

基于改良蜂群算法的 Web 服务组合优化方法

张志鹏, 周井泉

(南京邮电大学 电子与光学工程学院、柔性电子(未来技术)学院, 江苏 南京 210003)

摘要:为提高 Web 组合优化算法的开发能力和运行结果的适应度、稳定性,以满足用户对 Web 服务的服务质量(Quality of Service, QoS)需求,提出了一种基于 QoS 模型和改良蜂群算法(modified Artificial Bee Colony, mABC)的 Web 服务组合优化方法。构建应用于 Web 服务组合优化问题的 QoS 顺序数学模型,使用基于混沌的对立学习方法,在进程运行的初始化阶段生成更好的初始群体,在蜂群算法的雇佣蜂阶段和围观蜂阶段使用新的相位搜索方程和围观搜索策略,有效地提高蜂群算法的探测能力和开发能力。将改良蜂群算法与人工蜂群算法(Artificial Bee Colony, ABC)、差分进化算法(Differential Evolution, DE)、改进灰狼优化算法(Modified Grey Wolf Optimizer, MGWO)、最优导向人工蜂群算法(Guided-best Artificial Bee Colony, GABC)和改进人工蜂群算法(Improved Artificial Bee Colony, IABC)进行了多次对比实验。实验结果表明,改良蜂群算法尽管在执行时间方面比其余算法都要略微长一些,但它在更为重要的适应度、稳定性方面都优于其余几种对比算法。

关键词:云计算; Web 服务组合; 蜂群算法; QoS 属性; 混沌映射

中图分类号: TP301

文献标识码: A

文章编号: 1673-629X(2024)03-0064-06

doi: 10.3969/j.issn.1673-629X.2024.03.010

Web Service Composition Optimization Method Based on Modified Artificial Bee Colony

ZHANG Zhi-peng, ZHOU Jing-quan

(School of Electronics and Optical Engineering, School of Flexible Electronics (Future Technology),
Nanjing University of Posts and Telecommunications, Nanjing 210003, China)

Abstract: In order to improve the development capability of Web Combinatorial optimization algorithm and the adaptability and stability of the running results, and meet the user's demand for quality of service (QoS) of Web services, we propose a combinatorial optimization method of Web services based on QoS model and modified Artificial Bee Colony (mABC). The QoS sequential mathematical model applied to the combinatorial optimization problem of Web services is constructed. The chaos based opposite learning method is used to generate a better initial population in the initialization phase of the process operation. New phase search equations and spectator search strategies are used in the hire bee phase and spectator bee phase of the bee colony algorithm to effectively improve the detection and development capabilities of the bee colony algorithm, modified Artificial Bee Colony, Artificial Bee Colony (ABC), Differential Evolution (DE), Modified Grey Wolf Optimizer (MGWO), Guided-best Artificial Bee Colony (GABC), Improved Artificial Bee Colony (IABC) algorithms are compared and tested multiple times. The experiments show that although the modified ABC has slightly longer execution time than that of other algorithms, it outperforms other comparative algorithms in more important aspects of fitness and stability.

Key words: cloud computing; Web service composition; bee colony algorithm; QoS attribute; chaos mapping

0 引言

时代飞速发展, Web 服务已经成为软件行业的新技术融合点。由于用户的需求日益复杂,通过单一的 Web 服务几乎不可能满足用户的需求。因此,要以适

当的顺序组成一组基本服务,以满足用户的要求。组合现有 Web 服务以创建新 Web 服务的过程称为 Web 服务组合^[1],但 Web 服务组合往往是多种多样的,这就需要考虑如何处理此类复杂 Web 服务组合。

收稿日期: 2023-04-12

修回日期: 2023-08-16

基金项目: 国家自然科学基金(61401225)

作者简介: 张志鹏(1999-),男,硕士研究生,通信作者,研究方向为 Web 服务组合优化问题;周井泉(1963-),男,教授,博士,研究方向为通信网络的信息管理和控制。

为了为用户的任务设计复合 Web 服务 (Composite Web Service, CWS), 首先将任务划分为 n 个子任务。然后, 从有 m 个候选服务的候选池挑选一个基本的 Web 服务根据其功能属性映射到一个子任务中。最后, 将映射的 Web 服务集成到组合 Web 服务中, 子任务的执行顺序与服务组合顺序相同。Web 服务候选池中存在许多具有相同功能属性的基本服务, 从候选 Web 服务集中选择最优 CWS 称为 Web 服务选择。

