doi:10.3969/j.issn.1005-3026.2014.08.010

基于改进蚁群算法的Web服务选择

盛国军[1],温 涛[1,2],郭 权[2],印 莹[1]
(1.东北大学 信息科学与工程学院,辽宁 沈阳 110819;2.大连东软信息学院,辽宁 大连 116023)

摘 要:提出一种改进的蚁群算法并将其应用于Web服务选择问题中.该算法使用非线性动态变化的伪随机比例选择参数及蚂蚁多重最优解随机加权路由选择算法控制蚁群的行为,使用5维Web服务质量向量和蚁群适应度函数评价蚂蚁构造的路径质量,蚂蚁根据其构造的路径质量进行信息素更新;该算法使蚁群在其解空间的进化能力得到很大的提高.实验证明,该算法在Web服务选择问题上比传统的蚁群算法效率更高.

关 键 词:服务选择;蚁群算法;随机加权路由选择;动态伪随机比例选择参数;算法性能评价指标
中图分类号:TP 311 **文献标志码**:A **文章编号**:1005-3026(2014)08-1107-05

Web Service Selection Based on Modified Ant Colony Optimization

SHENG Guo-jun[1], *WEN Tao*[1,2], *GUO Quan*[2], *YIN Ying*[1]
(1. School of Information Science & Engineering, Northeastern University, Shenyang 110819, China; 2. Dalian Neusoft Information Institute, Dalian 116023, China. Corresponding author: SHENG Guo-jun, E-mail: shengguojun@neusoft.edu.cn)

Abstract: Focusing on Web service selection problem, a new modified ant colony optimization (ACO) algorithm is proposed. Both a nonlinear dynamic parameter of the pseudorandom proportion selection rule and a multiple-optimal-solution random-weighted route selection algorithm are employed in the algorithm proposed to control the behavior of ant colony. Besides, a five-dimensional service quality vector and the fitness function are used in the algorithm to evaluate the ant solutions, and each ant updates the pheromone according to the quality of their solutions they built. With these measures, the evolution ability of ant colony can be significantly improved. The experimental results show that the proposed algorithm outperforms traditional ACO algorithms.

Key words: service selection; ant colony optimization; random-weighted route selection; dynamic pseudorandom proportion selection parameter; algorithm performance evaluation index

Web服务具有完全开放、松散耦合、标准协议和高度可集成等特征,支持通过基于Internet的协议和基于XML的消息进行交互.Web服务组合按照业务需求将分布在Internet上的多个Web服务进行合成,在实现软件重用的同时,使系统更加适应开放动态的网络环境.组合服务流程中每个节点都可能存在大量功能相同且服务质量不同的候选服务;如何动态地从海量组合方案中选择出满足用户质量需求的服务组合方案,已成为服务组合优化领域中的一个关键问题,具有重要理论意义和实用价值,已有学者在Web服务选择领域进行了相关研究.Wang等[1]提出一种基于服务质量的服务选择方法,该方法使用模糊逻辑控制动态服务选择过程,并使用混合整数规划协助用户获得最合适的服务.Kumar等[2]提出一种基于Web服务质量约束的Web服务选择方法,该方法使用层次分析法对服务的各质量属性进行分级,以选择出质量级别最高的服务组合方案.文献[3]设计了一种面向动态服务选择的离散蚁群算法,针对进化算法容易陷入局部极值这

收稿日期:2013-10-21
基金项目:国家自然科学基金资助项目(61170168,61170169,61100028);中央高校基本科研业务费专项资金资助项目(N110404017).
作者简介:盛国军(1974-),男,辽宁大连人,东北大学博士研究生;温 涛(1962-),男,陕西宝鸡人,东北大学教授,博士生导师.

一缺陷定义了蚂蚁无希望/重希望准则以保证蚁群多样性. 文献[4]把信任和信任度引入到服务组合中来,采用蚁群系统作为优化工具,设计了两种动态服务选择方法. 文献[5]将智能优化蚁群算法与 Web 服务组合相结合,对 QoS 驱动下的服务组合进行求解. 文献[6]提出一种基于 QoS 和蚁群算法的动态服务组合优化方法,采用多目标路径优化选择模型来解决服务选择问题. 本文基于上述研究对 Web 服务选择问题进行相关研究.

