

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

倪志伟^{1,2}, 方清华^{1,2*}, 李蓉蓉^{1,2}, 李一鸣^{1,2}

(1. 合肥工业大学 管理学院, 合肥 230009; 2. 过程优化与智能决策教育部重点实验室(合肥工业大学), 合肥 230009)

(* 通信作者电子邮箱 qinghua565@163.com)

摘要:为了克服基础蚁群算法存在的前期搜索速度较慢、后期极易陷入局部最优解的缺点,提出初始信息素分布策略和局部优化策略;同时还提出了依赖解的质量的信息素更新依据,以增强算法过程中信息素的有效积累。将该改进蚁群算法应用于基于服务质量(QoS)的 Web 服务组合优化问题中,通过在数据集 QWS2.0 上的实验对改进蚁群算法的可用性和有效性进行了验证。结果表明改进的蚁群算法与基础蚁群算法、利用解与理想解距离更新信息素的改进蚁群算法以及用支配程度作为解的个体评价的改进遗传算法相比,能够找到更多的非劣解,寻优能力更优,表现出了较稳定的性能。

关键词: Web 服务; 服务组合技术; 蚁群算法; Pareto 最优解; 局部优化

中图分类号: TP183 **文献标志码:** A

Improved ant colony optimization for QoS-based Web service composition optimization

NI Zhiwei^{1,2}, FANG Qinghua^{1,2*}, LI Rongrong^{1,2}, LI Yiming^{1,2}

(1. School of Management, Hefei University of Technology, Hefei Anhui 230009, China;

2. Key Laboratory of Process Optimization and Intelligent Decision-making, Ministry of Education (Hefei University of Technology), Hefei Anhui 230009, China)

Abstract: The basic Ant Colony Optimization (ACO) has slow searching speed at prior period and being easy to fall into local optimum at later period. To overcome these shortcomings, the initial pheromone distribution strategy and local optimization strategy were proposed, and a new pheromone updating rule was put forward to strengthen the effective accumulation of pheromone. The improved ACO was used in QoS-based Web service composition optimization problem, and the feasibility and effectiveness of it was verified on QWS2.0 dataset. The experimental results show that, compared with the basic ACO, the improved ACO which updates the pheromone with the distance of the solution and the ideal solution, and the improved genetic algorithm which introduces individual domination strength into the environment selection, the proposed ACO can find more Pareto solutions, and has stronger optimizing capacity and stable performance.

Key words: Web service; service composition technique; Ant Colony Optimization (ACO); Pareto optimal solution; local optimization

0 引言

随着计算机应用的飞速发展,人们的研究重点已从网络层系统互联向应用层服务集成迁移,面向服务的新型计算模式——服务计算(service computing)也应运而生。作为面向服务的体系架构(Service Oriented Architecture, SOA)的一个重要拓展方向,Web 服务技术也随之发展起来,并逐渐成为下一代分布式处理系统的核心,共享在网络上的 Web 服务越来越趋于稳定和易用。但是单个 Web 服务功能较简单且有限,难以满足某些实际应用的需求。因此对多个 Web 服务进行组合,构建功能强大的 Web 服务组合来满足各种实际应用需求变得非常必要。Web 服务组合的理念是将“独立、封装”的服务组合成为更大粒度的服务,实现软件重用、发挥服务的潜力。Web 服务组合提供了一种高度互操作、跨平台和松耦合

的,成为解决困扰 B2B 电子商务、跨组织流程管理和企业应用集成等诸多难题的方法。

目前 Web 服务组合的研究一般是基于服务质量(Quality of Service, QoS)的服务组合,文献[1]已经证明基于 QoS 的服务选择是 NP(Non-deterministic Polynomial)难问题。启发式算法在求解 NP 难问题时表现出了优良的性能,因而出现了大量利用启发式算法求解 Web 服务组合问题的研究成果,如采用遗传算法(Genetic Algorithm, GA)^[2-3]、粒子群优化算法(Particle Swarm Optimization Algorithm, PSO)^[4-5]和蚁群算法(Ant Colony Optimization, ACO)^[6]等。

文献[7]提出了一种 QoS 感知的动态 Web 服务组合优化算法,其本质是将具有 QoS 全局最优 Web 服务选择的问题转化为一个 QoS 感知的多目标多选择 Web 服务组合优化问题,最终目的是为了找到帕累托最优解。文献[8]提出了一种多

收稿日期:2015-03-16;修回日期:2015-05-18。

基金项目:国家自然科学基金资助项目(71271071, 71490725);国家自然科学基金青年项目(71301041);国家 863 计划项目(2011AA040501)。

作者简介:倪志伟(1963-),男,安徽合肥人,教授,博士生导师,主要研究方向:人工智能、机器学习;方清华(1990-),女(壮族),广西南宁人,硕士研究生,主要研究方向:数据挖掘、群体智能;李蓉蓉(1990-),女(回族),福建泉州人,硕士研究生,主要研究方向:群智能算法、云计算;李一鸣(1990-),男,山东聊城人,硕士研究生,主要研究方向:数据挖掘、机器学习。

目标遗传算法 SMOGA (Simulation-based Multi-objective Optimization with Genetic Algorithm), 将个体的支配强度和被支配强度结合, 作为个体的评价, 根据评价结果进行环境选择及生成个体的交叉概率, 并提出了结合局部搜索的个体变异策略。SMOGA 提高了区分个体效用的精度和解集的优化效果。文献[9]把蚁群算法引入 Web 服务选择领域, 将基于 QoS 的 Web 服务选择问题转化为最优路径选择问题, 并给出了蚁群算法求解 Web 服务选择问题的模型, 测试了蚁群算法对于解决 Web 服务选择问题的有效性。文献[10]提出了结合蚁群算法和遗传算法的混合 Web 服务组合求解算法。该混合算法将 Web 服务组合的全局优化问题转化为寻找带 QoS 约束的加权有向无环图的最优路径问题, 利用改进蚁群算法找到问题的非支配解集; 同时将非支配解集作为改进遗传算法的初始种群, 利用遗传算法找到最优解。文献[11]阐述了解空间的理想解的概念, 利用解与理想解的差距来指导信息素的更新, 并通过实验验证了其改进的蚁群算法能够在巨大搜索空间找到接近最优的解, 支持非常复杂的 Web 服务组合。

