, 段彦忠. 融合可信性评价的 Web 服务组合 QoS 优化[J/OL]. 控制与决策. [2019-08-20]. https://doi.org/10.13195/j.kzyjc.2019.0006.
Han Min, Duan Yanzhong. Quality of Service Composition Optimization for Web Services Fusion of Credibility Assessmen[J/OL]. Control and Decision . [2019-08-20]. https://doi. org/10.13195/j.kzyjc.2019. 0006.

[11] 叶恒舟, 陆湘鹏. 基于离散粒子群优化的鲁棒 Web 服务组合[J]. 电子科技大学学报, 2018, 47(3): 443-448.
Ye Hengzhou, Lu Xiangpeng. Robust Web Services Composition Based on Discrete Particle Swarm Optimization[J]. Journal of University of Electronic Science and Technology of China, 2018, 47(3): 443-448.

[12] 张以文, 吴金涛, 赵姝, 等. 基于改进烟花算法的 Web 服务组合优化[J]. 计算机集成制造系统, 2016, 22(2): 422-432.
Zhang Yiwen, Wu Jintao, Zhao Shu, et al. Optimization service composition based on improved firework algorithm[J]. Computer Integrated Manufacturing Systems, 2016, 22(2): 422-432.

[13] 郭星, 陈姗姗, 张以文, 等. 烟花粒子群优化算法在 Web 服务组合上的应用[J]. 小型微型计算机系统, 2018, 39(6): 1312-1316.
Guo Xing, Chen Shanshan, Zhang Yiwen, et al. Application of Fireworks Particle Swarm Optimization Algorithm in Web Service Composition [J]. Journal of Chinese Computer Systems, 2018, 39(6): 1312-1316.

[14] Huo Y, Zhuang Y, Gu J, et al. Discrete gbest-guided artificial beecolony algorithm for cloud service composition[J]. Applied Intelligence (S0924-669X), 2015, 42(4): 661-678.

[15] 叶恒舟, 关云慧. 基于局部选择和遗传算法的 QoS 感知的服务组合方法[J]. 小型微型计算机系统, 2016, 37(7): 1389-1392.
Ye Hengzhou, Guan Yunhui. QoS-aware Web Service Composition Based on Local Selection and Genetic Algorithm[J]. Journal of Chinese Computer Systems, 2016, 37(7): 1389-1392.

[16] 倪志伟, 方清华, 李蓉蓉, 等. 改进蚁群算法在基于服务质量的Web服务组合优化中的应用[J]. 计算机应用, 2015, 35(8): 2238-2243, 2279.
Nie Zhiwei, Fang Qinghua, Li Rongrong, et al. Improved ant colony optimization for QoS-based Web service composition optimization[J]. Journal of Computer Applications, 2015, 35(8): 2238-2243, 2279.

[17] 陆湘鹏, 叶恒舟. 基于改进粒子群优化算法的 Web 服务组合[J]. 桂林理工大学学报, 2017, 37(4): 699-706.
Lu Xiangpeng, Ye Hengzhou. Web services composition based on modified particle swarm optimization[J]. Journal of Guilin University of Technology, 2017, 37(4): 699-706.

[18] 王亮, 郭星. 基于柯西烟花算法的大规模服务组合优化[J]. 计算机工程与应用, 2018, 54(24): 34-40.
Wang Liang, Guo Xing. Large-scale service portfolio optimization based on Cauchy fireworks algorithm[J]. Computer Engineering and Applications, 2018, 54(24): 34-40.

[19] 吴黎兵, 杨科, 聂雷, 等. 基于改进粒子群算法的 Web 服务组合推优方法[J]. 华中科技大学学报(自然科学版), 2014, 42(10): 69-73.
Wu Libing, Yang Ke, Nie Lei, et al. Web service composition recommendation based on modified particle swarm optimization[J]. Journal of Huazhong University of Science and Technology (Nature Science Edition), 2014, 42(10): 69-73.

[20] 尹浩, 张长胜, 张斌, 等. 一种求解 SLA 等级感知服务组合问题的多目标离散粒子群优化算法[J]. 电子学报, 2014, 42(10): 1983-1990.
Yin Hao, Zhang Changsheng, Zhang Bin, et al. A Multi-Objective Discrete Particle Swarm Optimization Algorithm for SLA-Aware Service Composition Problem[J]. Acta Electronica Sinica, 2014, 42(10): 1983-1990.

[21] 温涛, 李迎秋, 盛国军, 等. 不确定信息下基于改进粒子群算法的 Web 服务选择[J]. 吉林大学学报(工学版), 2014, 44(1): 129-136.
Wen Tao, Li Yingqiu, Sheng Guojun, et al. Improved PSO-based Web service selection under uncertain information[J], Journal of Jilin University (Engineering and Technology Edition), 2014, 44(1): 129-136.

[22] 王万良, 吴启迪, 徐新黎. 基于 Hopfield 神经网络的作业车间生产调度方法[J]. 自动化学报, 2002, 28(5):

838-844.
Wang Wanliang, Wu Qidi, Xu Xinli. Hopfield Neural Network Approach for Job-Shop Scheduling Problems[J]. Acta Automatica Sinica, 2002, 28(5): 838-844.

[23] 王秀利，吴惕华. 一种求解多处理器作业调度的 Hopfield 神经网络方法[J]. 系统工程与电子技术, 2002, 24(8): 13-16.
Wang Xiuli, Wu Tihua. Scheduling Mutiprocessor Job Using Hopfield Neural Network[J]. Systems Engineering and Electronics, 2002, 24(8): 13-16.

[24] 邓科，李挺，蔡昂，等. 基于Hopfield神经网络的变电站数据负载均衡系统研究[J]. 制造业自动化，2019, 41(3): 12-16.
Deng Ke, Li Ting, Cai Ang, et al. Research on substation data load balancing system based on Hopfield neural network[J]. Manufacturing Automation, 2019, 41(3): 12-16.

[25] Hopfield J J, Tank D W. Neural Computation of Decision in Optimization Problems[J]. Biological Cybernetics (S0340-1200), 1985, 52: 141-152.

[26] Zheng Z B, Lyu M R. Collaborative Reliability Prediction for Service-Oriented Systems[C]. Proceedings of the ACM/IEEE 32nd International Conference on Software Engineering (ICSE2010), Cape Town, South Africa, May 2-8, 2010: 35-44.