每个 Web 服务都有一个核心功能, 称为该 Web 服务的功能属性。除了功能属性外, 每个 Web 服务还拥有一些非功能属性, 这些属性也被称为 QoS 属性。最优选择定义为: 既能满足用户的功能性需求, 又能满足 QoS 属性的 CWS 称为最优选择。因此, 如何挑选出最优选择是解决 Web 服务组合问题的重点。

对于此类最优选择问题, 没有哪种特定的元启发式算法一定是最好的。因此, 新的算法在 Web 服务选择领域一直广受欢迎。Wang Hei-Chia 等人结合用户对 Web 服务的主观评价数据和服务本身的客观数据, 通过模糊专家系统建立服务的 QoS 评测模型。随后基于评测后的 QoS 属性值, 利用遗传算法 (Genetic Algorithm, GA) 进行服务组合方案的求解^[2]。范小琴等人根据自然界中的物种协同进化的思想, 提出了一种协同进化的元启发算法来求解全局 Web 服务选择问题^[3]。Klein 和 Adrian 等人利用遗传算法来解决包含多个子任务的 Web 服务选择问题。赵欣超等人利用粒子群优化算法对人工免疫算法进行改良, 把其应用到全局 Web 服务选择中^[4]。黄碧晴等人利用混沌的思想提出了一个新的元启发算法来解决 Web 服务组合问题, 并取得了比传统遗传算法更好的结果。Huo, Yin 等人利用单维度小扰动的思想改进了人工蜂群算法 (Artificial Bee Colony, ABC) 并应用于 Web 服务选择中^[5]。

可见, QoS 评价模型和群体智能算法经常被用于处理 Web 服务组合问题。QoS 模型能使处理结果更加符合客户非功能需求, 而群体智能算法中的蜂群算法具有设置参数少、实现简单、工程适用性强等优点, 近年来一直被用于处理 Web 服务组合问题。但上述研究对于种群初始化、求解策略未做过多改进, 并且算法适应度、稳定性有待进一步提高, 于是该文在这两个方面展开进一步研究。

为提高算法的适应度和稳定性, 该文提出了一种新的改良人工蜂群算法用于 Web 服务的选择。它使用基于混沌的初始化技术将初始解分散在搜索空间, 采用改良的蜜源搜索方程对搜索空间进行优化, 并改良了围观蜂相位的搜索方程, 获得了良好的性能。

1 Web 服务组合与 QoS 评价模型

1.1 Web 服务组合的概念

Web Services Architecture 组织给 Web 服务组合的定义是: 一个支持网络节点交互的应用程序, 每个 Web 服务都存在明确的、机器可识别的通用标准描述来说明节点如何以特定的方式彼此交互。

Web 服务是一种可以构建分布式系统的新的网络开发技术。它同时兼备低耦合性、可复用性, 独立于编程语言和操作平台性。Web 服务还有三大协议: SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), UDDI (Universal Description Discovery and Integration)。SOAP 是一种轻量级的交换数据的规范, 指可以在分布式系统中交换标准化的信息; WSDL 用于描述一个 Web 服务及如何与 Web 服务通信; UDDI 是服务提供者和用户之间的一个链接, 使用户更加容易搜索和找到目标服务^[6]。

1.2 Web 服务组合的形式

Web 服务组合是通过调用相互连接的其他服务来构造的。服务的组合允许不同应用程序之间在 Internet 上进行协作、协调和合作。并且服务组合可用于构建一些智能化定制的应用程序, 以交付给用户新的服务和功能。在日常使用越来越多的复杂系统中, 已经出现了这种便于用户实现其需求的组合技术系统。

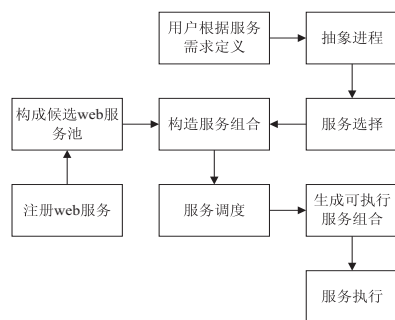


图1 Web 服务组合生命周期

Web 服务组合通常经历很多步骤, 图1展示了 Web 服务组合的一般生命周期, 主要分为4个步骤:

第一步: 定义。用户提出其个人服务需求, 先根据其需求提取关键信息, 并对其进行拆解, 分成多个子步骤去完成, 每个子步骤按一定的拓扑相连接组合, 定义为一个新的抽象流程。

第二步: 服务选择。不同的供应者向注册服务中心提供各种候选 Web 服务以形成候选 Web 服务集合。这时根据第一步得到的抽象流程, 向每一个子步骤中挑选多个符合其要求的 Web 服务。

第三步: 服务调度。对第二步得到的服务组合中每一个子步骤进行排列组合, 每一种不同的排列方法都会生成一个新的服务组合, 经过筛选对比后从中挑

选出最优的 Web 服务组合。

第四步:服务执行。面向客户执行满足其服务需求的最优 Web 服务组合。

当知道其任务具体分解形式时,便可以采用上述方法向用户提供最优的 Web 服务,当不知道任务具体需求时,即非功能需求,便可通过 QoS 模型进行处理。

1.3 Web 服务组合的 QoS 评价模型

QoS 属性是 Web 服务里非常重要的综合指标,可以将用户对某项服务非功能属性的满意度用数值表示,是客户的非功能需求的直观体现。常用的 QoS 属性包括可用性、可靠性、响应时间、延迟,它们的具体定义为:

(1)可用性:成功调用的数量与总调用数量的比率。

(2)可靠性:正确消息与总消息的比率。

(3)响应时间:发送请求到接收响应之间的时间跨度。

(4)延迟:执行请求所花费的时间。

可靠性与可用性属于积极的 QoS 属性,越高越好。响应时间与延迟属于消极的 QoS 属性,越低越好。

QoS 评价模型将 CWS 的聚合 QoS 值映射为实数,即效用值,以便后续计算使用,需要执行以下两个步骤:

(1)归一化:将不同 Web 服务的 QoS 属性值归一化为 0 到 1 之间的实数。

(2)加权:将每个归一化的 QoS 值与相应的权重相乘并相加,得到该 CWS 的效用值。

CWS 各个 QoS 属性归一化处理如下:

如果 q_k 为积极属性,则 q_k 的归一化处理结果如式 1 所示。

$$Q_k(s) = \begin{cases} \frac{q_k(s) - q_k^{\min}(s)}{q_k^{\max}(s) - q_k^{\min}(s)}, & \text{if } q_k^{\max}(s) \neq q_k^{\min}(s) \\ 1, & \text{if } q_k^{\max}(s) = q_k^{\min}(s) \end{cases} \quad (1)$$

如果 q_k 为消极属性,则 q_k 的归一化处理结果如式 2 所示。

$$Q_k(s) = \begin{cases} \frac{q_k^{\max}(s) - q_k(s)}{q_k^{\max}(s) - q_k^{\min}(s)}, & \text{if } q_k^{\max}(s) \neq q_k^{\min}(s) \\ 1, & \text{if } q_k^{\max}(s) = q_k^{\min}(s) \end{cases} \quad (2)$$

其中, Q_k 表示 QoS 属性归一化后的值, q_k 表示 QoS 属性归一化处理之前的值。该文涉及 4 种 QoS 属性,故 $k \in [1, 4]$ 且为整数。 s 表示基础 Web 服务, $q_k^{\max}(s)$ 和 $q_k^{\min}(s)$ 表示 Web 服务第 k 个属性的最大值和最小值。

服务组合执行有 4 种基本结构:顺序、并行、选择和循环,本次采用顺序结构。顺序结构下 CWS 的第 k 个 QoS 属性的聚合值如式 3 所示。

$$Q_k^{\text{agg}}(\text{CWS}) = \sum_{j=1}^n Q_k(s_j^i) \quad (3)$$

其中, s_j^i 表示第 j ($j = 1, 2, \dots, n$) 个子任务中第 i ($i = 1, 2, \dots, m$) 个候选 Web 服务。

利用上述 QoS 属性归一化处理 and QoS 聚合值计算公式,将寻找满足全局约束的最优 CWS 的问题表述为:

最大化

$$U(\text{CWS}) = \sum_{k=1}^r w_k \times Q_k^{\text{agg}}(\text{CWS}) \quad (4)$$