虽然上述研究都取得了不错的效果, 但也都无法避免算法初期由于缺乏对搜索空间的知识, 求解速度较慢, 或者容易陷入局部最优的问题。针对求解速度慢这个问题, 本文提出了两个改进策略: 初始信息素分布策略和信息素更新依据。而针对算法迭代后期容易陷入局部最优解的问题, 本文提出了局部优化策略, 并通过两个实验对算法的性能进行了验证。实验结果表明改进后的蚁群算法在寻优能力优于基本蚁群算法、文献[11]提出的改进蚁群算法和文献[8]提出的改进遗传算法, 表现出了较优的性能。

1 基于 QoS 的 Web 服务组合

1.1 相关概念

Web 服务是一种新型的在线应用服务, 由统一资源标识符 (Uniform Resource Identifier, URI) 标识的软件应用程序, 其接口和绑定可以通过 XML 标准进行定义、描述和发现, 通过 Internet 协议与其他软件应用程序直接进行交互^[12]。

具体服务 (Concrete Service, CS): 由服务提供者在统一描述、发现和集成 (Universal Description, Discovery and Integration, UDDI) 注册中心中注册的, 可以执行的 Web 服务。一个 CS 是一个三元组 $\langle S, G, P \rangle$, 其中: S 为基本描述, 即服务名称和文本描述; G 是服务功能描述, 即服务功能、行为描述; P 是属性描述, 主要包括费用 (cost)、时间 (time) 等非功能属性^[13]。

抽象服务 (Abstract Service, AS): 仅包含功能描述和接口信息, 是构成业务流程的基本逻辑单元。一个 AS 包含多个 CS, 这些 CS 可由不同服务提供者提供, 具有不同的 QoS 值, 但都具有相同的功能和调用接口。每一类 AS 包含 CS 的数量可以不同。

业务流程 (Business Processes, BP): 根据用户的功能需求进行分析建模和验证, 构造出符合用户需求的业务流程。业务流程由多个 AS 之间连接而成, 抽象服务之间有顺序、选择、并行和循环等控制关系。业务流程示意图如图 1 所示。

假设一个 BP 由 N 个 AS 组成 $\{AS_1, AS_2, \dots, AS_N\}$, 而每

个 AS 有 M 个候选功能相同的 CS, 所有候选 CS 为: $\{CS_{11}, CS_{12}, \dots, CS_{1M}, CS_{21}, CS_{22}, \dots, CS_{2M}, \dots, CS_{N1}, CS_{N2}, \dots, CS_{NM}\}$, 一个 CS 有 Q 个 QoS 属性 $\langle QoS_1, QoS_2, \dots, QoS_Q \rangle$ 。Web 服务组合就是为业务流程的每一个抽象服务选出一个具体服务, 使业务流程的整体评价达到最优, 选出的 CS 集合称为组合服务 SC, 如图 2 所示。这样, 各个小粒度的 Web 服务之间相互通信和协作, 可以实现大粒度的服务功能, 服务开发者可以借此满足各种复杂的功能需求。

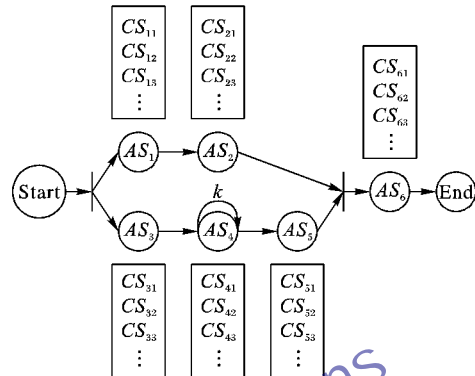


图1 业务流程示意图

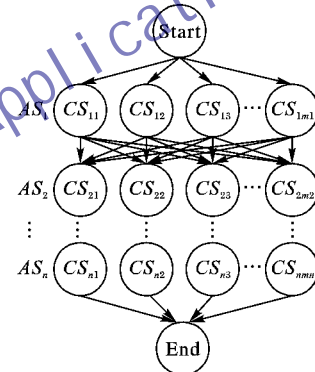


图2 Web服务组合流程

1.2 Web 服务的质量评价指标

Web 服务组合就是要在满足用户功能需求的前提下, 选出 QoS 尽量优的组合服务。满足功能性需求是基本任务, 而服务的非功能性属性是辨别 QoS 优劣的重要指标, 对用户来说最重要的其实是非功能属性 QoS。QoS 描述的是产品或服务满足消费者需求的能力, 可以从并发处理能力、吞吐量、安全性等方面来描述^[14]。出于实用性和可测量性的考虑, 本文主要从响应时间 T 、服务成本价格 C 、可用性 A 、可靠性 R 和信誉度 Re 五个方面来考虑 Web 服务的 QoS。这也是目前较常用、较具有代表性的 QoS 属性^[15]。

1.3 Web 服务组合优化问题的转化

基于 QoS 的 Web 服务选择问题是指如何从各抽象服务的候选服务集中分别选出一个具体服务, 使得选中的这组服务能在满足用户对组合服务的所有约束的前提下, 使组合服务的整体 QoS 最高。

