# 一种基于 QoS 全局最优的服务选择算法 *

王阳阳[1,2], 李 俊[1], 陈志国[1], 孟 芸[1], 丁海燕[1]
(1. 河南大学 计算机与信息工程学院, 河南 开封 475001; 2. 中国安全生产科学研究院, 北京 100029)

**摘 要**: 现有的服务选择算法存在低效、非全局最优等缺点。针对现有算法的不足,提出了一种基于 QoS 的高效服务选择算法。首先建立服务选择问题的多目标优化模型,之后用改进的多目标粒子群算法(IDMPSO)求解该模型,从而获得一组高质量最优解。在 IDMPSO 中,通过计算粒子的密集距离来进行 Pareto 最优解的保留,并把密集距离与欧几里德距离结合起来提出一种全局最优粒子选取的方法。实验结果表明,IDMPSO 得到的解相对较优,且分布均匀,并且随着问题规模的增加,运行时间呈线性增长。

**关键词**: Web 服务组合; 服务质量; 多目标粒子群; 密集距离; 服务选择
**中图分类号**: TP393 **文献标志码**: A **文章编号**: 1001-3695(2010)05-1659-03
doi:10.3969/j.issn.1001-3695.2010.05.013

# Web services selection algorithm with QoS global optimal

WANG Yang-yang[1,2], LI Jun[1], CHEN Zhi-guo[1], MENG Yun[1], DING Hai-yan[1]
(1. *School of Computer & Information Engineering, Henan University, Kaifeng Henan* 475001, *China*; 2. *China Academy of Safety & Technology, Beijing* 100029, *China*)

**Abstract**: The existing service selection algorithms have defects such as inefficient, non-global optimization. For overcoming the defects, this paper proposed an efficient service selction algorithm, IDMPSO based on services' QoS. Firstly, modeled the service selcetion as multiple objective optimization problem. Then, designed an improved multiple objective particle swarm optimization algorithm to solve the above problem, and obtained a group of high-quality solutions at last. In IDMPSO, employed crowding-measure to maintain Pareto optimal solutions and adopted a new meathod to find global optimal particles. The experiment results show that IDMPSO can obtain enough, good distribution solutions and its running time increases linearly with the problem size.

**Key words**: Web services composition; QoS; multi-objective particle swarm; intensive distance; service selection

Web 服务作为一种新型的分布式计算模型,近年来已经在电子商务、企业应用集成等领域扮演着越来越重要的角色。但是在实际应用中,单个的 Web 服务提供的功能比较单一,通常无法满足复杂应用的需求。因此,如何有效地组合分布于 Internet 中的各类服务,实现服务之间的无缝集成,已经成为 Web 服务发展过程中的一个重要步骤。随着 Web 服务数量的增多,出现许多服务提供者提供的服务具有相同的功能但具有不同的 QoS,所以在服务组合过程中,需要根据用户的 QoS 要求对 Web 服务进行选择。因此,基于 QoS 的服务选择问题已经成为服务组合中的关键问题。目前服务选择的算法主要有穷尽计算法、线性规划法、遗传算法、粒子群算法等。文献[1]提出了使用粒子群算法进行 Web 服务选择。但文献[1]存在如下问题: a)算法所产生的优化结果为单个解,用户没有选择的余地,通常情况下,用户更关心是否能获取一组可接受的最优非劣解,根据实际需要选择最满意的服务组合流程;b)最优解的质量对权重向量的选择非常敏感,而且需要用户对问题有一定的认识。为此,本文提出了在 Web 服务组合中基于 QoS 的改进型粒子群算法。

## 1 基于服务组合的多目标优化模型

### 1.1 问题描述

Web 服务组合有四种基本结构,即串联模型、并联模型、选择模型和循环模型[2]。其他服务组合流程均可以用这四种基本模型复合而成。在 Web 服务组合中,一个完整的 Web 组合服务是由多个 Web 子服务组成,每个 Web 子服务又对应多个候选服务。这些候选服务由不同提供者提供,具有相同调用接口、相同功能和不同 QoS 属性。因此每条组合路径又包含多个执行方案(从组合路径中的每个 Web 子服务的候选服务中选取一个服务来完成组合路径功能)。而每个执行方案的 QoS 属性可以通过上面介绍的四种基本模型的 QoS 计算公式[2]来获取。