1 问题描述

定义 1 Web 服务实例(Web service instance,WSI):指具有明确 URL 地址,可被直接调用的 Web 服务.

定义 2 抽象 Web 服务(abstract Web service,AWS):AWS = $\{WSI_i\}$ 是具有相同功能属性和不同非功能属性的一组 WSI 的集合.

定义 3 组合服务(composited service,CS)是一个 6 元组:CS = (S,R,QoS,C,W,f),其中 $S=\{AWS_i\}$ 表示构成组合服务的抽象 Web 服务集合,R 为 AWS 间关系的集合;QoS 是组合服务的质量;C 是组合服务质量约束;$W=\{w_i\}$ 是 QoS 中各元素对应的权重且 $\sum_{i=1}^{5} w_i = 1$;f 是组合服务质量评价函数.

定义 4 组合服务实例(composited service instance,CSI):从组合服务 CS 各抽象服务中选择一个具体 Web 服务实例替代抽象服务,形成一个可执行的组合服务方案.

2 算法描述

2.1 MACS 算法基本概念

定义 5 蚂蚁是一个 9 元组:$A=(k,C,c,l_t,P,f(P),P^*,f_v^*,P_L)$. 其中 $k\in\mathbf{N}^+$ 是蚂蚁标志,$C\in\mathbf{N}^+$ 是蚂蚁最大迭代次数,$c\in[1,C]$ 是蚂蚁当前迭代数且 c 以步长 1 递增,l_t 是蚂蚁路由禁忌表,P 是蚂蚁一次迭代后构造的解,$f(P)$ 是解 P 的评价函数,P^* 是在函数 $f(P)$ 评价下的蚂蚁历史最优解,$f_v^*=f(P^*)$ 是对最优解 P^* 的评价值,P_L 是蚂蚁上一次迭代过程中构造的解,第一次迭代 $P_L=\mathrm{null}$.

定义 6 蚁群是一个 4 元组 AS = (AC,P_g^*,$f_{v,g}^*$,ASD). AC = $\{A_i\}$ 是 m 个蚂蚁的集合,P_g^* 是蚁群历史最优解,$f_{v,g}^*=f(P_g^*)$ 是 P_g^* 对应的评价值,ASD 是蚁群多样性.

2.2 MACS 算法描述

采用 5 维服务质量[7] QoS = $\{Q_i \mid Q_i = t,c,a,r_{el},r_{ep}\}$,分别为响应时间、费用、可用性、可靠性、信誉度. 设用户对组合服务 CS 质量要求为:QC = $\{t\leqslant T, c\leqslant C, a\geqslant A, r_{el}\geqslant R_{EL}, r_{ep}\geqslant R_{EP}\}$,$T$,$C$,$A$,$R_{EL}$,$R_{EP}$为用户给定的值,则 Web 服务组合优化问题数学模型为 $\max f=\sum_{i=1}^{5}(Q_i\times w_i)$,MACS 算法以函数 f 作为各蚂蚁适应度函数,用于评价蚂蚁当前解向量的优劣. 与 MMAS[8] 算法相似,MACS 算法将蚂蚁路径上的信息素的变动范围设定为 $[PH_{max}, PH_{min}]$,$PH_{max}=0.999$,$PH_{min}=0.001$,将各条有向边上的初始信息素设置为信息素最大值 PH_{max},采取动态变化的伪随机比例选择参数 q_0,取值按照式(1)进行:

$$q_0 = 1-\sqrt{\frac{c\times Q_{max}}{C\times(1+C-c)}}. \tag{1}$$

式中:c 为蚂蚁当前迭代次数;C 为蚂蚁迭代的总次数;$Q_{max}=0.999$ 为预先给定的常量. 蚂蚁在选择下一节点前先生成一个[0,1]区间的随机数 rand,然后将 rand 与 q_0 进行比较,并根据比较结果进行路由选择,蚂蚁路由选择算法如下.

算法 1 MACS 中蚂蚁路由选择算法.

输入:蚂蚁 A_k,有向图 $G=\langle V,E\rangle$,信息素与启发信息重要性参数 α 和 β.

输出:蚂蚁 A_k 将要访问的下一个节点 nextNode.

步骤 1 q_0 = getDynamicQ0($A_k.c$,$A_k.C$);按照式(1)计算伪随机比例选择参数 q_0.