目前, 国内外很多学者已经从不同的角度将该问题转换为各种已知的数学问题, 分别有: 整数、线性规划问题, 多选择背包问题, 单目标组合优化问题, 多属性决策问题和多目标组合优化问题等, 也在此基础上构造模型以设计相应的选择算法^[16]。

然而,服务的 QoS 属性之间存在不可公度性和矛盾性,很难将其值规范到统一的度量空间,且用户缺乏相关领域的知识,难以确切地给出权重信息,因此难以准确地评估出服务的综合值,所以基于 QoS 的服务组合问题一般不存在通常意义上的最优解,讨论的多是非劣解。

解决多目标问题的一般手段是通过对各个目标进行权衡和折中处理,得到不劣于其他解的一个解集,称为非劣解集(Pareto 解集)^[17]。由此,下面给出 Pareto 支配及 Pareto 最优解集的定义。

定义 1 Pareto 支配。当且仅当以下 2 个条件都成立时,称解 x_1 支配解 x_2 (目标为最小化): 1) $f_k(x_1) \leq f_k(x_2)$ ($k = 1, 2, \dots, QN$), 即为 x_1 的各个目标都不比 x_2 差; 2) $\exists k \in \{1, 2, \dots, QN\} : f_k(x_1) < f_k(x_2)$, 即至少存在一个目标函数 $f_k(x)$, 使得 x_1 的目标值比 x_2 的目标值好。

定义 2 Pareto 最优解集。在一个可行解的集合 X 中, 那些不被 X 中任何一个解所支配的解的集合 X_p ($X_p \subset X$), 称为 Pareto 最优解集。

2 求解 Web 服务组合的改进蚁群算法

2.1 基本蚁群算法

蚁群算法(ACO)^[18]采用分布式并行计算机制,具有较强的鲁棒性,通过蚂蚁之间的协作机制来实现对多目标优化问题最优解的搜索,相比传统的穷举算法、贪婪算法,能够更快地找到问题的最优解。

蚂蚁的移动过程是蚁群算法解决 Web 服务组合问题的核心,利用蚁群算法求解 Web 服务组合优化模型中,蚂蚁每到达一个抽象服务下的 CS,需要计算当前 CS 与下一个 AS 所包含的所有候选 CS 的转移概率,概率计算公式由式(1)确定,用轮赌法确定蚂蚁的方向。

$$P_{(i,a)(j,b)} = \frac{[\tau_{(i,a)(j,b)}]^\alpha [\eta_{(j,b)}]^\beta}{\sum_{b=1}^M [\tau_{(i,a)(j,b)}]^\alpha [\eta_{(j,b)}]^\beta} \quad (1)$$

其中: $P_{(i,a)(j,b)}$ 表示当前选择了第 i 个 AS 的第 a 个 CS, 下一步选择 CS_{jb} 的选择概率; $\tau_{(i,a)(j,b)}$ 为路径 CS_{ia} 到 CS_{jb} 的信息素, $\eta_{(j,b)}$ 为 CS_{jb} 的启发信息; α, β 为信息素和启发函数的相对重要程度。

信息素的挥发与释放是蚁群算法正反馈机制的关键。基础蚁群算法采用式(2)和(3)的规则进行信息素的更新:

$$\tau_{(i,a)(j,b)}(t+n) = (1-\rho)\tau_{(i,a)(j,b)}(t) + \Delta\tau_{(i,a)(j,b)}; 0 < \rho < 1 \quad (2)$$

$$\Delta\tau_{(i,a)(j,b)} = Q/L_{sci} \quad (3)$$

文献[18]提出了三种不同的信息素更新策略,本文采用的是其中最方便,也是应用最广泛的 Ant-cycle system 模型。模型中 L_{sci} 的计算规则如式(4):

$$L_{sci} = \sum_{i=0}^{N-1} \sqrt{(T_{i+1} - T_i)^2 + (C_{i+1} - C_i)^2 + \dots + (Re_{i+1} - Re_i)^2} \quad (4)$$

其中: ρ 为信息素挥发因子; Q 为常数; C_i 为组合服务中第 i 个 Web 服务的成本值, R_i 为可靠性,以此类推。

启发函数的形式也有多种,本文基础蚁群算法的实现采

用文献[9]中提出的方法,计算公式如式(5):

$$\eta_{(j,b)} = \frac{1}{\sqrt{T_{j,b}^2 + C_{j,b}^2 + \dots + Re_{j,b}^2}} \quad (5)$$

其中: $i, j \in [1, N]; a, b \in [1, M]; k \in [1, QN]$ 。

2.2 改进蚁群算法

蚁群算法具有自组织、正反馈性等优点,且具有很强的发现较好解的能力,易于与其他方法相结合,但是算法初期缺乏对搜索空间的知识,信息素积累的过程比较漫长,求解速度较慢。在求解多目标优化问题时,蚂蚁依赖解空间中信息素积累来搜寻的方式,会使得解都集中在解空间的某一区域内,不利于发展群体的多样性,容易陷入局部最优解,使搜索停滞。针对这些问题,本文提出了改进的蚁群算法。

2.2.1 信息素更新依据

基础蚁群算法求解 Web 服务组合问题时,更新信息素利用的是解中每两个 CS 相应 QoS 值的差值信息 L , L 值越小,该解释放的信息素越多。如果一个解的每个 CS 的 QoS 值都是较差的,其值较接近,解的 L 值也是小的。这样的相对距离比较方式没有考虑到解本身的优劣,因此寻优效果较差。对此,文献[11]中提出了理想最优解的定义:一个理想解向量 f_b 是解空间中包含每个单目标最优解的点,这个理想解可能并不能达到,它是作为算法试图实现的一个目标解。简言之,越接近理想解的解就越好,理想解可以由计算得出。

