基于C-MMAS算法的组合服务动态选择研究

刘志中 王志坚 周晓峰 娄渊胜
(河海大学计算机与信息工程学院 南京 210098)

摘 要 将大规模的具有多种组合路径的QoS最优组合服务选择转换成带约束的最优路径选择问题,并提出了一种基于文化的最大-最小蚁群优化算法(C-MMAS)来完成最优路径选择。C-MMAS计算模型由基于MMAS的群体空间、基于优秀解的信仰空间及其之间的通信协议组成。群体空间在完成基于MMAS的演化后进行基于"变异"的进化操作,并将每次演化和进化后的优秀解作为知识贡献给信仰空间,信仰空间按照一定的优化规则更新空间里的知识,当信仰空间里的知识经过若干代的积累沉淀后再对群体的演化进行指导。此计算模型在知识和群体层面使用双重进化机制支持问题的求解和知识的提取,充分利用了种群的进化机制和知识的指导作用,在很大程度上提高了种群的多样性及收敛速度,达到了防止早熟、降低计算代价的目的。理论分析和实验结果说明了该算法的可行性和有效性。

关键词 组合服务选择,QoS约束,最大-最小蚁群算法,文化算法
中图法分类号 TP393 **文献标识码** A

Research on Composite Services Selection Based on C-MMAS Algorithm

LIU Zhi-zhong WANG Zhi-jian ZHOU Xiao-feng LOU Yuan-sheng
(College of Computer and Information Engineering, Hehai University, Nanjing 210098, China)

Abstract The problem of composite Web services selection with multiple composite paths was transformed into a constraint optimal path selection problem. A new optimization algorithm C-MMAS was proposed by integrating Max-Min Ant System into Culture algorithm framework, and was applied to solve the optimal path selection problem. This computing model consists of a MMAS-based population space, excellent-solution-based belief space and communication protocols between the two spaces. After completing MMAS-based evolution, population space carrys out variation-based evolution, and contributes excellent solutions as knowledge to belief space after evolutions. Belief space updates knowledge according to certain optimization principle. When the knowledge in belief has been accumulated and precipitated some generations, it is used to guide the MMAS-based evolution. Due to implementing two evolutionary mechanisms on population and knowledge, making the best use of population's evolutionary mechanism and guidance effect of knowledge, this computing model has improved population's diversity and convergence speed largely, realized the purpose of avoiding precocity and reducing computing expense. Theoretical analysis and experimental results indicate the feasibility and efficiency of this algorithm.

Keywords Composite services selection, QoS constraint, Max-min ant system, Culture algorithm

1 引言

在面向服务的体系结构(Service Oriented Architecture, SOA)中,Web服务组合是一种用来构建增值服务的方式。它将现有的Web服务按照一定的业务逻辑和组合模式进行集成,构建出满足用户需求的Web服务执行流程,已日益成为分布式计算的核心技术,得到了工业界和学术界的广泛关注,很多学者对此做了大量的研究并取得了丰富的成果。SOAP,WSDL,UDDI[1],BPEL[2]和ebXML[3]为Web服务组合提供了框架支持;组件服务通过标准协议(比如SOAP,WSDL)可以便捷地、与物理位置和平台无关地集成到一个组合服务中。组合服务包含多个子功能,本文称这些子功能为任务。在构建组合服务时会存在多种组合流程,这些组合流程包含的任务数量及任务组合顺序是不同的,本文将一个从起始任务到终止任务的组合流程称作一条组合路径。在Web环境中,每个任务会有多个Web服务与其对应,这些Web服务功能相同但QoS属性有很大差别,因此每条组合路径可以生成多个具有不同QoS属性的组合服务。如何从多种组合路径生成的众多组合服务中选出满足用户不同QoS需求的组合服务,是当前组合服务提供商所面临的难题。

到稿日期:2009-12-23 返修日期:2010-03-05 本文受国家自然科学基金项目(No. 60805022),国家高技术研究发展计划(863)(No. 2007AA01Z178)资助。

刘志中(1981-),男,博士生,CCF会员,主要研究方向为Web服务组合、进化算法研究,E-mail:lzzmff@126.com;王志坚(1958-),男,博士,教授,博士生导师,CCF会员,主要研究方向为软件自动化、软件构件复用;周晓峰(1965-),男,博士,教授,博士生导师,CCF会员,主要研究方向为软件构件复用、分布式计算;娄渊胜(1968-),男,博士,副教授,CCF会员,主要研究方向为软件测试、软件构件复用。