约束,

$Q_k \leq Q_k^{\text{agg}(\max)} - \lambda \times (Q_k^{\text{agg}(\max)} - Q_k^{\text{agg}(\min)})$, 如果为 Q_k 积极属性

$Q_k \geq Q_k^{\text{agg}(\min)} + \lambda \times (Q_k^{\text{agg}(\max)} - Q_k^{\text{agg}(\min)})$, 如果为 Q_k 消极属性

w_k 为权值且 $\sum_{k=1}^r w_k = 1$, 通常一个 QoS 属性越受客户重视其所分配的权值越大, $r = 4$, 这里表示共有 4 个 QoS 属性。对 Q_k 进行约束防止 Q_k 过大或过小, λ 为约束强度。优化函数 $U(\text{CWS})$ 即让效用值 U 最大化。

2 基于改良蜂群算法的 QoS-Web 组合优化

近年来,多种智能算法用于处理 Web 组合优化问题,其中蜂群算法在解决问题过程中具有设置参数少、实现简单、工程适用性强等优点,近年来一直是热门算法。针对蜂群算法中的群体初始化、相位搜索方程、围观搜索策略进行改良,该文提出一种改良的蜂群算法(modified Artificial Bee Colony, mABC)来解决 Web 组合优化问题。

2.1 改良蜂群算法

ABC (Artificial Bee Colony) 算法由 Karaboga 和 Basturk 提出,是一种基于蜂群的元启发式算法,基于蜜蜂的觅食行为。一群蜜蜂一起生活在一个蜂箱被称为蜂群。蜂群由三种不同类型的蜜蜂组成:(i) 雇佣蜂;(ii) 围观蜂;(iii) 侦察蜂。不同的蜂种执行各自的任務,蜂群的觅食任务是由雇佣蜂发起的。每只雇佣蜂都瞄准食物来源的位置,并收集有关该食物来源中可用花蜜量的信息。之后,它们与巢友分享关于花蜜量的信息。在收集了食物来源的信息后,围观蜂会选择更好的食物来源。当一只雇佣蜂发现食物来源被遗弃时,会恢复它作为侦察蜂的角色。这样,开发的责任由雇佣蜂和围观蜂共同承担,而探索的责任则由侦查

蜜蜂独自承担^[7]。

对传统蜂群算法的初始化策略、雇佣蜂和围观蜂的方程进行修改。

(1) 初始化策略。

为了选择更好的初始种群,提出了一种基于混沌映射的对立学习方法,如式 5 所示。

$$ch_{k+1} = \mu \times ch_k(1 - ch_k) \quad (5)$$

其中, ch_k 为 0 到 1 之间的随机数, μ 设置为 4。

基于对立学习 (Opposition Based Learning, OBL), 同时考虑估计解 X 及其对应的相反估计值, 以覆盖整个搜索空间。每个解的相反估计值的计算方法如式 6 所示。

$$X_i^o = \bar{X} + \bar{X} - X_i \quad (6)$$

其中, \bar{X} 与 \bar{X} 分别代表变量的上下限, $i \in (1, SN)$ 。因此, 该初始化策略通过同时探测原始猜测解及其 OBL 对应解, 能从更大范围的搜索空间中选择解。SN (Number of food Sources) 指食物来源的数量, 即种群大小 m 。

蜂群算法的种群初始化中有两个循环: 第一个循环中, 使用基于混沌映射原理的式 5 来进一步生成估计解 X 。第二个循环中, 使用基于 OBL 原理的式 6 得到另一个解的集合 X^o , 对所有解进行求值, 形成组合集, 利用精英主义原理得到更好的初始化种群。

(2) 雇佣蜂方程。

被雇佣的蜜蜂首先寻找新的潜在候选解, 为了充分挖掘每一种可能, 需要对各个维度信息进行交换, 提出一个新的相位搜索方程, 见式 7。

$$V_i = \varphi_{ij} X_i + (1 - \varphi_{ij}) X_k \quad (7)$$

其中, $i, k \in (1, SN)$, $k \neq i$, $\varphi_{ij} \sim U(-1, 1)$ 为 -1 和 1 之间的随机分布数。