受此启发,本文用同样的方法计算了解空间的最差解 f_w , 越接近最差解的解就越差,并定义 d_b, d_w 分别为一个解与理想解、最差解的欧氏距离,计算公式分别为式(6)、(7),利用它们的比值 $D = d_b/d_w$ 来指导信息素的更新,即信息素的更新规则按式(8)进行。

$$d_b = \sqrt{\sum_{k=1}^{QN} [f(k) - f_b(k)]^2} \quad (6)$$

$$d_w = \sqrt{\sum_{k=1}^{QN} [f(k) - f_w(k)]^2} \quad (7)$$

$$\Delta\tau_{(i,a)(j,b)} = Q/D \quad (8)$$

并且本文算法在执行迭代的过程中,只有能添加到 Pareto 解集的解才能对信息素进行更新操作。

计算理想解与最差解的算法描述如下:

算法 1 CalFF(计算理想解与最差解的算法)。

输入 全部 CS 的 QoS 值数据;

输出 解空间的理想解 f_b 和最差解 f_w 。

1) For $k = 1$ to QN

2) For $i = 1$ to N

3) 计算第 i 个 AS 所包含所有 CS 的 k 属性值的最大、最小值: $(k_{i \max}, k_{i \min})$;

4) End For

5) 根据 k 属性的聚合计算方式,将 N 个 $k_{i \min}$ 相加或相乘,得到 $f_b(k)$; 将 N 个 $k_{i \max}$ 相加或相乘,得到 $f_w(k)$;

6) End For

2.2.2 初始信息素释放策略

基础蚁群算法初期收敛较慢,是因为基础蚁群的初始信息素值一般设成一样的数值,没有加以区分,因此要经过多次迭代,各路径信息素才会产生落差,加速蚁群算法的收敛。针对这个问题,本文对初始信息素的分布设计了如下的算法:随机生成 PN 个解,并用它们更新 Pareto 解集,最后用 Pareto 解

集里的解来释放信息素,作为信息素的初始分布。

算法2 PreTao(初始信息素释放算法)。

输入 随机解个数 PN ;

输出 初始信息素 Tao 。

- 1) For $p = 1$ to PN
- 2) 产生一个随机解 scp ;
- 3) ChangeParetoList($scp, PList$);
- 4) End For
- 5) For each sci in $PList$
- 6) 用 sci 更新信息素;
- 7) End For

2.2.3 局部优化策略

针对基础蚁群算法容易陷入局部最优的问题,本文借鉴遗传算法变异操作的思想,提出了蚁群算法的局部优化策略。该策略的具体做法为:随机复制 Pareto 解集中的一个解,随机选择解的一位进行随机变异,形成新解;然后尝试用新解更新 Pareto 解集。

算法3 Var(局部优化算法)。

输入 信息素 Tao , Pareto 解集 $PList$;

输出 信息素 Tao , Pareto 解集 $PList$ 。

- 1) 随机从 $PList$ 中抽取一个解 scp ;
- 2) 生成一个0至 N 的随机数 rN , 一个0至 M 的随机数 rM ;
- 3) If rM 与解 scp 的 rN 位的值相同
- 4) 重新生成一个0至 M 的随机数 rM ;
- 5) Else
- 6) 将解 scp 的 rN 位的值变为 rM , 成为新解 scr ;
- 7) End If
- 8) $isin = \text{ChangeParetoList}(scr, PList)$;
- 9) If $isin == \text{true}$
- 10) 用 scr 更新信息素 Tao ;
- 11) End If

Pareto 解集更新函数的具体实现步骤如下:

算法4 ChangeParetoList(非劣解集更新算法)。

输入 解 sca , Pareto 解集 $PList$;

输出 Pareto 解集 $PList$, 解插入成功标签 $IsInsert$ 。

- 1) $IsGood = \text{true}; IsInsert = \text{false}$
- 2) If $PList.Length == 0$
- 3) 将 sca 插入 $PList$; $IsGood = \text{true}; IsInsert = \text{false}$;
- 4) End If
- 5) For each sci in $PList$
- 6) If sci 支配 sca
- 7) $IsGood = \text{false}; \text{break}$;
- 8) End If
- 9) If sca 支配 sci
- 10) 将 sci 从 $PList$ 中删除; continue;
- 11) End If
- 12) End For
- 13) If $IsGood == \text{true}$
- 14) 将 sca 插入 $PList$; $IsInsert = \text{false}$;
- 15) End If
- 16) return $IsInsert$;

2.3 求解 Web 服务组合问题的改进蚁群算法的性能分析

2.3.1 算法的实现步骤

步骤1 算法初始化。初始化各个参数以及 Pareto 解集,初始化蚂蚁种群,将蚂蚁置于起始 S 节点。

步骤2 初始信息素释放策略。用随机产生的 PN 个解

更新 Pareto 解集,用 Pareto 解集中的解释放初始信息素。令 $NC = 1$, 迭代开始。

步骤3 为每一只蚂蚁计算当前 CS 到下一个 AS 的所有 CS 的选择概率,用轮赌法选择 CS,将其序号加入蚂蚁的路径 $List$, 聚合其 QoS 值,直到蚂蚁得到一个完整的解。用该解尝试更新 Pareto 解集。

步骤4 信息素的更新。每段路径执行信息素的挥发操作。在本次迭代中,能成功更新 Pareto 解集的解,可以为经过的路径上释放相应信息素。

步骤5 局部优化策略。随机复制 Pareto 解集中的一个解,进行单位变异,形成新解,尝试用其更新 Pareto 解集。若更新成功,用其更新信息素。

步骤6 $NC = NC + 1$ 。若 $NC > NC_{\max}$, 结束循环,输出 Pareto 解集中的解;否则,返回步骤3,继续迭代。

2.3.2 算法的时间复杂度分析

各种群体算法中在实际应用中通常采用时间复杂度估算算法执行效率的高低。在算法的设计和分析中,仍然采用实用性的复杂度概念,把求解时间的关键操作(如加、减、乘、除、比较等运算)定义为基本操作,一般把算法执行基本操作的次数视为算法的时间复杂度。