M. R. Garey 等人已经证明具有全局 QoS 约束的组合服务选择是一个 NP-Hard 问题[4]。一些学者针对这个问题提出了不同的解法。Zeng 等人[5]给出了一种基于整数规划的求解方法，D. Ardagna 等[6]考虑了局部 QoS 约束和全局 QoS 约束，提出了一种基于混合整数规划的解法，但这两种方法的时间复杂度比较高；S. L. Liu 等[7]将具有单一组合路径的组合服务选择问题转化成一个带 QoS 约束的多目标服务组合优化问题，并应用多目标遗传算法对其进行了求解；G. Canfora[8]给出了一种基于遗传算法的组合服务选择方法；YU 和 Lin[9]把组合服务选择问题转化成多选择的 0-1 背包问题(MCKP)，给出了穷尽搜索法、动态规划和 Pisinger 算法 3 种解法；PENG Xiao-ming[10]提出了一种基于蚁群算法的解法。上述求解方法一个共同的特点是只考虑了具有单一组合路径的组合服务选择，不能用来解决具有多种组合路径的组合服务选择问题。ZHANG Cheng-wen 等[11]给出了一种用于求解具有多种组合路径的组合服务选择的遗传算法，设计了一种基于关系矩阵的编码方式，克服了一维编码方式表示的局限性，但代表不同组合路径的染色体在交叉时会产生无效组合方案，每次交叉、变异后都需要进行正确性检查，增加了算法的复杂度。因此，设计一种能够高效地解决具有多种组合路径的组合服务选择方法仍然是组合服务应用领域所要解决的问题。

为了快速、灵活地解决具有多种组合路径 QoS 全局最优的组合服务选择问题，本文将大规模的具有多种组合路径的最优组合服务选择问题转换成带约束的最优路径选择问题，然后将改进的最大-最小蚁群算法(MMAS)[12]纳入文化算法(Culture Algorithm)[13-15]框架之内，构造了基于文化的最大-最小蚁群(C-MMAS)优化算法来求解最优路径选择问题。该计算模型由基于 MMAS 的群体空间、基于优秀解的信仰空间及其之间的通信协议组成。群体空间在完成基于 MMAS 的演化后进行基于“变异”的进化操作，并将每次演化和进化后的优秀解作为知识贡献给信仰空间，信仰空间按照一定的优化规则更新空间里的知识。当信仰空间里的知识经过若干代的积累沉淀后，再用其指导群体的演化。此计算模型在知识和群体层面使用双重进化机制支持问题的求解和知识的提取，充分利用了种群的进化机制和知识的指导作用，在很大程度上提高了种群的多样性及收敛速度，达到了防止早熟、降低计算代价的目的。

本文第 2 节描述了要解决的问题及其数学模型；第 3 节给出了基于 C-MMAS 算法的组合服务选择；第 4 节给出了本文的实验结果及比较分析；最后对本文的工作进行了总结。

2 问题描述及其数学模型

2.1 问题描述

通常情况下服务组合可以分为两个阶段，第一阶段产生抽象的组合服务，并将其描述成从虚拟起点通往虚拟终点的有向路径，路径上的每一个节点代表一个任务。由于任务分解粒度和分解方式的不同，针对同一个用户请求可以存在多种组合路径，每种组合路径由不同数目的任务节点组成。将不同的组合路径融合在一起就构成了一个有向图，如图 1 所示。图 1 由 4 条不同的组合路径构成，其中 V_s，V_d 为虚拟起点和虚拟终点，$T_i(1\leqslant i\leqslant 10)$为任务节点。

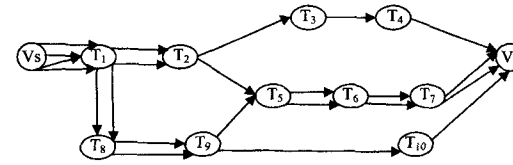


图 1 组合服务有向图

第二阶段实例化组合服务，从组合路径中每一个任务节点对应的候选服务类中选择一个服务实体，组成一个完整的组合方案，如图 2 所示。

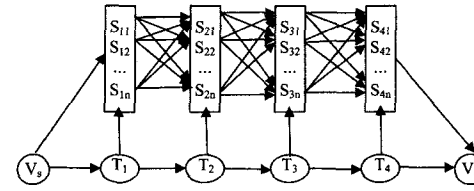


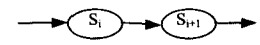
图 2 一条组合路径的实例化

图 2 表示一条组合路径上组合服务的实例化。每个任务节点对应的候选服务类的规模为 n，从这条组合路径上可以得到 n^4 个组合方案；当存在多种组合路径并且候选服务的规模比较大时，组合方案的数量是庞大的。本文要解决的问题就是从众多的组合方案中找出满足用户 QoS 约束具有最优 QoS 属性的组合服务。

