

QoS 全局最优的多目标 Web 服务选择算法^{*}

方其庆^{1,2}, 刘庆华², 彭晓明³, 胡亚慧²

(1. 通信指挥学院, 武汉 430010; 2. 空军雷达学院, 武汉 430019; 3. 武汉大学 计算机学院, 武汉 430072)

摘 要: 针对现有方法的不足, 提出一种基于 QoS 全局最优的多目标动态 Web 服务选择算法。在给出动态服务组合模型的基础上, 以“抽象服务规划”为输入, 以用户的非功能性需求为全局约束, 将动态服务选择问题转换为一个带 QoS 约束的多目标服务组合优化问题; 利用多目标蚁群算法, 多个目标函数被同时优化并产生一组满足约束条件的 Pareto 优化解。通过运用实验与基于多目标遗传算法的 Web 服务选择算法进行对比, 证明了该方法的可行性和有效性。

关键词: Web 服务; 服务选择; 全局优化; 多目标蚁群算法

中图分类号: TP393; TP301.6

文献标志码: A

文章编号: 1001-3695(2009)12-4442-04

doi:10.3969/j.issn.1001-3695.2009.12.011

Global QoS optimizing and multi-objective Web service selection algorithm

FANG Qi-qing^{1,2}, LIU Qing-hua², PENG Xiao-ming³, HU Ya-hui²

(1. Commanding Communications Academy, Wuhan 430010, China; 2. Air Force Radar Academy, Wuhan 430019, China; School of Computer, Wuhan University, Wuhan 430072, China)

Abstract: To resolve the deficiency of the existing approaches, this paper presented a novel global QoS optimizing dynamic Web service selection algorithm. Firstly, proposed a model of Web service composition. Consequently, with the “abstract service plan” as input and the user non-functional requirements as global constraints, converted the Web service selection problem into a multi-objective optimization problem. Furthermore, based on the multi-objective ant colony algorithm (MOACO), the proposed algorithm simultaneously optimized the different QoS parameters of the workflow. In this way, calculated a set of optimal solutions, known as Pareto set. Experimental results prove the proposed algorithm outperforms the recently published QoS global optimization based on multi-objective genetic algorithm (MOGA), specially designed for solving the Web services selection problem.

Key words: Web service; Web service selection; global optimization; MOACO

动态服务组合^[1]是构建面向服务、松耦合、集成化的应用系统的主要途径。随着越来越多的 Web 服务被共享在网络之上, 服务选择作为动态服务组合的关键技术之一也将面临着如下挑战: a) 为组合服务提供相似功能, 但性能指标各不相同的 Web 服务越来越多, 为服务选择问题增加了复杂度; b) 普通用户无法为组合服务中的每一 Web 服务实例具体给出性能指标, 通常只能给出其对组合服务整体性能所能承受的下限; c) 第三方提供的 Web 服务各项性能参数通常相互对立, 如对于性能和可靠性较高的服务, 往往其费用也较高。现有的研究成果^[2~6]大多基于 QoS 的局部优化, 无法解决上述问题。为此, 如何为组合服务选择全局最优并满足相关用户约束的 Web 服务已成为迫切需要解决的问题。

蚁群算法^[7]最初由意大利学者 Dorigo 于 1991 年首次提出, 是一种用来在图中寻找优化路径的几率型技术, 其灵感来源于蚂蚁在寻找食物过程中发现路径的行为。目前人们对蚁群算法的研究已由当初单一的 TSP 领域渗透到了多个应用领域, 由解决一维静态优化问题发展到解决多维动态组合优化问题, 同时在蚁群算法的模型改进及与其他仿生优化算法的融合

方面也取得了相当丰富的研究成果, 但将蚁群算法运用到动态服务选择的研究成果几乎没有。

本文运用多目标蚁群算法^[8] (MOACO) 解决全局最优的动态 Web 服务选择问题。其基本思路是: 将全局最优动态 Web 服务选择问题转换为一个带 QoS 约束的多目标服务组合优化问题, 通过多目标蚁群算法, 对动态服务选择问题中的多个目标函数同时进行优化, 产生一组满足约束条件的 Pareto 优化解, 最终用户则根据自己的偏好从中进行选择。相关对比实验证明了该方法的可行性和有效性。