求解 Web 服务组合问题的改进蚁群算法中,蚂蚁种群规模为 AN , 根据算法实现步骤分析算法的时间复杂度:

初始化其他参数需要常数次,初始化蚂蚁种群需要 AN 次,时间复杂度为 $O(AN)$; 产生 PN 个初始解,尝试更新 Pareto 解集,最少比较一次,最多比较 $(PN * (PN - 1)) / 2$ 次。因此,算法初始化部分的时间复杂度为 $O(AN) + O(PN^2)$ 。

寻优过程中,每只蚂蚁找到一个解,都要与 Pareto 解集中所有的解进行比较,算法总共迭代 NC_{\max} 次,时间复杂度为 $O(NC_{\max} * (AN^2 + AN^3))$ 。

综上所述:改进蚁群算法经过 NC_{\max} 次迭代后,整个算法的时间复杂度为: $O(AN + PN^2 + NC_{\max} * (AN^2 + AN^3))$ 。

3 实验及结果分析

为了提高数据的可信度,本文实验采用的是 QWS 2.0 数据集,它是 Eyhab Al Masri 博士于 2008 年收集的真實数据集^[19],包含了 2507 个正在互联网中提供服务的 Web 服务。该数据集的服务包含 11 个服务属性,本文实验抽取其中 5 个:响应时间、可用性、执行成功率、可靠性和网络延时。本实验选取抽象服务 AS 个数 N 为 40, 每个 AS 包含的具体服务 CS 个数 M 为 40, 即从 QWS2.0 中抽取 1600 个服务的 QoS 值作为实验数据。

为了验证本文算法的可行性及实用性,利用 Java 和 MyEclipse 软件编写相关代码。本实验编译运行的计算机参数为:32 位 Windows 7 操作系统, Intel Core 2 E8400 3.00 GHz CPU, 2.00 GB 内存。共设置两个实验,根据文献^[20]得出的蚁群算法相关参数范围,本文选择的参数值分别为: $\alpha = 1$, $\beta = 5$, $\rho = 0.3$, $\tau_0 = 10$, $Q = 10$, $P_0 = 0.8$ 。

3.1 本文算法其他参数的确定

3.1.1 对参数 PN 的分析

图3为其他参数不变,只改变 PN 的取值,各独立运行 20 次所得到的结果($AN = 80$, $NC_{\max} = 200$)。从图中可以看出,

随着 PN 的增大,算法找到的解的平均个数有较平缓的增加趋势,但收敛时间的增加趋势较快。即算法初始时,随机产生的解越多,算法收敛得越快。但是随着 PN 的增大,算法耗时也将大大地增加。因此本文折中选择 $PN = 200$ 。

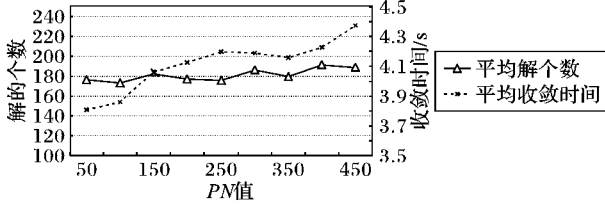


图3 解的个数及收敛时间随 PN 值的变化

3.1.2 对参数 NC_{\max} 和 AN 的分析

图4、5分别为其他参数不变,只改变 NC_{\max} 和 AN 的取值,各独立运行20次所得到的结果。从图中可以看出,随着 NC_{\max} 和 AN 的增大,算法找到的解的平均个数不断增加,收敛时间也在渐渐增加,当 $NC_{\max} = 200, AN = 80$ 时,平均解个数增加的幅度开始减少,因此本文取 $NC_{\max} = 200, AN = 80$ 。

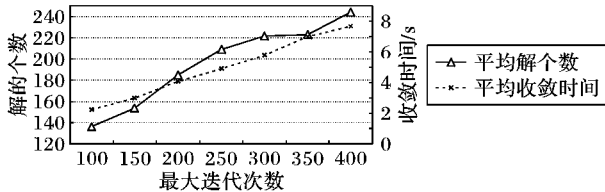


图4 解的个数及收敛时间随 NC_{\max} 值的变化

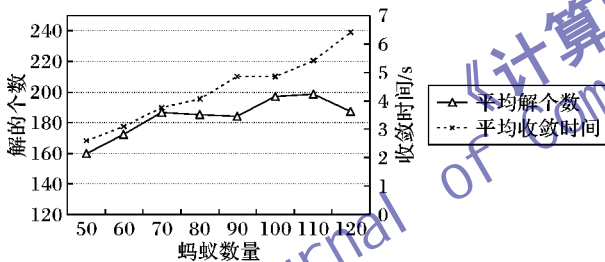


图5 解的个数及收敛时间随 AN 值的变化

3.2 多目标实验

根据3.1节的分析,本节实验的其他相关参数取值分别为:蚂蚁数量 $AN = 80, NC_{\max} = 200, PN = 200$ 。

本文实验分别与基础蚁群算法、文献[11]提出的改进蚁群算法、文献[8]提出的改进遗传算法、只含A改进策略(信息素更新依据)的蚁群算法(方法1)、含有A和B改进策略(初始信息素分布策略)的蚁群算法(方法2)以及含有A和C改进策略(局部优化策略)的蚁群算法(方法3)进行对比(为表示方便,在各实验结果图中分别用字母a~g表示这些对比算法)。各算法各独立运行20次,得到如图6~8所示的结果,分别考察算法求解得到的解的平均数、得到的最多解、平均收敛代数、最小收敛代数、平均收敛时间和最小收敛时间(单位:ms)。