2.2 服务组合模式及 QoS 聚合公式

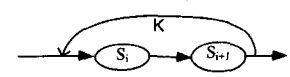
组合服务的 QoS 属性由组件服务的 QoS 属性及组件服务间的组合模式决定。根据组件服务的 QoS 属性及组件服务间的组合模式可以计算出整个组合服务的聚合 QoS 属性。目前关于组合服务 QoS 属性聚合方面的研究比较多，Cardoso[16]，M. C. Jaeger[17]以工作流模式为基础给出了 7 种基本的服务组合模式及部分 QoS 属性的聚合公式。本文给出了服务组合流程的 4 种基本组合模型，大部分聚合流程都可以由这 4 种基本模型组合而成，并给出了 4 种基本组合模式的 QoS 聚合公式。假设 Web 服务包括 4 种 QoS 参数，即执行时间 T(time)、执行费用 C(cost)、信誉等级 Rep(reputation)和可靠性 R(reliability)。设 cs 为多个服务形成的聚合服务，s_i 为组成组合服务的单个服务，s_i 和 cs 的 QoS 属性模型分别为 $Qs_i=(C_i,T_i,Rep_i,R_i)$，$Qcs_i=(C_{cs},T_{cs},R_{cs},Rep_{cs})$。服务组合基本模式的 QoS 参数计算方法如下。

(1) Sequential



$C_{cs}=C(s_i)+C(s_{i+1})$，$T_{cs}=T(s_i)+T(s_{i+1})$，$Rep=\min\{Rep(s_i),Rep(s_{i+1})\}$，$Rel_{cs}=Rel(s_i)\cdot Rel(s_{i+1})$

(2) Loop

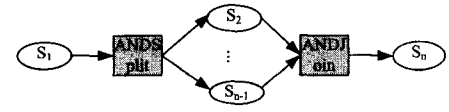


$C_{cs}=k(C(s_i)+C(s_{i+1}))$，$T_{cs}=K(T(s_i)+T(s_{i+1}))$

$Rep_{cs}=\min\{Rep(s_i),Rep(s_{i+1})\}$，$Rel_{cs}=(Rel(s_i)\cdot Rel(s_{i+1}))^k$

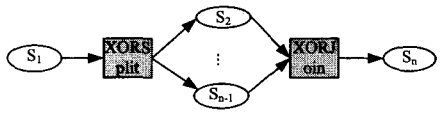
其中，k 为循环次数。

(3) AND Split AND Join



$C_{cs}=C(s_i)+\cdots+C(s_{i+1})$，$T_{cs}=T(s_1)+\max\{T(s_2),\cdots,T(s_{n-1})\}+T(s_n)$，$Rep_{cs}=\min\{Rep(s_1),\cdots,Rep(s_n)\}$，$Rel_{cs}=Rel(s_1)\cdot\cdots\cdot Rel(s_n)$

(4) XOR Split XOR Join



$$C_{cs}=\begin{cases}C(s_1)+C(s_2)+C(s_n)\\ \cdots \quad \text{或}\\ C(s_1)+C(s_{n-1})+C(s_n)\end{cases}$$

$$T_{cs}=\begin{cases}T(s_1)+T(s_2)+T(s_n)\\ \cdots \quad \text{或}\\ T(s_1)+T(s_{n-1})+T(s_n)\end{cases}$$

$$Rep_{cs}=\begin{cases}\min\{Rep(s_1),Rep(s_2),Rep(s_n)\}\\ \cdots \quad \text{或}\\ \min\{Rep(s_1),Rep(s_{n-1}),Rep(s_n)\}\end{cases}$$

$$Rel_{cs}=\begin{cases}Rel(s_1)*Rel(s_2)*Rel(s_n)\\ \cdots \quad \text{或}\\ Rel(s_1)*Rel(s_{n-1})*Rel(s_n)\end{cases}$$

2.3 数学模型

设用户提出全局 QoS 约束和偏好为$\{\{C_{cs}\leqslant c|w_1\};\{T_{cs}\leqslant t|w_2\};\{R_{cs}\geqslant r|w_3\};\{Rep_{cs}\geqslant rep|w_4\}\}$，其中 $w_i(1\leqslant i\leqslant 4)$表示用户对每个属性的偏好，并且 $w_1+\cdots+w_4=1$。由于各个 QoS 属性的度量单位及取值范围不相同，首先要对 QoS 属性值按式(1)、式(2)进行标准化。其中 $q_{i_{com}}$ 为组合服务的聚合 QoS 属性，$q_{i\max}$，$q_{i\min}$ 为所有组合服务聚合 QoS 属性值的最大、最小值。

$$q'_{cs}=\begin{cases}\dfrac{q_{com}-q_{\min}}{q_{\max}-q_{\min}}, & \text{if } q_{\max}-q_{\min}\neq 0\\ 1, & \text{if } q_{\max}-q_{\min}=0\end{cases} \quad \text{positive attribute} \tag{1}$$

$$q'_{cs}=\begin{cases}\dfrac{q_{\max}-q_{com}}{q_{\max}-q_{\min}}, & \text{if } q_{\max}-q_{\min}\neq 0\\ 1, & \text{if } q_{\max}-q_{\min}=0\end{cases} \quad \text{negative attribute} \tag{2}$$