步骤 2 rand = makeRand();生成一个[0,1]之间的随机数 rand.

步骤 3 if(rand > q_0){采用传统的 ACO 算法的选择路径方法选择下个节点 nextNode. return nextNode};else{采用算法 2 的方法选择下个节点 nextNode. return nextNode}.

算法 2 随机加权路由选择算法.

输入:蚂蚁 A_k,有向图 $G=\langle V,E\rangle$,信息素与启发信息重要性参数 α 和 β.

输出:蚂蚁 A_k 将要访问的下一个节点 nextNode.

步骤 1 neighbours$_k$ = getNeighbours($A_k.l_t$,G);根据蚂蚁禁忌表 $A_k.l_t$ 和图 G 的拓扑结构获取 A_k 下一跳可能访问的所有邻居节点形成的集合,用各节点在该集合中的顺序索引编号来表示

节点的位置.

步骤 2　获取 A_k 上次迭代得到的解$A_k.P_L$中下一跳节点的位置 $\mathrm{idx}(A_k.P_L)$.

步骤 3　获取 A_k 历史最优解 $A_k.P^*$ 中下一跳节点的位置 $\mathrm{idx}(A_k.P^*)$.

步骤 4　获得整个蚁群历史最优解AS. P_g^* 中下一跳节点的位置 $\mathrm{idx}(\mathrm{AS}.P_g^*)$.

步骤 5　计算随机变量 P^k 的值：

$$P^k=(\tau^\alpha(A_k.P_L)\mu^\beta(A_k.P_L)+\tau^\alpha(A_k.P^*)\times\mu^\beta(A_k.P^*)+\tau^\alpha(\mathrm{AS}.P_g^*)\mu^\beta(\mathrm{AS}.P_g^*))/3.$$

步骤 6

$$r=\mathrm{if}\ f(\mathrm{isStagnation}(\mathrm{AS}.f_{v,g}^*,\eta),\mathrm{randl},P^k).$$

步骤 7　计算 A_k 下一跳节点的位置（即 Web 服务实例的索引编号）：$\mathrm{idx}_{next}=\mathrm{idx}(A_k.P_L)+2r\times(\mathrm{idx}(A_k.P^*)+\mathrm{idx}(A_k.P^*)-2\times\mathrm{idx}(A_k.P_L))$.

步骤 8

return $\mathrm{getRightNode}(\mathrm{idx}_{next},\mathrm{neighbours}_k)$.

在算法 2 的步骤 5 中，$\tau^\alpha(\mathrm{AS}.P_g^*)\times\mu^\beta(\mathrm{AS}.P_g^*)$是 A_k 在当前节点到蚁群全局最优解对应的下一跳节点之间信息素与启发信息的带权乘积，则随机变量 P^k 表示 3 个路由选择概率：$\tau^\alpha(\mathrm{AS}.P_g^*)\mu^\beta(\mathrm{AS}.P_g^*)$，$\tau^\alpha(A_k.P^*)\times\mu^\beta(A_k.P^*)$，$\tau^\alpha(A_k.P_L)\mu^\beta(A_k.P_L)$的平均值. 在算法 2 的步骤 6 中，根据蚁群 AS 的进化是否停滞（用函数 $\mathrm{isStagnation}(\mathrm{AS}.f_{v,g}^*,\eta)$表示）来获得随机变量 r 的值：若 AS 进化停滞，则 $r=\mathrm{rand1}$，rand1 是一个随机生成的(0,1]之间的随机数；当 AS 全局最优解对应的适应值 $f_{v,g}^*$在 η 期迭代内没有进化，则判断蚁群 AS 进化停滞，此处用到 $\mathrm{AS}.f_{v,g}^*$的指标是指蚁群在上一次及以前迭代中获得的最优适应值. 在算法 2 的步骤 7 中，idx_{next}表示蚂蚁 A_k 下一跳将要选择的节点位置是 A_k 上一代在该跳上所选择的对应位置 $\mathrm{idx}(A_k.P_L)$的基础上再加上一个随机加权（权值为 $2r$）有符号量，经过步骤 7 所计算出来的 idx_{next}可能为浮点数、负数或是绝对值大于$|\mathrm{neighbours}_k|$的数，因此用步骤 8 中函数 $\mathrm{getRightNode}(\mathrm{idx}_{next},\mathrm{neighbours}_k)$计算正确范围内下一个节点位置 $\mathrm{idx}_{next}\in\{1,2,\cdots,|\mathrm{neighbours}_k|\}$，其主要逻辑如下：