由图6~8可知,在求解Web服务组合问题时,与其他对比算法相比,本文算法能在最小的收敛代数及较短的收敛时间内找到更多的非劣解;遗传算法的时间复杂度比蚁群算法小,因此收敛时间较短。结果表明,本文算法不仅寻找非劣解的能力较优,效率也较高,是比较实用和高效率的算法。

由图6知,方法3在寻找非劣解的能力比方法1和方法2

好一些,而在收敛代数与收敛时间上三个方法均相差不大,说明改进策略C在算法的寻优中发挥了较大的作用。

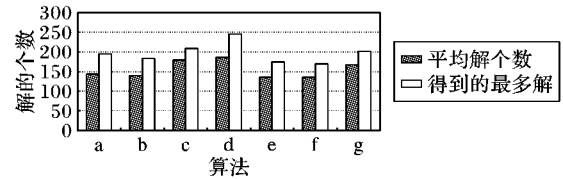


图6 各算法求得解的对比

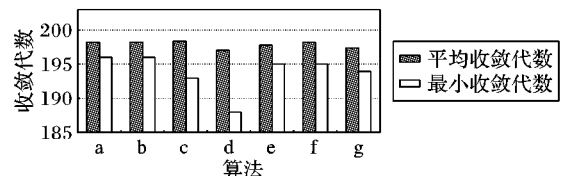


图7 各算法求解过程中收敛代数对比

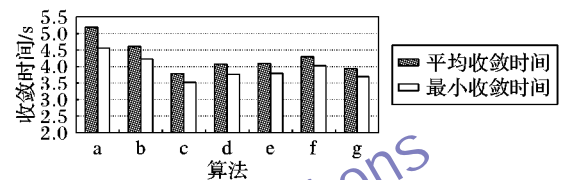


图8 各算法求解过程中收敛时间对比

为验证本文改进蚁群算法在不同问题规模情况下的求解情况,设置9组规模分别为 $10 \times 10, 15 \times 15, 20 \times 20, 25 \times 25, 30 \times 30, 35 \times 35, 40 \times 40, 45 \times 45$ 和 50×50 (为表示方便,在各实验图中分别用10,15,...,50表示),算法参数设置不变,分别独立运行20次得到的结果如图9所示。

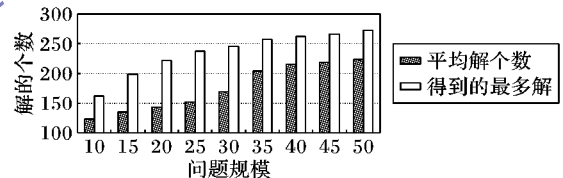


图9 不同规模下本文算法寻优情况

图10为本文改进蚁群算法在9组不同规模情况下寻优过程的收敛情况。可以看出,随着问题规模的增加,算法找到的解越多,收敛时间也越长;规模较小时,解的增加速率比较稳定,但当规模达到 40×40 及以上时,解的增加速度有所下降,可能是QWS数据集中,排序靠后的Web服务的QoS的整体水平较差的原因。随着规模的增大,算法的收敛时间增加的速率比较稳定。由此说明,本文算法在不同问题规模下也能有效地找到更多的解,算法性能比较稳定。

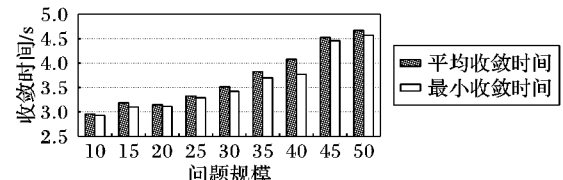


图10 不同规模下本文算法收敛情况

3.3 单目标实验

为了验证算法能找到较优的解,本文还进行了单目标实验,给定5个QoS属性的权重,分别为响应时间0.3,网络延时0.1,可靠性0.2,成功率0.2,可用性0.2,其他参数设置与3.1节一致。将Web服务组合优化问题转化为最小化的单目标优化问题。

各对比算法分别独立运行20次,得到20个最优解,求其平均值得到平均解,而20个最优解中最优的为最终最优解。各算法的平均解和最终最优解如图11所示,可以看出,本文算法所得平均解与最终最优解都是对比算法中最优的,也就是说,本文算法找到的最优解的质量要优于其他对比算法。

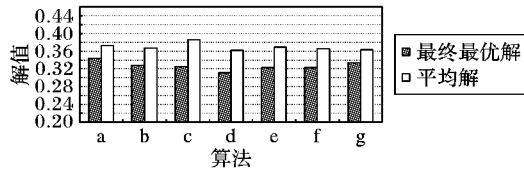


图11 单目标实验中各算法找到的解的情况

图12、13所示分别为各算法所得的最终最优解对应的组合服务的响应时间和网络延时。

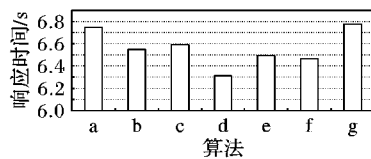


图12 各算法最终最优解对应组合服务的响应时间

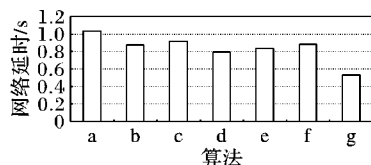


图13 各算法最终最优解对应组合服务的网络延时

分析整体实验结果可知,在求解Web服务组合优化问题时,本文算法在寻找非劣解以及最优解方面体现了较优的性能,能够较快从所有候选服务集中得到一个优质的组合服务。

4 结语

现有的服务组合优选方法多是基于效用函数的评价,返回单个效用最优的组合方案。在QoS维度较多时,难以精确给出每个维度的偏好权重,使这些方法的可用性和有效性大大降低。而本文改进蚁群算法能够对较大规模服务组合问题进行求解,最终得到解空间的一个非劣解集,为决策者提供有效的参考。但是本文对Web服务组合过程中Web服务的动态性、不稳定性等问题还没有进行讨论,还需要进一步的研究。下一步我们将在更多数据集中验证改进算法的性能,并对算法参数的优化选择以及算法的收敛问题给予研究,并进一步寻找制约蚁群算法收敛速度的深层原因以及算法对数据敏感性的规律。