(3) 围观蜂方程。

围观蜜蜂的功能很大程度上取决于雇佣蜜蜂, 因为它们需要从雇佣蜂那里得到有用信息再进行下一步判断处理。为了提高搜索强度, 对传统 ABC 算法中围观蜂的搜索方程进行了改良。在文献[8-9]的研究中都可以明显看出, 涉及最优解并探索其周围的搜索空间可以提高开发性能。新的围观搜索方程如式 8。

$$x_{ij} = x_{ij} + \varphi_{ij}(x_{best,j} - x_{r_1,j}) + (1 - \varphi_{ij})(x_{r_2,j} - x_{r_3,j}) \quad (8)$$

其中, r_1, r_2, r_3 是互斥随机数, 且 $i, r_1, r_2, r_3 \in (1, SN)$, $r_1 \neq r_2 \neq r_3$, $j \in (1, D)$ 是随机选择的指标。类似地, x_{best} 是当前种群中具有最佳适应度的最佳解向量。

2.2 QoS-Web 组合优化

由 QoS 的 Web 服务组合模型得到的 QoS 效用函

数 U 如式 4 所示, 作为适应度值。

改良蜂群算法应用于 Web 服务组合问题的具体步骤描述如下:

步骤 1: 参数设置阶段, 种群大小为 m , 维数为 n , 限制为 $mn/2$, 最大迭代次数为 Max。

步骤 2: 种群初始化阶段, 根据基于混沌的对立学习方法随机生成种群, 且有边界限制, 最小界为 \bar{X} , 最大界为 \bar{X} 。

步骤 3: 雇佣蜂阶段, 雇佣蜂根据邻域搜索方程 (式 7) 产生新解, 计算其适应度值, 并进行贪婪选择。

步骤 4: 围观蜂阶段, 围观蜂进行轮盘选择, 从雇佣蜂产生的解中选择一个更合适的解, 解决方案的质量取决于适应度值, 在评估适应度后, 围观蜂更新解决方案的位置, 即围观搜索策略 (式 8)。

步骤 5: 侦察蜂阶段, 如果搜索了预定义的实验次数即限制次数后, 某个解决方案的位置没有更新, 则该解决方案将被放弃, 与该解决方案相关的雇佣蜂将变为侦察蜂。侦察蜂将会生成新的解决方案。

步骤 6: 记录目前获得的最优解。

步骤 7: 若当前没有达到迭代次数则回到步骤 3, 否则输出最优解。

通过 U 的值来判断得出解的质量, 即顾客对于 Web 服务组合的满意度, U 越高表示结果越符合客户需求, U 越低表示结果不满足客户需求, U 的值越高越好。

3 实验结果与讨论

此次实验采用 Web 服务的 QWS 数据集^[10]。该数据集包含 2 507 行和 9 列。一行代表一个基本的 Web 服务, 一列代表一个 QoS 属性。因此此次实验使用 2 500 行, 4 列, 即 Web 服务数量为 2 500, QoS 属性数量为 4。

为了分析算法性能, 将 mABC 与 5 种现有的元启发式算法 (人工蜂群算法^[11] (ABC)、差分进化算法^[12] (DE)、改良灰狼算法^[13] (MGWO)、最优导向人工蜂群算法^[14] (GABC) 和改进人工蜂群算法^[15] (IABC)) 进行了比较。

QoS 的 4 个属性是: 响应时间 (ms)、延迟 (ms)、可用性 (%) 和可靠性 (%)。所有的实验都是在 64 位 3.40 GHz Intel (R) 酷睿 (TM) i7-3770 PC 上的窗口环境中实现的, 具有 8 GB RAM。在实验中对 mABC 进行了以下参数设置。在引言中提到的 n 个子任务和每个子任务中的 m 个候选 Web 服务, 对应算法中的维数 (n) 和种群大小 (m), 分别设置为 10 和 250, 限制设置为 ($mn/2$)。所有算法的停止条件是最大迭代次数

(Max), 设置为 500, 所有算法迭代次数为 Max=500。

图 2 显示了 mABC 算法与其它算法的适应度比较。可以看出, 当迭代次数达到 25 次时, mABC 算法适应度为 5.69, 开始始终优于其他算法。当迭代次数达到 50 次时, 适应度为 5.72, 已经高于其他算法的最优值。迭代次数 100 时, 其余算法适应度趋于稳定不再增加, 而改良蜂群算法仍在逐步提升直至 200 次时达到最优值 5.8, 其余算法在迭代 50 到 75 次之间逐渐趋于稳定, 但探索的不够充分, 无法达到更高的适应度, 而 mABC 算法虽然收敛的慢, 但仍在探索解, 并在不断提高最大适应度。在整个迭代过程中可以看出, 从第 25 次迭代开始 mABC 的适应度都比其余算法的高。并且 mABC 算法能在保证收敛的情况下取得更高的适应度, 更适合解决 Web 服务组合问题, 与理论上的预期相同。