在本文中，C_{cs}，T_{cs} 取值方向为负，R_{cs}，Rep_{cs} 取值方向为正。具有多种组合路径且支持用户 QoS 偏好的组合服务选择问题的数学模型为：

ObjectiveFunction

$$\text{Max } f(CS_i),f(CS_i)=w_1*C'_{cs_i}+w_2*T'_{cs_i}+w_3*R'_{cs_i}+w_4*Rep'_{cs_i}$$

$$\text{Constraints:}\begin{cases}C'_{cs_i}\leqslant c\\ T'_{cs_i}\leqslant t\\ R'_{cs_i}\geqslant r\\ Rep'_{cs_i}\geqslant rep\end{cases} \quad cs_i \text{ 为第 } i \text{ 个组合服务，}$$

$i=1,2,\cdots$

3 基于 C-MMAS 的组合服务选择算法

3.1 改进的最大-最小蚁群算法

在 MMAS[14]算法中，蚂蚁每一步都沿着选择概率最大的方向转移到下一个节点，直到找到终点为止。如果始终以最大概率作为转移条件，将会造成多数蚂蚁聚集到相同的路径上，降低了算法的随机搜索能力。为了增加算法搜索的随机性，本文采用轮盘赌方式选择候选服务。

$$P_{ij}=\begin{cases}\dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{s\in allowed}[\tau_{is}(t)]^{\alpha}[\eta_{is}(t)]^{\beta}}, & j\in \text{allowed}\\ 0, & \text{其他}\end{cases} \tag{4}$$

式(4)为每个候选服务选择概率计算公式。$\tau_{ij}(t)$表示 t 时刻候选服务 s_{ik} 与 s_{jl} 之间的信息素强度；$\eta_{ij}(t)$表示 t 时刻从候选服务 s_{ik} 转移到 s_{jl} 的启发信息，本文定义 $\eta_{ij}(t)=f(CS_i)$，$f(CS_i)$为蚂蚁访问过的服务组成的子组合服务的聚合 QoS 属性关于用户 QoS 偏好的综合评价值；α 为信息素的相对重要程度，β 为启发信息的相对重要程度。

MMAS 算法只增加每次迭代后当前最优解路径上信息素，挥发衰减其他解路径上的信息素，这样可能会造成不是全局最优解的路径上信息素浓度不断增加，从而影响算法的收敛速度。为了加快算法的搜索速度，提高算法的搜索性能，并且体现解的优秀程度对信息素增量的不同影响，本文设计了一种全局信息素更新规则：在每次循环结束后，对适应度值排在前 λ 位的解路径以式(5)进行信息素增加；其余解路径上的信息素以式(7)进行衰减。

$$\tau_{ij}(t+1)=\rho\tau_{ij}(t)+\Delta\tau^{k}_{ij}(t) \tag{5}$$

$$\Delta\tau^{k}_{ij}(t)=\begin{cases}\dfrac{1}{f(CS_k)}, & \text{if } e_j\in L_k\\ 0, & \text{otherwise}\end{cases} \tag{6}$$

$$\tau_{ij}(t+1)=\rho_2\tau_{ij}(t) \tag{7}$$

式中，$\rho(0<\rho<1)$为全局信息素保留因子，$\Delta\tau^{k}_{ij}(t)$表示适应度值排在第 k 位的解路径信息素增量，$f(CS_k)$表示排在第 k 位的解的适应度值。L_k 表示其路径，e_j 为 L_k 的边，δ 为需要通过实验来确定的参数。

本文采用文献[14]中的最大息素计算公式：

$$\tau_{\max}=\begin{cases}Q, & \text{at beginning}\\ \dfrac{1}{\rho}\dfrac{1}{f(S_{best\text{-}so\text{-}far})}, & \text{otherwise}\end{cases} \tag{8}$$

并且令

$$\tau_{\min}=\frac{\tau_{\max}}{20} \tag{9}$$

式中，Q 为一个常量，$S_{best\text{-}so\text{-}far}$ 是当前所找到的最优解。

3.2 基于文化的最大-最小蚁群算法(C-MMAS)

C-MMAS 算法由基于 MMAS 的群体空间、信仰空间及两者之间的通信协议组成。群体空间包含一系列所求问题的解，信仰空间用于保存种群空间在进化过程中获取的优秀解，作为经验和知识用来指导群体空间种群的演化。其计算框架如图 3 所示。