参考文献:

- [1] CANFORA G, PENTA M D, ESPESITO R, *et al.* A lightweight approach for QoS-aware service composition [C]// ICSOC'04: Proceedings of the 2nd International Conference on Service Oriented Computing. New York: ACM, 2004: 36-47.
- [2] LIU S, LIU Y, ZHANG F, *et al.* A dynamic Web services selection algorithm with QoS global optimal in Web services composition [J]. Journal of Software, 2007, 18(3): 646-656. (刘书雷, 刘云翔, 张帆, 等. 一种服务聚合中QoS全局最优服务动态选择算法[J]. 软件学报, 2007, 18(3): 646-656.)
- [3] ZHANG C, SU S, CHEN J. Genetical gorithm on Web services selection supporting QoS [J]. Chinese Journal of Computers, 2006, 29(7): 1029-1037. (张成文, 苏森, 陈俊亮. 基于遗传算法的QoS感知的Web服务选择[J]. 计算机学报, 2006, 29(7): 1029-1037.)
- [4] SHENG G, WEN T, GUO Q, *et al.* Web service composition based on modified particle swarm optimization [J]. Chinese Journal of Computers, 2013, 36(5): 1031-1045. (盛国军, 温涛, 郭权, 等. 基于改进粒子群的Web服务组合[J]. 计算机学报, 2013, 36(5): 1031-1045.)
- [5] XIA H, LI Z. A particle swarm optimization algorithm for service selection problem based on quality of service in Web services composition [J]. Journal of Beijing University of Posts and Telecommunications, 2009, 32(4): 63-67. (夏虹, 李增智. 粒子群算法求解Web服务组合中基于QoS的服务选择[J]. 北京邮电大学学报, 2009, 32(4): 63-67.)
- [6] XIA Y, CHENG B, CHEN J, *et al.* Optimizaing services composition based on improved ant colony algorithm [J]. Chinese Journal of Computers, 2012, 35(2): 270-281. (夏亚梅, 程渤, 陈俊亮, 等. 基于改进蚁群算法的服务组合优化[J]. 计算机学报, 2012, 35(2): 270-281.)
- [7] ZHAO S, WANG L, MA L, *et al.* An improved ant colony optimization algorithm for QoS-aware dynamic Web service composition [C]// ICICEE 2012: Proceedings of the 2012 International Conference on Industrial Control and Electronics Engineering. Washington, DC: IEEE Computer Society, 2012: 1998-2001.
- [8] LIU L, YANG D. Multi-objective genetic optimization algorithm for SLA-aware service composition problem [J]. Journal of Jilin University: Engineering and Technology Edition, 2015, 45(1): 267-273. (刘磊, 杨冬. 求解服务等级感知服务组合问题的多目标遗传算法[J]. 吉林大学学报: 工学版, 2015, 45(1): 267-273.)
- [9] WANG X, MA L. Web service selection based on ant colony algorithm [J]. Modern Electronics Technique, 2013, 36(12): 9-11. (王秀亭, 马力. 基于蚁群算法的Web服务选择[J]. 现代电子技术, 2013, 36(12): 9-11.)
- [10] ZHAO C, WANG J, QIN J, *et al.* A hybrid algorithm combining ant colony algorithm and genetic algorithm for dynamic Web service composition [J]. The Open Cybernetics & Systemics Journal, 2014, 8: 146-154.
- [11] ZHANG W, CHANG C K, FENG T, *et al.* QoS-based dynamic Web service composition with ant colony optimization [C]// AC-SAC 2010: Proceedings of the IEEE 34th Annual Computer Software and Applications Conference. Piscataway: IEEE, 2010: 493-502.
- [12] DAI Y, YANG L, ZHANG B, *et al.* QoS for composite Web services and optimizing [J]. Chinese Journal of Computers, 2006, 29(7): 1167-1178. (代钰, 杨雷, 张斌, 等. 支持组合服务选取的QoS模型及优化求解[J]. 计算机学报, 2006, 29(7): 1167-1178.)
- [13] KONG W, LIU Q, YANG Z, *et al.* Composition of Web services based on two-dimension QoS model [J]. Computer Science, 2008, 35(11): 131-134. (孔维梁, 刘清堂, 杨宗凯, 等. 基于二维QoS模型的Web服务组合[J]. 计算机科学, 2008, 35(11): 131-134.)