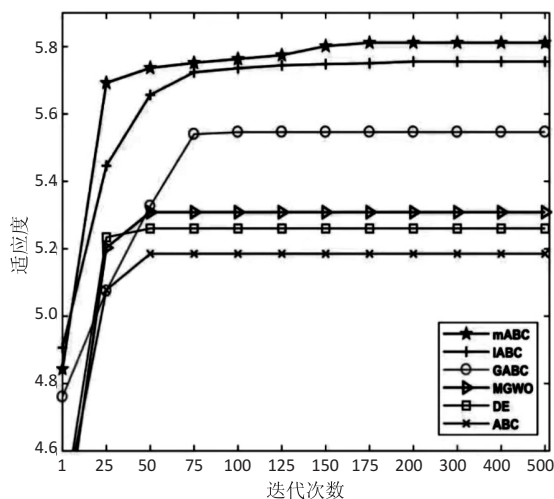


图 2 随迭代次数变化的适应度

为对比不同算法适应度标准差的情况, 分析了数据的最大值、最小值、中间值、平均值, 如表 1 所示, 其中 mABC 算法适应度的最大值 5.8、最小值 5.69、中间值 5.78、平均值 5.76, 与其余几种算法的数据对比可见 mABC 算法的最大值、最小值、中间值、平均值都比其他几种算法的优。并且 mABC 算法相较原始 ABC 算法在适应度最大值提升了 11.9%, 适应度平均值提升了 12.5%。

表 1 适应度 4 种数据

	mABC	IABC	GABC	MGWO	DE	ABC
最大值	5.8	5.72	5.53	5.3	5.26	5.18
最小值	5.69	5.45	5.31	5.2	5.23	5.08
中间值	5.78	5.7	5.51	5.29	5.25	5.18
平均值	5.76	5.67	5.48	5.25	5.21	5.12

标准差是衡量一个数据稳定状况的值, 标准差越低表示该数据越稳定, 越高则表示越不稳定。通过表 1 数据计算几种算法适应度的标准差, 如图 3 所示,

mABC 的标准差为 0.043, 明显低于其余几种算法, 说明 mABC 算法相较于其余几种算法具有较好的稳定性, 更适合解决 Web 服务组合问题, 与理论上的预期相同。

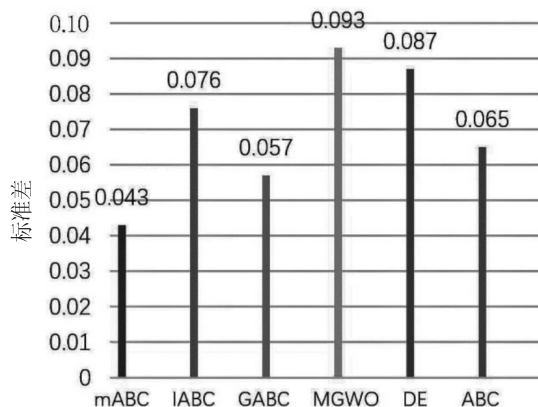


图 3 算法的适应度标准差

执行时间指的是算法完成一次实验需要的时间, 通过将实验运行 100 次记录算法的平均时间使结果更加可靠。图 4 为 6 种算法执行时间统计图, 其中 mABC 的执行时间为 1.81 s, 高于其余几种算法, 这是由于在算法的种群初始化阶段加入了混沌映射, 同时这也使得算法更加复杂, 执行时间增加, 但增加了算法的探索性。从图 2 中也可以看出, mABC 的最大适应度在其余算法都已经收敛时仍在增加, 说明初始化策略的改动使得算法牺牲一定的执行时间去换取更大的适应度^[16-19]。