if($|\mathrm{idx}_{next}|>|\mathrm{neighbours}_k|$),

　　$\mathrm{idx}_{next}=\mathrm{idx}_{next}\bmod|\mathrm{neighbours}_k|$;

if($\mathrm{idx}_{next}<0$), $\mathrm{idx}_{next}=|\mathrm{neighbours}_k|+\mathrm{idx}_{next}$;

if($\mathrm{idx}_{next}<\lfloor\mathrm{idx}_{next}\rfloor+0.5$), $\mathrm{idx}_{next}=\lfloor\mathrm{idx}_{next}\rfloor$;

if($\mathrm{idx}_{next}>\lfloor\mathrm{idx}_{next}\rfloor+0.5$), $\mathrm{idx}_{next}=\lceil\mathrm{idx}_{next}\rceil$;

if($\mathrm{idx}_{next}=\lfloor\mathrm{idx}_{next}\rfloor+0.5$),

$\mathrm{idx}_{next}=\mathrm{if}\ f(R_{0,1}=0,\lfloor\mathrm{idx}_{next}\rfloor,\lceil\mathrm{idx}_{next}\rceil)$.

随机变量 $R_{0,1}$是做一次概率为 0.5 的伯努利实验结果：

$$P\{R_{0,1}=k\}=p^k(1-p)^{1-k},k=0,1,(p=0.5).$$

MACS 算法描述如下.

算法 3　改进的蚁群系统.

输入：蚁群规模 m，最大进化代数 C，CS 中抽象服务数 $n=|\mathrm{CS}.S|$，CS 中每个抽象服务 AWS_d ($d\in\{1,2,\cdots,n\}$)上的候选 Web 服务实例数 m_d，AWS_d 中每个 Web 服务实例 WSI_i ($i\in\{1,2,\cdots,|\mathrm{AWS}_d|\}$)的 5 维质量向量，组合服务 CS 的质量权重向量 CS. W.

输出：蚁群搜索空间中的最优解或次优解.

步骤 1 根据$|\mathrm{CS}.S|$及各 AWS 对应的 WSI 数初始化有向图 $G=\langle V,E\rangle$，设置图各有向边初始信息素量为 PH_{max}，各服务实例节点 WSI_i 的启发信息设置为μ_{init}. μ_{init}的计算方法如下：

$$\mu_{init}=\frac{Q}{\sqrt{t^2+c^2+1/a^2+1/r_{el}^2+1/r_{ep}^2}}. \tag{2}$$

其中 QoS 是 WSI_i 的 5 维质量向量，$Q>0$ 为给定的常数. 设置当前进化代数 $c=0$，生成蚁群中 m 只蚂蚁.

步骤 2　初始化 m 只蚂蚁的状态，将它们置于图的开始节点，蚂蚁当前进化代数 c 递加 1.

步骤 3　do：for $k=1$ to m

蚂蚁 A_k 使用算法 1 获得下一个节点 nextNode；

蚂蚁 A_k 访问 nextNode，将 nextNode 加入 A_k 禁忌表；

蚂蚁 A_k 进行信息素局部更新；

endfor

until：每只蚂蚁都构造了一个完整的解（禁忌表满）.

for $k=1$ to m

用$f(A_k.P)$评价蚂蚁 A_k 解的质量；

更新蚂蚁 A_k 本身的历史最优解 $A_k.P^*$；

$A_k.P_L=A_k.P$.

endfor

根据每只蚂蚁解的质量选出蚁群全局最优蚂蚁 A_k：

$\mathrm{AS}.P_g^*=A_{best}.P^*$, $\mathrm{AS}.f_{v,g}^*=A_{best}.f_v^*$；

蚂蚁 A_{best}进行信息素的全局更新.

步骤 4　若 $c\leqslant C$ 则跳到步骤 2.

步骤 5　输出 AS. P_g^* 和 AS. $f_{v,g}^*$.