1 相关工作

目前, 关于服务动态选择的研究绝大部分是基于 QoS 局部最优的原则, 文献[2~4]对于为组合服务提供相似功能的一组服务, 根据服务的各个 QoS 参数信息进行加权和排序, 并以此为依据分别为组合服务模型中的各个节点选择加权和最大的服务来构建组合服务。在这些方法中, 各个服务节点对应具体服务实例的选择是相互独立的, 而对于基于全局优化的服务选择, 目前的研究成果较少。文献[5, 6]通过把服务组流

收稿日期: 2009-03-11; 修回日期: 2009-05-16 基金项目: 国家“863”计划资助项目(2007AA01Z138)

作者简介: 方其庆(1979-), 男, 安徽无为, 博士研究生, 主要研究方向为动态 Web 服务组合、军事电子信息系统体系结构(fangqiqing@yeah.net); 刘庆华(1966-), 男, 副教授, 硕士, 主要研究方向为装备管理与保障; 彭晓明(1965-), 男, 副教授, 博士研究生, 主要研究方向为动态服务组合、应用系统集成; 胡亚慧(1980-), 女, 讲师, 硕士, 主要研究方向为 Web 个性化。

程的各个 QoS 约束参数线性加权转换为一个单目标函数,利用线性规划的基本原理来解决服务选择的 QoS 全局最优化问题。其存在的问题是:把组合服务中各个 QoS 参数线性加权转换为一个目标函数,因此产生的最优路径是满足约束条件的单目标最优解,不能解决多目标的优化问题,如同时优化组合服务中的执行时间和费用;由于加权和解,其结果不仅对权重向量非常敏感,而且需要用户对问题有一定的认识,如参数的优先级、一个参数对其他参数的影响程度等;产生的优化结果为单个解,用户没有选择的余地;另外,线性规划要求目标函数和约束均是线性的,这在一定程度上限制了算法的实用性。

最近的成果^[10]以组合服务信誉等级和可靠性为约束条件,以执行费用和时间最小为目标函数,基于多目标遗传算法(MOGA)提出了一种全局优化的多目标 Web 服务选择算法。但公开发表的研究成果中尚无运用蚁群算法解决 QoS 全局最优的多目标 Web 服务选择问题。

2 动态服务组合模型

定义 1 用户需求模型(user requirements model)

$$R_{CW} = \langle \text{input}, \text{output}, \text{precondition}, \text{result}, \text{QoS} \rangle$$

其中:input 和 output 为组合服务的输入及输出参数;precondition 为执行该组合服务所应满足的前提条件;effect 为执行组合服务后产生的效果,这四类元素通常简称为 IOPE。基于 IOPE 可以较完整地描述一个组合服务的功能性需求,并且可以借助本体,通过运用相关 AI 规划工具^[11],实现基于功能性约束的 Web 服务自动组合。 $\text{QoS} = \langle q_1, q_2, \dots, q_m \rangle$ 为用户对组合服务的整体非功能性(QoS)需求。其中 q_i 为一个 QoS 参数,如可靠性、执行费用及时间等。

定义 2 Web 服务类型模型(Web service type model)

$$\text{WS}_T = \langle \text{input}, \text{output}, \text{precondition}, \text{result} \rangle$$

一个 Web 服务类型 WS_T 是具备相似功能的一组 Web 服务。

定义 3 Web 服务实例模型(Web service instance model)

$$\text{WS}_I = \langle \text{IOPE}, \text{WS}_T, \text{URL}, \text{QoS} \rangle$$

服务实例是指具有明确 URL 地址的、实际可被调用的 Web 服务。其中: $\text{QoS} = \langle q_1, q_2, \dots, q_m \rangle$; WS_T 为该服务实例所从属的服务类型;URL 为调用该 Web 服务实例的实际 URL 地址。对于具备相同功能性的 Web 服务实例,其 QoS 往往不同。对于某一服务类型 WS_{T_i} 中的某一服务实例 WS_{I_j} 可以显式地给出为 $\text{WS}_{I_j}^T$ 。

定义 4 抽象服务规划模型(abstract service plan)

$$\text{AP} = \langle \text{WS}_{T_1}, \text{WS}_{T_2}, \dots, \text{WS}_{T_n} \rangle$$