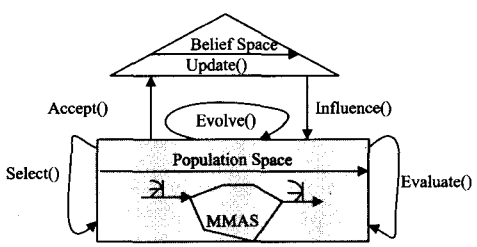


图 3 C-MMAS 算法计算框架

C-MMAS 算法的计算过程是：首先由 MMAS 算法生成种群空间，并对种群个体进行评价(Evaluate())；通过接受函数(Accept())将优秀的个体知识提取到信仰空间，并根据更

新函数(Update())按一定的优化规则更新信仰空间的知识；然后对群体空间的个体进行进化操作(Evolve())，评价通过进化得到的新解，并再次将优秀的个体提取到信仰空间，更新信仰空间的知识；在信仰空间的知识经过 η 代更新、积累、沉淀后通过影响函数(Influence())来指导种群演化。下面给出各种函数的定义。(1)Evaluate()：计算每个解的适应度值。本文中解的适应度值为各组合方案聚合 QoS 属性关于用户 QoS 偏好的综合评价值 $f(\cdot)$。(2) Accept()：提取适应度值排在前若干位的解到信仰空间。Accept()$=m\cdot\chi$，m 为种群的数量，χ 为提取系数，本文设 $\chi=10\%$。(3)Update()：在信仰空间用适应度值较好的新知识替代适应度值较差的旧知识。(4) Evolve()：对群体空间的个体实施变异操作。具体操作是随机选择由解构成的组合服务中的一个组件服务，再从被选择的组件服务所属的候选服务类中随机选择一个服务进行替换，本文采用两点式变异。(5) Influence()：用信仰空间的知识替换种群中较差的解，然后依据全信息素更新规则更新全局信息素。

在进行组合服务选择之前，根据文献[17]中的方法将 Loop，AND Split AND Join，XOR Split XOR Join 等组合模式转换成 Sequential 组合模式，并用有向图的邻接矩阵来描述服务组合中各任务之间的组合顺序，根据任务节点间的组合顺序确定任务节点对应的候选服务类中的候选服务之间的组合顺序，这就大幅度地简化了问题的描述，提高了问题的求解效率。蚂蚁在行进的过程中首先确定当前所在服务 s_{ik} 所对应的任务节点 T_i，然后找出 T_i 的后续任务节点 T_j，进而找出 T_j 对应的候选服务类 S_j，即蚂蚁下一步要选择的候选服务集。对于所有的候选服务，根据相应的 QoS 聚合公式计算将每一个候选服务添加到蚂蚁当前记录表后得到的子组合方案的聚合 QoS 属性，并判断当前子组合方案的聚合 QoS 属性是否满足 QoS 约束。如果不满足，则舍弃这个候选服务；如果满足，则计算该子组合方案的聚合 QoS 属性关于用户 QoS 偏好的综合评价值，并依据式(7)计算这个候选服务的选择概率，最后依据转移规则选择一个候选服务并更新路径记录表；如果所有的候选服务都不满足被选择的条件，则这只蚂蚁停止搜索；到达终点后，蚂蚁的路径记录表即为问题的一个解。下面给出 C-MMAS 算法的描述。

3.3 C-MMAS 算法描述

Step1　初始化阶段

输入：服务组合有向图的邻接矩阵、候选服务及其 QoS 属性值、用户 QoS 约束及偏好；初始化参数 $\alpha,\beta,\rho,Q,\lambda,\eta,p_{best},\tau_{max},\tau_{min}$；初始化候选服务之间的信息素矩阵 A、每只蚂蚁的路径记录表 S；设置蚂蚁的个数为 m，并将其放在路径的初始节点 V_s 上，设置最大迭代次数 N_{max}；

Step2　蚁群的一次完整搜索

```
For k=1 to P /*P为有向图中最长路径上的任务个数*/
 {
  For k=1 to m /*m只蚂蚁的一步转移*/
   { If (当前节点不是终点)
     {For(对于所有的候选服务)
       {计算添加一个候选服务得到的子组合服务的聚合 QoS
        属性;}
        If(聚合 QoS 属性满足用户 QoS 约束)
          {计算聚合 QoS 属性关于用户 QoS 偏好的综合评价
           值并计算候选服务的选择概率;}
        Else
          {过滤掉当前的候选服务;}
       }
     If(存在满足选择条件的候选服)
       {采用轮盘赌方式选择一个服务;}
     Else
       {该蚂蚁停止搜索;}
   If(到达终点)
     {存储路径表并停止搜索}
   }
}
```