在步骤 3 中，蚂蚁 A_k 使用式(3) 更新局部信

息素:

$$\left.\begin{aligned}\tau_{rs}(c)&=(1-\varphi)\times\tau_{rs}(c-1)+\varphi\times\Delta\tau_{rs},\\ \Delta\tau_{rs}&=\theta\times A_k.f_{\mathrm{v}}^{*}/n.\end{aligned}\right\}\quad(3)$$

有序偶〈r,s〉是蚂蚁路径上正在构建的一条边;φ 是信息素挥发系数;$\theta\in(0,1]$,用于控制信息素增量的比例,本文取 $\theta=0.1$;$A_k.f_{\mathrm{v}}^{*}$ 是蚂蚁 A_k 到上一次迭代($c-1$)为止找到的最优解对应的适应值;n 是蚂蚁解向量的维数. 局部信息素更新规则的运用使蚂蚁路径各边上的信息素量逐渐减少,可以有效地避免各蚂蚁过早收敛于同一条路径. 在步骤 3 中,蚂蚁 A_{best} 使用式(4)进行全局信息素更新:

$$\left.\begin{aligned}\tau_{ij}(c)&=(1-\rho)\times\tau_{ij}(c-1)+\Delta\tau_{ij}^{\mathrm{best}},\\ \Delta\tau_{ij}^{\mathrm{best}}&=\rho\times A_{\mathrm{best}}.f_{\mathrm{v}}^{*}.\end{aligned}\right\}\quad(4)$$

MACS 采用与 MMAS 相同的全局信息素更新方法,在所有蚂蚁都构造完成一条完整的路径后,只有解质量最高的蚂蚁 A_{best} 才有机会更新其路径上的信息素. 式(4)中,有序偶〈i,j〉是 A_{best} 构建的路径上的一条边,ρ 是信息素挥发系数,$A_{\mathrm{best}}.f_{\mathrm{v}}^{*}$ 是蚂蚁 A_{best} 本次迭代找到的最优解对应的适应值. 全局信息素更新规则和伪随机比例规则的运用使得蚂蚁的搜索范围主要集中在蚁群全局最优路径附近,避免了蚂蚁的盲目随机搜索,使整个搜索过程更加具有指导性.

3 实验与结果分析

3.1 实验环境与实验数据

本实验组合服务采用顺序模型,以随机生成的 5 组数据 $D_1\sim D_5$ 作为实验数据集,旨在测试 MACS 及相关算法在不同解空间大小的数据集下的表现. 本实验设置总迭代数 $C=300$,蚁群种群规模 $m=16$. 实验环境为:HP Proliant DL380 G5, Intel(R) Xeon(R) CPU E5450 @ 3.00GHz,4GB RAM, Microsoft Windows Server 2003 Enterprise Edition Service pack 2, JDK1.6.0, Eclipse SDK Version:3.5.2, MATLAB7.4.0(R2007a). ACO 相关各算法全部用 Java 语言实现,并在实验过程中将实验结果写入文件,然后用 MATLAB 从文件中读取实验结果数据并作图.

3.2 实验评价方法

定义 7 算法性能评价指标(algorithm performance evaluation index, APEI):

$$\mathrm{APEI}=\left(\frac{\overline{f_{\mathrm{v,g\_}n}}+\overline{\overline{f_{\mathrm{v,c\_}n}}}}{2\times f_{\mathrm{v,g\_best}}^{*}}\right)\times\left(1+\frac{(t_{\mathrm{g\_}n}^{*})^{\lambda}}{1+(c_{\mathrm{g\_}n}^{*})^{\gamma}}\right).\quad(5)$$