抽象服务规划是指由 Web 服务类型构成的、满足相应功能性需求的组合服务序列(不包含具体的服务实例)。

定义 5 具体工作流程模型(concrete workflow model)

$$\text{CW} = \{ \text{WS}_{I_1}^T, \text{WS}_{I_2}^T, \dots, \text{WS}_{I_n}^T \}$$

将抽象服务规划中的每一个 Web 服务类型实例化,进而产生的既满足功能性需求又满足相应非功能性需求的、由具体的 Web 服务实例所构成的组合服务。

通过上述定义,可将一个 Web 服务组合问题定义为

定义 6 Web 服务组合模型(concrete workflow model)

给定一套 Web 服务的类型及该类型下 Web 服务的实例,首先基于用户的功能性需求产生一个基于 Web 服务类型的抽象服务规划;进而以该抽象服务规划为输入,以用户的非功能性需求为全局约束,为该抽象服务规划中的每个服务类型选择具体的服务实例;最终创建出一个既满足功能性需求又满足非功能性需求的具体工作流。

3 Web 服务选择问题定义

基于上述动态服务组合模型,假设已通过相应的 AI 规划工具自动生成,或由相关人员手工编排而成一个满足用户功能性需求的串行抽象服务规划 $\text{AP} = \langle \text{WS}_{T_1}, \text{WS}_{T_2}, \dots, \text{WS}_{T_n} \rangle$ 。其中共包含 n 个服务类型,并且每个 Web 服务类型包含 M 个服务实例。每个服务实例包含三个 QoS 参数, $\text{QoS} = \langle \text{cost}, \text{time}, \text{reliability} \rangle$, cost 为该 Web 服务实例的执行费用, time 为该服务实例的执行时间, reliability 为该服务实例的可靠性。将每个服务类型所包含的服务实例看做是加权有向图中的节点,将服务实例包含的 QoS 参数看做是该节点的权重向量,另外增加一个虚拟起始节点和虚拟结束节点,则服务实例选择图可如下定义:

定义 7 服务实例选择图(service instance selection graph)

服务实例选择图定义为一个加权有向图 $G = \langle V, E, W \rangle$, 如图 1 所示。其中, $V = \text{WS}_{I_1}^T \cup \text{WS}_{I_2}^T \cup \dots \cup \text{WS}_{I_n}^T \cup \{S, T\}$, $\text{WS}_{I_j}^T \in \text{WS}_{T_i} \in \text{AP} (i \leq n, j \leq M)$ 为抽象服务规划 AP 中每个服务类型所包含的服务实例, S 和 T 为虚拟起始节点和虚拟结束节点。链路 E 为相邻 Web 服务类型的 Web 服务实例之间的连接。每个服务节点的权重向量定义为该服务实例所具备的 QoS 参数向量, $W(n) = \{ \text{cost}(n), \text{time}(n), \text{reliability}(n) \}$, 显然 $W(S) = W(T) = 0$ 。

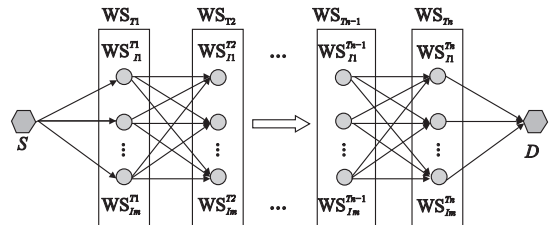


图1 Web服务实例选择图

基于上述 Web 服务实例选择图,一个对应于抽象服务规划 AP 的具体工作流可表示为一条自初始节点 S 至结束节点 D 的路径 $P = P(S, D)$, 根据 cost 和 time 的可加性、reliability 的可乘性,路径 P 的相关参数为

- 执行费用 $\text{cost}(P) = \sum_{n \in P} \text{cost}(n)$;
- 执行时间 $\text{time}(P) = \sum_{n \in P} \text{time}(n)$;
- 可靠性 $\text{reliability}(P) = \prod_{n \in P} \text{reliability}(n)$ 。

假设 P 和 P' 为服务实例选择图中自初始节点 S 到结束节点 D 的两条均满足用户 QoS 的约束路径,令 $\varphi = [C, T, 1/R]$ 为 R_{CW} 中由最终用户给定的 QoS 约束向量,令

$$x = [\text{cost}(P), \text{time}(P), 1/\text{reliability}(P)]$$

$$z = [\text{cost}(P'), \text{time}(P'), 1/\text{reliability}(P')]$$

显然, P 和 P' 均满足

- $x > z (x \text{ 支配 } \varphi)$, 即 $x_i \leq \varphi_i \wedge x_i \neq \varphi_i, \forall i \in \{1, 2, 3\}$;

b) $z > x$ (z 支配 φ), 即 $z_i \leq \varphi_i \wedge z_i \neq \varphi_i, \forall i \in \{1, 2, 3\}$;