工作放在 Intel 核芯显卡并行模块执行,其 CPU 占用率仅为 3.5%。

4 结语

本文提出了一种快速准确的背景建模方法,该方法采用了 USILT 编码及多颜色通道背景建模,并使用了 Intel HD 核芯显卡进行并行计算。在算法设计中提出了 USILT 纹理特征算子,通过压缩编码空间,降低直方图维度,从而降低了算法复杂度;此外,使用多颜色通道信息取代灰度信息,有效地融合颜色信息,改善了前景分割效果。另一方面,设计了在 Intel HD 核芯显卡上利用 OpenCL 技术进行并行计算的背景建模方法。实验结果表明,本文所提出的算法能取得更好的前景分割效果,并且提高了处理速度,降低了 CPU 占用率以及内存开销,因此更适用于大规模视频数据处理的实际需求。

参考文献:

- [1] BARRON J L, FLEET D J, BEAUCHEMIN S S. Performance of optical flow techniques [J]. *International Journal of Computer Vision*, 1994, 12(1): 43–77.
- [2] BANH N D, LO T K, HOLTHAUS K D, *et al.* Moving target detection method using two-frame subtraction and a two quadrant multiplier: US Patent 5150426 [P]. 1992-09-22.
- [3] ELGAMMAL A, HARWOOD D, DAVIS L. Non-parametric model for background subtraction [C]// *ECCV 2000: Proceedings of the 6th European Conference on Computer Vision*, LNCS 1843. Berlin: Springer, 2000: 751–767.
- [4] STAUFFER C, GRIMSON W E L. Adaptive background mixture models for real-time tracking [C]// *CVPR 1999: Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington, DC: IEEE Computer Society, 1999: Article No. 2.
- [5] WANG D, CHENG Y, YANG T, *et al.* Moving cast shadow suppression from a Gaussian mixture shadow model [J]. *Journal of Computer Applications*, 2006, 26(5): 1021–1023, 1026. (王典, 程咏梅, 杨涛, 等. 基于混合高斯模型的运动阴影抑制算法 [J]. *计算机应用*, 2006, 26(5): 1021–1023, 1026.)
- [6] BARNICH O, van DROOGENBROECK M. ViBe: a universal background subtraction algorithm for video sequences [J]. *IEEE Transactions on Image Processing*, 2011, 20(6): 1709–1724.
- [7] van DROOGENBROECK M, BARNICH O. ViBe: a disruptive method for background subtraction [M]// *Contribution to Collective Works: Background Modeling and Foreground Detection for Video Surveillance*. London: Chapman and Hall/CRC, 2014: 7.1–7.23.
- [8] HEIKKILA M, PIETIKAINEN M. A texture-based method for modeling the background and detecting moving objects [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006, 28(4): 657–662.
- [9] LIAO S, ZHAO G, KELLOKUMPU V, *et al.* Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes [C]// *CVPR 2010: Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington, DC: IEEE Computer Society, 2010: 1301–1306.
- [10] QIN R, LIAO S, LEI Z, *et al.* Moving cast shadow removal based on local descriptors [C]// *ICPR 2010: Proceedings of the 20th International Conference on Pattern Recognition*. Piscataway: IEEE, 2010: 1377–1380.
- [11] NVIDIA. CUDA C programming guide [EB/OL]. [2015-03-22]. http://docs.nvidia.com/cuda/pdf/CUDA_C_Programming_Guide.pdf
- [12] WANG L, SHI X. Parallel face contour extraction algorithm based on Chan-Vese model on multi-core CPU and GPU [J]. *Journal of Computer Applications*, 2014, 34(11): 3121–3125. (王丽娜, 史晓华. 基于 Chan-Vese 模型的面向多核 CPU 和 GPU 的人脸轮廓提取并行算法 [J]. *计算机应用*, 2014, 34(11): 3121–3125.)
- [13] TAN X, TRIGGS B. Enhanced local texture feature sets for face recognition under difficult lighting conditions [J]. *IEEE Transactions on Image Processing*, 2010, 19(6): 1635–1650.
- [14] LI L, HUANG W, GU I Y H, *et al.* Foreground object detection from videos containing complex background [C]// *MULTIMEDIA '03: Proceedings of the eleventh ACM International Conference on Multimedia*. New York: ACM, 2003: 2–10.
- [15] OJALA T, PIETIKAINEN M, MAENPAA T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2002, 24(7): 971–987.
- [16] YANG S, SHI M. A model for Web service discovery with QoS constraints [J]. *Chinese Journal of Computers*, 2005, 28(4): 589–594. (杨胜文, 史美林. 一种支持 QoS 约束的 Web 服务发现模型 [J]. *计算机学报*, 2005, 28(4): 589–594.)
- [17] HE Y, WU Z. Performance analysis of dynamic Web service composition key technologies [M]. Beijing: Tsinghua University Press, 2011: 48. (何炎祥, 吴钊. 动态 Web 服务组合关键技术与性能分析 [M]. 北京: 清华大学出版社, 2011: 48.)
- [18] LI J, XIA J, TANG W, *et al.* Survey on Web services selection algorithms based on QoS [J]. *Application Research of Computers*, 2010, 27(10): 3622–3627. (李金忠, 夏洁武, 唐卫东, 等. 基于 QoS 的 Web 服务选择算法综述 [J]. *计算机应用研究*, 2010, 27(10): 3622–3627.)
- [19] ZHENG Y, TIAN J, DOU W. A QoS routing algorithm based on Pareto optimal [J]. *Journal of Software*, 2005, 16(8): 1484–1489. (郑彦兴, 田菁, 窦文华. 基于 Pareto 最优的 QoS 路由算法 [J]. *软件学报*, 2005, 16(8): 1484–1489.)
- [20] COLONNI A, DORIGO M, MANIEZZO V. Distributed optimization by ant colonies [C]// *PECAL 1991: Proceedings of the 1st European Conference on Artificial Life*. New York: Elsevier Publishing, 1991: 134–142.
- [21] AL-MASRI E, MAHMOUD Q H. Investigating Web services on the world wide Web [C]// *WWW 2008: Proceedings of the 17th International Conference on World Wide Web*. New York: ACM, 2008: 795–84.
- [22] YANG Y. Parameter optimization and applications of ant colony algorithm [D]. Nanjing: Nanjing University of Technology, 2008: 12. (杨亚南. 蚁群算法参数优化及其应用 [D]. 南京: 南京理工大学, 2008: 12.)

(上接第 2243 页)