APEI 指标由两部分构成. 在第一部分中,$\overline{f_{\mathrm{v,g\_}n}}=\frac{1}{n}\sum_{e=1}^{n}f_{\mathrm{v,g}}$ 是对算法做 n 次独立重复实验(e 为实验序号),所有实验所获得的最终适应值的平均;$\overline{\overline{f_{\mathrm{v,c\_}n}}}=\frac{1}{n}\sum_{e=1}^{n}\left(\frac{1}{C}\sum_{c=1}^{C}f_{\mathrm{v,g}}\right)$ 是 n 次独立重复实验中的每次实验所有迭代所获得的适应值的平均值的和除以实验次数 n;$f_{\mathrm{v,g\_best}}^{*}$ 是解空间中真正最优适应值. 进化比例$(\overline{f_{\mathrm{v,g_}n}}+\overline{\overline{f_{\mathrm{v,c_}n}}})/(2\times f_{\mathrm{v,g_best}}^{*})$的含义是:算法重复运行 n 次后得到的平均迭代过程适应值与平均迭代最终适应值占解空间中真正最优适应值的百分比,该值越大说明迭代过程中以及迭代结束后所获得的适应值至解空间中真正的最优适应值的距离越近,所获得的解的质量越高,算法的进化能力越强. 在 APEI 指标第二部分中,$t_{\mathrm{g\_}n}^{*}$ 是算法重复执行 n 次发现空间真正最优解的总次数,$c_{\mathrm{g\_}n}^{*}$ 是算法重复执行 n 次发现空间真正最优解所用的总迭代次数;λ 与 γ 分别是 $t_{\mathrm{g\_}n}^{*}$ 和 $c_{\mathrm{g\_}n}^{*}$ 重要程度的影响因子. APEI 指标第二部分含义是:在 n 次重复实验中,获得解空间真正最优解次数 $t_{\mathrm{g\_}n}^{*}$ 越多且所用的迭代次数 $c_{\mathrm{g\_}n}^{*}$ 越少的算法更优. 本实验参与比较的 3 个算法分别为:传统 ACO 算法、最大最小蚂蚁系统 MMAS,以及本文所提算法 MACS. 各算法的相关参数如下:全局更新信息素蒸发系数 $\rho=0.178\,4$,局部更新信息素蒸发系数 $\varphi=0.178\,4$,随机比例规则和伪随机比例规则中的信息素重要性参数 $\alpha=1.0$,启发信息重要性参数 $\beta=1.0$, MMAS 算法和 MACS 算法中,所有路径上信息素含量的上限和下限分别设置为:$\mathrm{PH}_{\max}=0.999$, $\mathrm{PH}_{\min}=0.001$, MMAS 算法中伪随机比例选择参数 $q_0=0.812\,5$.

3.3 各算法实验结果

在数据集 D_4 上做单次实验,记录该实验 300 次迭代中获得的各指标随迭代次数的变化过程,并取移动平均值作图. 蚁群全局最优解向量对应的适应值 AS. $f_{\mathrm{v,g}}^{*}$ 的进化过程如图 1 所示.

由图 1 可见,在整个 300 次迭代过程中,相对传统 ACO 和 MMAS 算法,MACS 算法表现出优秀的进化能力,在整个迭代过程中发现的最优适应值始终高于其他算法;在本次实验中,MACS 在第 100 代左右找到并最终收敛于解空间最优适应值 2.897 3. MACS 算法使用最优解向量随机加权路由选择算法(见算法 2),使得蚁群在迭代早期解空间中"更优解"较丰富的情况下加大空间开拓能力,尽快找到更优解;在迭代的后期由于解空

间中“更优解”变得稀少而难找，MACS 算法将蚁群中的大部分蚂蚁集中在目前最优解附近进行精细查找. 图 2 为各蚁群算法在数据集 D_4 上分别做 150 次独立实验，每次实验所获得的最终适应值的情况；可见在多次独立重复实验中，MACS 算法所获得的最终适应值都优于 MMAS 和 ACO 算法，且其进化结果保持稳定.

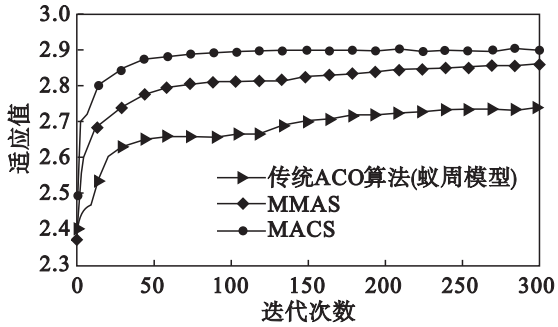

图 1　蚁群全局最优适应值随迭代次数进化过程
Fig. 1　Global best fitness value of the ant colony vs iterations

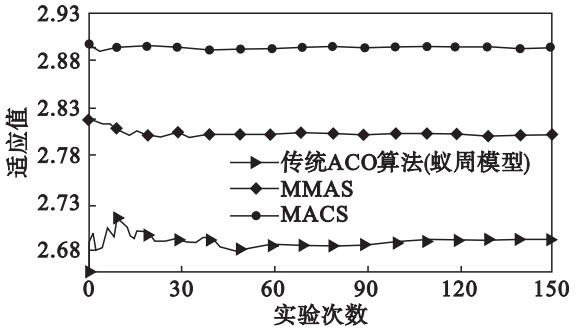

图 2　数据集 D_4 上 150 次重复实验各算法最终适应值
Fig. 2　Fitness values in 150 repeated experiments on D_4