并且 P 和 P' 将满足且仅满足如下三个条件之一:

c) $x > z$ (x 支配 z), 如果 $x_i \leq z_i \wedge x_i \neq z_i, \forall i \in \{1, 2, 3\}$;

d) $z > x$ (z 支配 x), 如果 $z_i \leq x_i \wedge z_i \neq x_i, \forall i \in \{1, 2, 3\}$;

e) $x \sim z$ (x 和 z 不可比较), 如果 $x_i > z_i \wedge z_i > x_i, \forall i \in \{1, 2, 3\}$ 。

当 P 对于服务实例选择问题的可行解集 Q 中的任意解 P' 均满足 $P > P' \wedge P > \varphi, P' \in Q$, 称 P 为一个满足用户 QoS 约束的非支配解。整个可行解搜索空间的非支配解集就是 Pareto 优化解集^[12], 记做 P_{true} 。与该优化解集相对应的目标向量在空间上形成的曲面称为 Pareto 前沿面^[12]。

由定义 7 可以进一步把服务组合的服务实例选择 QoS 全局最优化问题转换为一个求解服务实例选择图中从 S 到 D 带 QoS 约束条件的多目标最优路径问题。

定义 8 Web 服务实例选择问题 (Web service instance selection problem)

服务动态选择 QoS 全局最优就是在上述定义的动态服务组合模型执行过程中, 从各个服务类型对应的服务实例中选择具体的 Web 服务实例组成一个可执行的具体工作流, 使得其在满足用户特定 QoS 约束的前提下, 多个目标 (QoS 参数) 达到最优化。对应于 Web 服务实例选择图, 该问题可转换为一个求解服务实例选择图中从 S 到 D 带 QoS 约束条件的多目标最优路径问题, 也即在其可行解搜索空间中求解尽可能逼近 Pareto 前沿, 且均匀分布的非支配解集问题。

基于上述思想和模型, 本文基于多目标蚁群算法设计了求解 QoS 全局最优的多目标 Web 服务选择算法。

4 基于 MOACO 的多目标 Web 服务选择方法

4.1 算法通用过程

文献[8]给出了 MOACO 的通用过程, 该方法的基本思想是: 在蚁群循环过程中设立一个解集合, 解集合初始时随机选择一个服务组合放入解集中。在每只蚂蚁循环结束之后, 首先对所有节点信息素进行一次挥发, 挥发因子为 ρ ; 然后对所得的解与解集合中的每个解进行比较, 如果不被解集中的任何一个解支配, 则将该解写入解集中, 并在该蚂蚁经历过的节点上释放增量信息素 τ_{ij} , 同时删除解集中被该解支配的解。本文基于上述思想, 结合 Web 服务实例选择问题实际, 设计了运用在全局 QoS 最优 Web 服务选择问题上的多目标蚁群算法 (MOACO4WS, 以下简称为 MOACO)。其通用过程如下:

```

procedure MOACO4WS
  初始化  $S, D, N_r, \varphi, \tau_{ij}$ 
  while 不满足循环条件
    repeat-for  $k = 1$  to  $m$ 
      调用 (construct solution  $P$  见 4.3 节)
      if ( $P \notin P_s \mid P_s \in P_{\text{know}}$ ) then
         $P_{\text{know}} = P_{\text{know}} \cup P - \{P_y \mid P \rightarrow P_y\} \forall P_y \in P_{\text{know}}$ 
      end-if
    end-repeat-for
    调用 (Update of  $\tau_{ij}$  (见 4.2 节))
  end-while
end-procedure