Step3　基于文化的演化过程

```
For k=1 to q
  {Evaluate() and Sort();}/*评价解并排序*/
Then
{Accept() and Update();}/*提取知识,更新知识*/
Then
Evolve();/*对群体进行进化操作*/
For i=1 to n /*对进化得到的新解进行评价、排序/*
{Evaluate() and Sort();}
Then
{Accept() and Update();}
If(积累代数==η)
{Influence()andUpdatePheromone();}
  /*信仰空间的知识对群体的演化进行一次指导*/
Else
{UpdatePheromone();} /*全局信息素更新*/
```

Step4　判断算法是否结束

```
If(达到结束条件)
      {输出最优解 并 停止计算;}
Else
  {迭代次数增加一次并返回 Step2}
```

4　实验结果及比较分析

4.1　实验设计

本文采用图 1 所示的服务组合有向图作为实验对象。C-MMAS 算法用 C++实现，算法运行的微机配置为 Pentium (R)4 2.66GHz 处理器，512M 内存，操作系统为 Windows-XP2002。每个候选服务类的规模为 30，候选服务的 QoS 属性值采用随机方法在一定范围内生成。本文共考虑了 4 种 QoS 属性，它们的取值范围分别设定为 $0\leqslant C\leqslant 100$ 元，$0<T\leqslant 10$s，$0.75<Rel\leqslant 1$，$Rep\in\{1,2,3,4,5\}$。用户的 QoS 约束及其偏好为 C:{$C\leqslant 200|0.4$}，T:{$T\leqslant 25|0.25$}，R:{$R\geqslant 0.65|0.2$}，Rep:{$Rep\geqslant 3|0.15$}。

4.2　算法参数确定

C-MMAS 算法性能的优劣取决于算法中参数的取值。在应用 C-MMAS 算法求解本文中的问题时，需要为算法中的参数设定合适的取值。算法的系统参数包括信息素影响因子 α、启发式信息影响因子 β、信息素全局挥发因子 ρ、信息素更新规则中优秀解个数 λ、信仰空间中知识积累和沉淀的代数 η。本文采用实验的方法测试算法中各关键参数的设置对算法搜索性能的影响，并为各个参数选取最为合适的值。通过

实验发现，为上述参数设置不同的值时，只会影响算法搜索到的解的优劣，而对算法的运行时间基本没有影响。所以在下面的实验中着重考察了在参数取不同的值时算法搜索到的解的适应度值，不再考虑算法的运行时间。

首先测试参数 α 与参数 β 的最佳比值来确定其相对重要程度。令 $\alpha=1.0$，$\beta=b\alpha$，测试的目的是选取合理的比例系数 μ。初始时设置蚂蚁的数量为 $m=50$，挥发因子 $\rho=0.5$，初始最大信息素 $Q=1$，最大循环次数 $N_{max}=100$，通过设置不同的 b 值，观察算法搜索到的解优劣，如图 3 所示。从图 3 可以看出，当 $b=4$ 时算法搜索到的解最优，因此确定 b 的取值为 4。

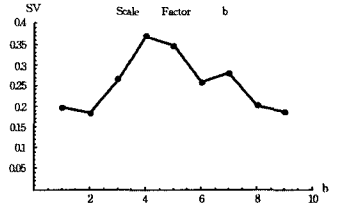


图 3　b 对算法搜索能力的影响图

基于上面的测试结果，令 $b=4$，然后测试信息素挥发因子 ρ 对算法搜索能力的影响并确定其合适的取值。测试方法与上面相同。测试结果如图 4 所示。从图 4 可以看出 ρ 的取值对算法的性能有较大的影响，ρ 偏小时，蚂蚁在行进中难以根据信息素值的大小判断路径的优劣；而 ρ 偏大时，则容易陷于局部最优。根据测试结果，对于 C-MMAS 算法，当 $\rho=0.6$ 时，算法搜索到的解最优，因此确定 ρ 的取值为 0.6。

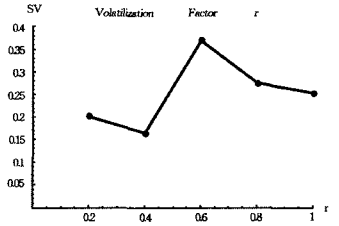


图 4　ρ 对算法搜索能力的影响

令 $b=4$，$\rho=0.6$，测试信息素更新规则中优秀解的个数 λ 对算法性能的影响。如图 5 所示，通过实验发现，在本文中取 $\lambda=5$ 比较合适，即在进行全局信息素更新时按式(5)对适应度值排在前 5 位的解路径上的信息素进行更新。