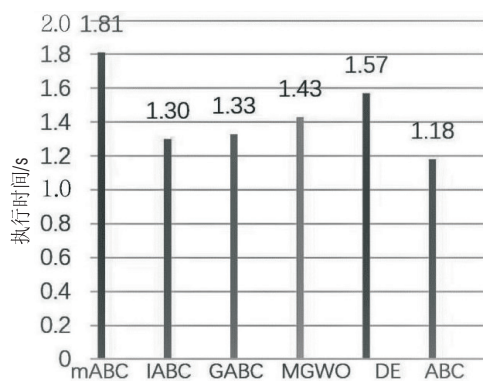


图 4 算法执行时间

4 结束语

文中给出了 Web 服务组合的 QoS 数学模型, 通过使用基于混沌的对立学习方法生成初始种群提高传统 ABC 算法的开发能力, 再修改雇佣蜂和围观蜂的搜索方程, 改良了 ABC 算法并用来解决 Web 服务组合优化问题。实验数据表明, mABC 算法在执行时间方面比其余算法略微都要长一些, 但在适应度、稳定性方面有较好表现, 优于其余参照算法, 在处理 Web 服务组合问题上具有可行性和有效性。

该文的研究空间还很大,在后续的工作中,笔者将会对 ABC 算法的寻优过程进行进一步优化并投入到 Web 服务组合优化研究。

参考文献:

- [1] 刘 聪,杨怀洲. Web 服务动态组合关键方法研究[J]. 计算机时代,2021(8):21-23.
- [2] ZHONG Fuli, LI Hui, ZHONG Shouming. An improved artificial bee colony algorithm with modified neighborhood based update operator and independent inheriting search strategy for global optimization[J]. Application of Artificial Intelligence, 2017, 58(15):134-156.
- [3] DOAA E, EMAN N, ALAA E G, et al. A self-healing model for QoS-aware web service composition[J]. International Arab Journal of Information Technology, 2020, 17(6):87-93.
- [4] 吴明礼,魏瑞珍. 主客观权重遗传算法在服务选择中的研究[J]. 计算机工程与设计, 2020, 41(3):729-734.
- [5] WANG Yucai, WANG Jiesheng, HOU Jianing, et al. Natural heuristic algorithms to solve feature selection problem[J]. Engineering Letters, 2023, 31(1):34-39.
- [6] 胡卓娅,翁 健. 基于人工蜂群算法的自适应谱聚类算法[J]. 重庆理工大学学报:自然科学版, 2020, 34(3):137-144.
- [7] 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.
- [8] AL-MASRI E, MAHMOUD Q H. QoS-based discovery and ranking of Web services[J]. ICCCN, 2007, 8(2):529-534.
- [9] 胡 强,田雨晴,綦浩泉,等. 基于改进人工蜂群算法的云制造服务组合优化方法[J]. 通信学报, 2023, 44(1):200-210.
- [10] 张明珠,曹 杰,王 斌. 基于精英集的多目标差分进化聚类算法[J]. 计算机工程与科学, 2021, 43(1):170-179.
- [11] 杨沐杰,钟羽中,郭 斌,等. 基于改进灰狼优化算法的点云配准[J]. 计算机仿真, 2022, 39(12):513-518.
- [12] HUO Y, ZHUANG Y, GU J, et al. Discrete guided-best artificial bee colony algorithm for cloud service composition[J]. Intell, 2015, 42(4):661-678.
- [13] 杨 帆,李邵梅,金柯君. 基于改进人工蜂群算法的文本对抗样本生成[J]. 计算机系统应用, 2022, 31(11):238-245.
- [14] 叶廷宇,叶 军,王 晖,等. 结合人工蜂群优化的粗糙 K-means 聚类算法[J]. 计算机科学与探索, 2022, 16(8):1923-1932.
- [15] AL-MASRI E, MAHMOUD Q H. Toward quality-driven Web service discovery[J]. IT Professional, 2008, 10(3):24-28.
- [16] 庄鹤林,杨火根,夏小云,等. 关于矩阵乘法问题的人工蜂群优化算法研究[J]. 计算机工程与科学, 2021, 43(12):2131-2138.
- [17] 董 晓. 一种高效的改进人工蜂群算法设计[J]. 现代电子技术, 2021, 44(22):179-182.
- [18] 高文斌,王 睿,王田丰,等. 基于深度强化学习的 QoS 感知 Web 服务组合[J]. 计算机技术与发展, 2022, 32(6):92-98.
- [19] YANG Hongwei, XUE Fucheng, ZHU Hongmei, et al. Web service composition optimization based on adaptive mutant beetle swarm[J]. Journal of Physics: Conference Series, 2020, 1651(1):432-436.