```

以上过程中, S, D 分别为虚拟起始和虚拟结束节点, N_r 为当前人工蚂蚁所在节点的相邻点集, φ 为用户 QoS 约束向量。

4.2 信息素更新

算法中的信息素更新采用全局更新方式, 即信息素 τ_{ij} 的更新主要取决于优化解集 P_{know} 的状态。如果 P_{know} 被修改, 则 τ_{ij} 被重新设定为初始值 ($\tau_{ij} = \tau_0 = 0$); 如果 P_{know} 未被修改, τ_{ij} 将按如下过程进行设定:

```

procedure Update of  $\tau_{ij}$ 
  repeat-for every  $P \in P_{\text{know}}$ 
    repeat-for every  $P_{\text{know}}$ 
       $\tau_{ij} = (1 - \rho) * \tau_0 + \rho * \Delta\tau$ 
    end-repeat-for
  end-repeat-for
end-procedure

```

其中: $\rho \in (0, 1]$ 为挥发系数, 其目的是为防止信息素的无限累积; $\Delta\tau$ 表示本次循环中路径 (i, j) 上的信息素增量, 取值为

$$\Delta\tau = \frac{1}{\sum_{P \in P_{\text{know}}} (\sqrt{\text{cost}^2(P) + \text{time}^2(P)} + (1/\text{reliability}(P))^2)}$$

4.3 服务实例选择优化解集构建过程

优化解集构建过程如下:

```

procedure construct solution
  设置禁忌表  $\text{tabuList} = \phi$ ; 设置可行解  $P = \phi$ ;
  int  $i = 0$ ;
  while ( $|P| < n$ ) do
    设置  $N_i$  为当前节点  $i$  未被访问的相邻节点集合;
    if  $N_i \neq \phi$  then
      repeat-for each node  $j \in N_i$ 
        设置节点  $j$  的被选择概率:
         $P_{ij} = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{s \in N_i} [\tau_{is}]^\alpha [\eta_{is}]^\beta}$ ;
        通过随机方法基于概率选择相邻节点  $j$ :
         $P = P \cup \{j\}$ ;
         $\text{tabuList} = \text{tabuList} \cup \{j\}$ ;
        更新参数;
         $i = j$ ;
      end-repeat-for
    else
      重新设置参数;
    end-while
  end-procedure

```

在优化解集构建过程中, 要使组合服务的服务质量指标 cost , time 和 $1/\text{reliability}$ 尽可能地小, 则要求每个服务类集中的 Web 服务的服务质量指标尽可能地小, 即向量 (cost , time , $1/\text{reliability}$) 与 $(0, 0, 0)$ 之间的距离尽可能地小。因此, 受此启发, 令启发函数 $\eta_{ij} = 1/\sqrt{\text{cost}^2(j) + \text{time}^2(j) + (1/\text{reliability}(P))^2}$, 这可使得蚂蚁在选择前进路径的时候更容易向 η_{ij} 大的方向前进。

5 实验结果及讨论

通过实验将本文研究的算法 MOACO 与基于多目标遗传算法^[10] (MOGA) 的 Web 服务选择算法进行对比分析。实验共分四组, 每组采用的服务类型及服务实例规模如表 1 所示 (其中, n 为抽象服务规划中服务类型的个数, M 为每个服务类型所包含的服务实例个数), 每组实验分别运行两次, 第一次运行循环次数设为 100, 第二次运行循环次数设为 200。对于 MOACO 和 MOGA, 实验硬件条件均为 Intel Pentium Dual 2.00 GHz CPU, 2 GB 内存; 软件环境均为操作系统 Windows XP, 算法编译环境均为 VC++6.0。

表 1 各组测试用例中的服务类型及服务实例规模

组	<i>n</i>	<i>M</i>
group1	5	10
group2	10	20
group3	20	20
group4	20	40

为了使得测试数据具备较强的客观性,针对同一 Web 服务类型下不同 Web 服务实例的 QoS 参数,在随机生成的基础之上以高斯分布函数对其进行验证与修改,使得那些运行时间和可靠性较好的 Web 服务实例执行费用相对较高。

用户对具体工作流的整体性约束设置 $\varphi = [C, T, 1/R] = [800, 10, 3]$, 即 $0 \leq \text{cost} \leq 800, 0 < \text{time} \leq 10, 1/3 < \text{reliability} \leq 1$, 即用户对具体工作流提出的费用上限为 800 元, 执行时间上限为 10 s, 可靠性不得低于 1/3。

对于蚁群算法的相关参数设置为 $\alpha = 1.0, \beta = 1.0, \rho = 0.3, Q = 100$, 蚂蚁数目为 30。

对于每组实验, 实验中的比较过程如下:

- a) 对于每组实验数据, MOACO 和 MOGA 各运行五次。
- b) 每个算法通过五次运行, 各生成五组 Pareto 优化解集 $(P_1, P_2, \dots, P_5)$ 。令 $P_{\text{over-ACO}}$ 为 MOACO 生成的五组 Pareto 优化解集相并后的解集, $P_{\text{over-GA}}$ 为 MOGA 生成的五组 Pareto 优化解集相并后的解集。
- c) 通过计算, 分别为 $P_{\text{over-ACO}}$ 和 $P_{\text{over-GA}}$ 去除其中的支配解, 所生成的 Pareto 优化解集分别为 P_{ACO} 和 P_{GA} 。
- d) $\hat{P} = P_{\text{ACO}} \cup P_{\text{GA}}$ 。
- e) 通过计算, 去除 $\hat{P}$ 中的支配解, 得到两个算法的联合 Pareto 优化解集, 记做 P_{apr} 。
- 根据上述过程, 每组实验数据所生成的 P_{apr} 中解的个数如表 2 所示。

表 2 测试生成的 P_{apr} 解的个数

	group1	group2	group3	group4
$ P_{\text{apr}} $	15	24	36	51

为了从多方面进行比较, 令 $[\in P_{\text{apr}}]$ 表示算法产生的 Pareto 优化解集中属于 P_{apr} 的解的个数, $[P_{\text{apr}} >]$ 表示算法产生的 Pareto 优化解集中被 P_{apr} 支配的解的个数, 算法产生的 Pareto 优化解集的个数为 $[|P_{\text{alg}}|]$ 。其中属于 P_{apr} 的非支配解所占的百分比记做 $[\%(\in P_{\text{apr}})]$ 。实验结果如表 3 所示。

表 3 实验结果对比表

组	循环次数	算法	$\in P_{\text{apr}}$	$P_{\text{apr}} >$	P_{alg}	$[\%(\in P_{\text{apr}})]$
group1	100	P_{ACO}	14.5	0	14.5	97%
		P_{GA}	13.8	0	13.8	92%
	200	P_{ACO}	15	0	15	100%
		P_{GA}	15	0	15	100%
group2	100	P_{ACO}	22.1	4.2	26.3	91%
		P_{GA}	17.3	1	18.3	72%
	200	P_{ACO}	23.3	5.1	28.4	97%
		P_{GA}	21	7.3	28.3	88%
group3	100	P_{ACO}	30.1	3.3	33.4	84%
		P_{GA}	26.8	6.4	33.2	74%
	200	P_{ACO}	33	2.1	35.1	92%
		P_{GA}	29.3	7.3	36.6	81%
group4	100	P_{ACO}	41	6.3	47.3	80%
		P_{GA}	33	9.4	42.4	64%
	200	P_{ACO}	45.2	7.1	52.3	89%
		P_{GA}	38.7	12.3	51	76%

通过表 3 可以发现:

a) 当 Web 服务类型和实例数目相对较小 (group1)、循环次数足够大时, 本文的算法 MOACO 和 MOGA 所产生的非支配解集均 100% 属于 P_{apr} 。而当循环次数相对较小时, 本文算法产生的非支配解集较 MOGA 更逼近 Pareto 前沿。

b) 当 Web 服务类型和实例数目逐渐增大 (group2 ~ group4), 本文的算法能够产生更多的属于 P_{apr} 的非支配解。

与此同时, 基于以上四组测试数据, 本文还对两种算法的时间开销进行了对比, 如图 2 所示。本文所提出的算法 MOACO 在执行效率上要高于基于 MOGA 的算法。

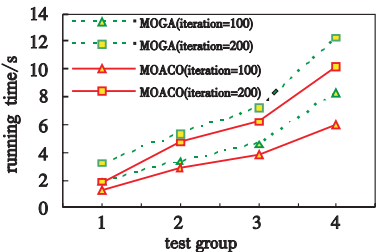


图2 执行时间比较

6 结束语

本文针对现有方法的不足, 基于 MOACO 和串行组合流程, 提出一种基于 QoS 全局最优的多目标动态 Web 服务选择算法, 并通过与基于 MOGA 的同类方法进行比较, 证明了该算法的可行性和有效性。本文下一步的研究工作主要基于两个方面: a) 进一步完善算法, 使其可运用于 WfMC 定义的其他三种基本模型 (并联、选择、循环); b) 在参数选择对优化结果的影响及结果是否分配均匀、逼近 Pareto 前沿方面进行进一步的论证。

参考文献:

[1] BENATALLAH B, DUMAS M, SHENG Q Z, *et al.* Declarative composition and peer-to-peer provisioning of dynamic Web services[C]//Proc of the 18th International Conference on Data Engineering. San Jose: IEEE Computer Society, 2002:297-308.