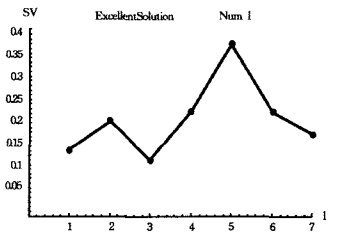


图 5　λ 对算法搜索能力的影响

令 $b=4$，$\rho=0.6$，$\lambda=5$，测试信仰空间的知识积累、沉淀的代数 η 对算法搜索能力的影响，也即是确定每隔多少代用信仰空间的知识来指导群体的演化会使得算法达到更好的搜索效果。其实验结果如图 6 所示。从图 6 可以看出，当 $\eta=5$ 时，算法搜索到的解最优；而当 $\eta\geqslant10$ 时，η 不再影响算法的性能。因此确定，在本文的 C-MMAS 算法中蚁群每迭代 5 次后用信仰空间的知识对种群的演化进行一次指导。

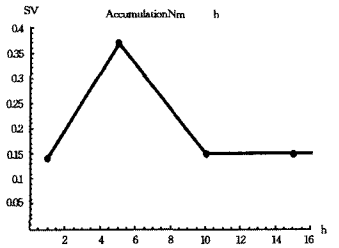


图 6　η 对算法搜索能力的影响

4.3　与 CAS、MMAS 算法的比较分析

鉴于此问题与 TSP 问题、路径寻优问题、任务调度问题具有相似性，为了验证 C-MMAS 算法的求解效果，我们分别用在求解 TSP、二次规划、任务调度等问题时表现良好的 ACS(ant colony system)[18] 算法、MMAS 算法及本文的 C-MMAS 算法对 4.1 节中设计的实验对象进行了求解，分别测试了每种算法在迭代 20，40，60，80，100，120 次时搜索到的解的适应度值及相应的运行时间，如图 7 所示。

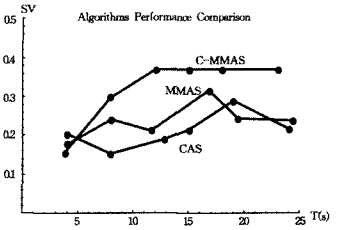


图 7　不同算法性能比较

从图 7 中可以看出 3 种算法在迭代相同次数运行的时间基本相同，但所搜索到的解的优劣有很大差别：C-MMAS 算法在迭代 60 次、运行时间为 12s 时搜索的解远远优于 CAS、MMAS 算法迭代 60 次时搜索到的解，并且趋于收敛状态。而 CAS 算法及 MMAS 算法在整个测试过程中都没有呈现出收敛趋势，并且在相同迭代次数的情况下所搜索到的解远远劣于 C-MMAS 搜索到的解。另外，我们还分别将 3 种算法分别迭代了 200 次，测试结果是：C-MMAS 算法搜索到的解与迭代 60 次搜索到的解相同，适应度值为 0.3687，所用时间为 39s，而 CAS 算法与 MMAS 算法所搜索到的解的适应度值及运行时间分别为 0.233/40s，0.207/39s。通过对比可以得出，C-MMAS 算法的收敛速度及寻优性能远远优于 ACS 算法及 MMAS 算法，并且其求解效果在运行时间及解的优劣上都能够满足实际应用的需要。由此可以得出 C-MMAS 算法在求解大规模的具有多种组合路径的组合服务选择问题上是非常有效的。

C-MMAS 优化算法在 MMAS 算法的基础上增加了并行演化的信仰空间，并且在种群空间中对群体进行进化操作，进化过程涉及到的运算比 MMAS 算法中的循环过程要简单得多，因此只需较短的时间便可完成相同次数的运算。并且在信仰空间知识的指导下 MMAS 进行全局信息素更新，使算法跳出局部最优解的能力得到明显提高，从而大大改善了整个群体的寻优性能。信仰空间的存在增强了蚁群的记忆功能，提高了算法的寻优效率及收敛速度，减少了计算时间。

结束语　本文研究目前基于全局 QoS 约束的组合服务选择中存在的问题并给出了有效的解决方法。将具有多种组合路径的 QoS 最优的组合服务选择问题转换成带约束的最优路径选择问题，将改进的最大-最小蚁群算法纳入文化算法

框架之内，构造了基于文化的最大-最小蚁群算法，并用来解决最优路径选择问题。此计算模型在知识和群体层面使用双重进化机制支持问题的求解和知识的提取，充分利用了种群的进化机制和知识的指导作用，在很大程度上提高了种群的多样性及收敛速度，达到了防止早熟、降低计算代价的目的。通过大量的实验验证及与其他算法的比较得出，C-MMAS算法求解速度快、寻优成功率高，是一种求解复杂组合服务选择问题的有效方法。

参 考 文 献

[1] Curbera F, et al. Unraveling the web services: An introduction to SOAP, WSDL, and UDDI[J]. IEEE Internet Computing, 2002, 6(2): 86-93

[2] Web Service — Business Process Execution Language (WS BPEL), Version 2.0[S]. OASIS Committee Draft, 17th May, 2006

[3] Patil S, Newcomer E. ebXML and Web services[J]. IEEE Internet Computing, 2003, 7(3): 74-82