3.4　各蚁群算法综合性能实验

由于参与比较的各蚁群算法中均使用了随机变量，使单次实验结果具有不确定性；而实验所用数据为随机生成，同一算法中的蚁群在不同问题规模下的表现可能存在差异，因此本节将 3 种蚁群算法在具有不同问题规模的 5 个数据集 $D_1 \sim D_5$ 上分别做 150 次独立重复实验，每次实验迭代 300 次，记录各算法在各数据集上多次实验所得适应值的平均值、发现极值的次数，以及发现所有极值所用的平均迭代次数，并以这些指标为基础，用式(5)计算各算法的 APEI 值，以及各算法的 APEI 与 ACO 算法 APEI 的差值. 计算各算法在数据集 $D_1 \sim D_5$ 上 APEI 均值：$\mathrm{APEI}_{\mathrm{ACO}} = 0.92$，$\mathrm{APEI}_{\mathrm{MMAS}} = 1.02$，$\mathrm{APEI}_{\mathrm{MACS}} = 1.23$. 可见，在 Web 服务选择问题中，针对综合算法性能评价指标 APEI，MACS 算法的综合性能比 MMAS 算法高出 20.323 9%. 在 Web 服务组合优化问题上，MACS 算法表现出良好的综合性能，MMAS 算法次之，传统 ACO 算法最差.

4　结　　语

与标准 MMAS 算法相比，本文提出的 MACS 算法在 Web 服务选择问题上的综合性能更高. 提出的动态变化伪随机比例路由选择参数、最优路径随机加权路由选择算法以及实验与评价方法等为 Web 服务选择问题和 ACO 相关算法的研究提供了新思路；但 MACS 算法能否在其他优化问题上取得好的应用效果还有待进一步研究.

参考文献：

[1] Wang S G, Sun Q I, Zou H, *et al*. Fuzzy logic control for Web service selection[J]. *Information Technology Journal*, 2012, 11(3): 399 – 401.

[2] Kumar R D, Zayaraz G. A QoS aware quantitative Web service selection model[J]. *International Journal on Computer Science and Engineering*, 2011, 3(4): 1534 – 1538.

[3] 范小芹，蒋昌俊，方贤文，等. 基于离散微粒群算法的动态 Web 服务选择[J]. 计算机研究与发展，2010，47(1)：147 – 156.
(Fan Xiao-qin, Jiang Chang-jun, Fang Xian-wen, *et al*. Dynamic Web service selection based on discrete particle swarm optimization[J]. *Journal of Computer Research and Development*, 2010, 47(1): 147 – 156.)

[4] 王勇，代桂平，侯亚荣. 信任感知的组合服务动态选择方法[J]. 计算机学报，2009，32(8)：1668 – 1675.
(Wang Yong, Dai Gui-ping, Hou Ya-rong. Dynamic methods of trust-aware composite service selection[J]. *Chinese Journal of Computers*, 2009, 32(8): 1668 – 1675.)

[5] 刘波，吴惕华，张庆彬. 基于改进蚁群算法的 Web 服务组合优化[J]. 仪器仪表学报，2008，29(4)：161 – 164.
(Liu Bo, Wu Ti-hua, Zhang Qing-bin. Web services composition optimization based on the modified ant colony algorithm[J]. *Chinese Journal of Scientific Instrument*, 2008, 29(4): 161 – 164.)

[6] Zhang W, Chang C K, Feng T M, *et al*. QoS-based dynamic Web service composition with ant colony optimization[C]// Proceedings of IEEE 34th Annual Computer Software and Applications Conference. Seoul, 2010: 493 – 502.

[7] Zeng L, Benatallah B, Ngu A H H, *et al*. QoS-aware middleware for Web services composition[J]. *IEEE Transactions on Software Engineering*, 2004, 30(5): 311 – 326.

[8] Stützle T, Hoos H. Improvements on the ant system: introducing MAX-MIN ant system[C]// Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms. Norwich, 1997: 245 – 249.