[2] FASATI, LNICI S, JIN Li-jie, *et al.* eFlow: a platform for developing and managing composition e-services, HPL-2000-36[R]. Palo Alto: HP Laboratories, 2004.

[3] GRAFEN P, ABERER K, HOFFNER Y, *et al.* Cross-low: cross-organizational workflow management in dynamic virtual enterprises[J]. International Journal of Computer Systems Science & Engineering, 2000,15(5):277-290.

[4] LIU Yu-tu, NGU A H, ZENG L Z. QoS computation and policing in dynamic Web services selection[C]//Proc of the 13th International WWW Conference. New York: ACM Press, 2004:66-73.

[5] JORGE C, AMIT S, JOHN M. Quality of service for workflows and Web service processes[J]. Journal of Web Semantics, 2004,1(3):281-338.

[6] AGGARWAL R, VERMA K, MILLER J, *et al.* Constraint driven Web service composition in METEOR-S[C]//Proc of IEEE International Conference on Service Computing. Washington DC: IEEE Computer Society, 2004:23-30.

[7] DORIGO M, CARO G D. The ant colony optimization meta-heuristic[M]//New Ideas in Optimization. Maidenhead: Mc Graw-Hill Ltd, 1999:11-32.

(下转第 4448 页)

函数进行实验。该实验用 MATLAB 6.5 编程, 其中算法的参数如下: 人工鱼群规模为 50, 其感知距离 $visual = 1.5$, 每次移动的步长 $step = 0.3$, 知识解的规模 $p = 0.3N$, $acceptStep = 10$, $N_1 = 15, N_2 = 100$ 。

$$f_1(x) = 4 + 4.5x_1 - 4x_2 + x_1^2 + 2x_2^2 - 2x_1x_2 + x_1^4 - 2x_1^2x_2 - 100 < x_1, x_2 < 100$$

最小值为 $-0.513\ 4$ 。

$$f_2(x) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}, -100 < x_1, x_2 < 100$$

最小值为 0 。

$$f_3(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4, -5 < x_1, x_2 < 5$$

最小值为 $-1.031\ 628\ 5$ 。

$$f_4(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10 - 5 < x_1 < 0, 10 < x_2 < 15$$

最小值为 0.398 。

$$f_5(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \cdot [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] - 2 \leq x_1 \leq 2, -2 \leq x_2 \leq 2$$

最小值为 3 。

$$f_6(x) = x_1^2 + x_2^2 - \cos 18x_1 - \cos 18x_2, -1 \leq x_1 \leq 1, -1 \leq x_2 \leq 1$$

最小值为 -2 。

$$f_7(x) = \{ \sum_{i=1}^5 i \cos[(i+1)x_1 + i] \} \{ \sum_{i=1}^5 i \cos[(i+1)x_2 + i] \} - 10 \leq x_1, x_2 \leq 10$$

最小值为 $-186.730\ 9$ 。

仿真过程中进化代数设为 50, 程序连续运行 10 次所得函数全局最优值、最差值及其平均值作为算法的衡量标准, 仿真结果如表 1 所示。图 3 是根据测试函数 $f_1(x)$ 的实验结果得到的误差曲线图, 图中横坐标为进化代数, 纵坐标为测试函数的目标值与最优值之间的误差。

表 1 CAAFSa 和 AFSA 算法计算结果的比较

测试函数	代数	MATLAB 方法	CAAFSA 算法				AFSA 算法		
		精确值	最优值	最差值	平均值		最优值	最差值	平均值
f_1	50	-0.5134	-0.51340015	-0.51340908	-0.51340602	-0.51340355	-0.51339709	-0.51339412	
f_2	50	0	0.0003852	0.0097159	0.0086271	0.0097159	0.0097235	0.00972	
f_3	50	-1.0316285	-1.0316275	-1.0316069	-1.0316285	-1.0316268	-1.03152025	-1.0315734	
f_4	50	0.398	0.398017	0.397895	0.397939	0.3979101	0.48216323	0.418137	
f_5	50	3	3.000053	3.001385	3.001963	3.0009524	3.0018303	3.0016017	
f_6	50	-2	-1.9994297	-1.877745	-2.08396	-1.875473	-1.712884	-1.859553	
f_7	50	-186.7309	-1.897e+002	-2.249e+002	-1.859e+002	-1.8452e+002	-2.2497e+002	-1.847e+002	