[4] Garey M R, Johnson D S. Computers and Intractability: A Guide to the Theory of NP-Completeness[M]. New York: Freeman Press, 1979

[5] Zeng L, Benatallah B, Ngu A H H, et al. QoS-aware Middleware for web services composition[J]. IEEE Transaction on Software Engineering, 2004, 30(5): 311-327

[6] Ardagna D, Pernici B. Global and local QoS guarantee in Web service selection[C]// Proc. of Business Process Management Workshops. 2005: 32-46

[7] 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，2007，18：646-656

[8] Canfora G, Penta M D, Esposito R, et al. An approach for QoS-aware service composition based on genetic algorithms[C]// Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2005). 2005: 1069-1075

[9] Yu T, Lin K J. Service selection algorithms for Web services with end-to-end QoS constraints[C]// Proceedings of IEEE International Conference on E-Commerce Technology (CEC). 2004: 129-136

[10] 彭晓明，何炎祥，朱兵舰. 蚁群算法在Web服务组合中的应用[J]. 计算机工程，2009，35：182-187

[11] 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，2006，29：1029-1037

[12] Thomas S, Holger H, et al. Max-Min ant system [J]. Future Generation Computers System, 2000, 16(8): 889-914

[13] Reynolds R G. An Introduction to Cultural Algorithms [C] // Proceedings of the Third Annual Conference on Evolutionary Programming. River Edge, New Jersey: World Scientific, 1994: 131-39

[14] 刘升，王行愚，游晓明. 求解TSP问题的文化蚁群优化算法[J]. 华东理工大学学报，2009，35(2)：288-292

[15] 郭一楠，王辉. 文化算法研究综述[J]. 计算机工程与应用，2009，45(9)：41-46

[16] Cardoso J, Sheth A P, Miller J A, et al. Modeling quality of service for workflows and Web service processes[J]. Web Semantics Journal: Science, Services and Agents on the World Wide Web Journal, 2004, 1(3): 281-308

[17] Jaeger M C, Rojec-Goldmann G, Muhl G. QoS aggregation for Web service composition using workflow patterns[C]// Proceedings of IEEE International Conference on Enterprise Distributed Object Computing (EDOC). 2004: 149-159

[18] Dorigo M. Ant colony system: a cooperative learning approach to the traveling salesman problem [J]. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 53-56

(上接第134页)

面临的难题。以往常用的基于变更覆盖的选择方法，由于仅考虑了测试用例覆盖变更的多少，没有考虑测试用例对变更测试的典型性，使得最终选择出的回归测试子集难以具有原测试集等效的测试效果。利用多维标度法获得的回归测试集合则解决了这一问题，使得最终获得的回归测试子集不仅检验了软件变更，而且代表了原测试集不同的测试目的，从而使测试结果能够代表原测试集的检验效果，能够说明更改后软件的质量。

参 考 文 献

[1] Kim Jung-Min, Adam P, Gregg R. An empirical study of regression test application frequency[J]. Software Testing, Verification & Reliability, 2005, 15(4): 257-279

[2] Leon D, Podgurski A. A Comparison of Coverage-based and Distribution-based Techniques for Filtering and Prioritizing Test Cases[C]// Proceedings of the 14th International Symposium on Software Reliability Engineering. 2003: 442-454

[3] Leon D, Masri W, Podgurski A. An Empirical Evaluation of Test Case Filtering Techniques Based on Exercising Complex Information Flows[C]// Proceedings of the 27th International Conference on Software Engineering. 2005: 412-421

[4] Dennis J, Neelam G. Experiments with Test Case Prioritization Using Relevant Slices[J]. The Journal of Systems and Software, 2008, 81(2): 196-221

[5] Dennis J, Neelam G. Test Case Prioritization Using Relevant Slices[C]// Proceedings of the 30th Annual International Computer Software and Applications Conference. 2006: 411-420

[6] Dickinson W, Leon D, Podgurski A. Finding Failures by Cluster Analysis of Execution Profiles[C]// Proceedings of the 23rd International Conference on Software Engineering. 2001: 339-348

[7] Cesin L, Zaen D. Profile Analysis Techniques for Observation-based Software Testing. 2005: 132

[8] Amitabh S, Jay T. Effectively prioritizing tests in development environment[J]. ACM SIGSOFT Software Engineering, 2002, 27(4): 97-106

[9] 林亚平，胡玉鹏，陈治平. 基于执行剖面过滤的分割测试[J]. 湖南大学学报，2006，33(1)：110-113

[10] 赵亮，王建民，孙家广. 软件测试准则的有效性度量研究[J]. 计算机研究与发展，2006，43(8)：1457-1463

[11] 朱建平. 应用多元统计分析(第一版)[M]. 北京：科学出版社，2006：213