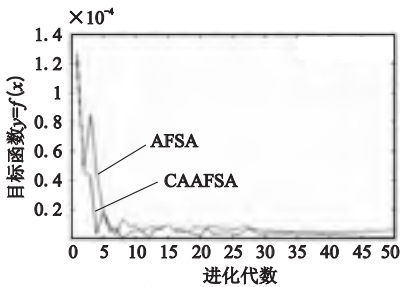


图 3 本文算法与基本AFSA的误差曲线

由表 1 可以看到, 文化人工鱼群算法的结果更加接近于理论最优值, 最差值也更加优化, 整体优化结果明显好于基本人工鱼群。从图 3 中可以看出, 在进化代数相同的条件下, 文化人工鱼群算法的误差更小; 用更少的进化代数得到最优值, 减少了计算量。

5 结 束 语

本文提出一种基于人工鱼群和文化算法的新型混合全局优化算法。通过对数值实例进行数值和结果比较可以看出, 本文算法在最优值、稳定性等方面都取得了不错的效果, 较有效地克服了人工鱼群算法容易陷入局部最优解的缺点, 增加了种群的多样性和全局搜索能力。该文化算法框架利用问题信息指导搜索, 针对不同具体问题可以采用不同内涵的主群体空间和知识空间, 具有较广泛的应用前景。

参 考 文 献:

[1] 李晓磊, 邵之江, 钱积新. 一种基于动物自治体的寻优模式: 鱼群算法[J]. 系统工程理论与实践, 2002, 22(11): 32-38.

[2] REYNOLDS R G. An introduction to cultural algorithms[C]//Proc of the 3rd Annual Conference on Evolutionary Programming. San Diego: World Scientific Press, 1994: 131-139.

[3] JIN X D, REYNOLDS R G. Using knowledge-based system with hierarchical architecture to guide the search of evolutionary computation[J]. International Journal on Artificial Intelligence Tools, 2000, 9(1): 27-44.

[4] REYNOLDS R G, ZHU Shi-nin. Knowledge-based function optimization using fuzzy cultural algorithms with evolutionary programming[J]. IEEE Trans on Systems, Man, and Cybernetics, 2001, 31(1): 1-18.

[5] 艾景波, 腾弘飞. 文化粒子群优化算法[J]. 大连理工大学学报, 2006, 47(4): 539-544.

[6] 李宏, 焦永昌, 张莉. 求解全局优化问题的混合智能算法[J]. 计算机工程与应用, 2006, 42(16): 9-11.

[7] 黄海燕, 顾幸生, 刘漫丹. 求解约束优化问题的文化算法研究[J]. 自动化学报, 2007, 33(10): 1116-1120.

[10] LIU Shu-lei, LIU Yun-xiang, JING Ning, et al. A dynamic Web services selection strategy with QoS global optimization based on multi-objective genetic algorithm[C]//Proc of Conference on Grid and Cooperative Computing. Berlin: Springer, 2005: 84-89.

[11] SIRIN E, PARSIA B, WU Dan, et al. HTN planning for Web service composition using SHOP2[J]. Journal of Web Semantics, 2004, 1(4): 377-396.

[12] Avan VELDHIJZEN D A. Multiobjective evolutionary algorithms: classifications, analyses and new innovations[D]. Kettering: Air Force Institute of Technology, 1999.

(上接第 4445 页)

[8] SCHAERER M, BARA B. A multi-objective ant colony system for vehicle routing problem with time windows[C]//Proc of the IASTED International Conference on Applied Informatics. Innsbruck: IEEE Computer Society, 2003.

[9] GARCI-MARTÍNEZ C, CORDÓN O, HERRERA F. A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the Bi-criteria TSP[C]//Proc of Algorithmic Number Theory Symposium. Berlin: Springer-Verlag, 2004: 116-148.