假设 Web 服务是顺序模型的结构,并包括四种 QoS 参数:执行时间(time)、执行费用(cost)、可靠性(reliability)、信誉等级(reputation)。令 WS 代表某个 Web 组合服务,该 Web 组合服务包含的 Web 子服务个数为 $n$,$S_i(1\leqslant i\leqslant n)$表示第 $i$ 个 Web 子服务所对应的候选服务集,$|S_i|$表示第 $i$ 个 Web 子服务所对应的候选服务集的个数,$s_{ij}(1\leqslant i\leqslant n, 1\leqslant j\leqslant |S_i|)$表示第 $i$ 个

**收稿日期**: 2009-09-08; **修回日期**: 2009-10-21 **基金项目**: 国家"十一五"科技支撑计划重大资助项目(2007BAK23B01);2006 年度安科基金资助项目(AK2007-05)

**作者简介**:王阳阳(1983-),男,河南新乡人,硕士研究生,CCF 会员,主要研究方向为软件工程、语义网(wangyang083@vip.qq.com);李俊(1981-),女,硕士研究生,主要研究方向为计算机网络;陈志国(1955-),男(回族),河南开封人,教授,主要研究方向为软件工程、人工智能;孟芸(1979-),女,硕士研究生,主要研究方向为软件工程、语义网;丁海燕(1984-),女,硕士研究生,主要研究方向为空间数据.

Web 子服务中的第 $j$ 个服务实例，$q_{ij}$ 表示第 $i$ 个 Web 子服务中的第 $j$ 个服务实例的 QoS 属性，$\mathrm{Rep}_0$ 表示该 Web 组合服务的最小信誉等级，$R_0$ 表示该 Web 组合服务的最小可靠性。

### 1.2 优化目标和约束函数的确定

基于 QoS 的 Web 服务组合就是要从某个 Web 组合服务(WS)中的每个 Web 子服务所对应的候选服务集中选取一个合适的服务实例 $s_{ij}(1\leqslant i\leqslant n,1\leqslant j\leqslant |S_i|)$，使得组合服务(WS)的属性可靠性、信誉等级在满足给定约束的条件下，属性执行费用、执行时间达到最优。由此可确定问题的优化目标和约束函数为

$$\text{优化目标：}\min F(\mathrm{WS})=(T(\mathrm{WS}),C(\mathrm{WS})) \tag{1}$$

$$\text{约束函数：}R(\mathrm{WS})\geqslant R_0\ \mathrm{Rep}(\mathrm{WS})\geqslant \mathrm{Rep}_0 \tag{2}$$

其中：$T(\mathrm{WS})$ 对应服务组合流程中 QoS 参数中的执行时间的计算式；$C(\mathrm{WS})$ 对应服务组合流程中 QoS 参数中的执行费用的计算式；$\mathrm{Rep}(\mathrm{WS})$ 对应服务组合流程中 QoS 参数中的信誉等级的计算式；$R(\mathrm{WS})$ 对应服务组合流程中 QoS 参数中的可靠性的计算式；$\mathrm{Rep}_0$ 表示该 Web 组合服务的最小信誉等级；$R_0$ 表示该 Web 组合服务的最小可靠性。式(1)表示要取向量的极小化，使得优化目标 $F(\mathrm{WS})$ 中的执行时间 $T(\mathrm{WS})$ 和执行费用 $C(\mathrm{WS})$ 同时极小化。式(2)表示可靠性 $R(\mathrm{WS})$ 和 $\mathrm{Rep}(\mathrm{WS})$ 要满足约束条件。

## 2 算法描述

在对服务选择问题建立相应的优化模型后，必须设计相应的算法来求解。服务组合问题属于 NP-Complete 问题[3]，因此不能应用传统的优化求解技术求解。本文基于群体智能算法来设计服务选择算法。粒子群优化算法是一种新兴的群体智能优化技术，是由 Eberhart 等人于 1995 年共同提出的一种模仿鸟类群体行为的智能优化方法。由于算法思想直观、容易实现、具有较高的执行效率，且只有少数参数需要调整，得到了广泛的应用。本文根据 Web 服务组合的特点，对传统的粒子群算法进行改进，通过同时优化一个 Web 组合服务流程的多个 QoS 参数，最终产生一组满足约束条件的 Pareto 最优解[2]，用户可以根据自己的需要从中选择最满意的解。

算法从 QoS 全局最优的角度出发，搜索 Web 服务组合流程中满足所有 QoS 约束条件的一组非劣解[2]。基本思想是：将一个 Web 服务组合流程编码为一个粒子，粒子的维数代表组成该条路径的 Web 子服务的个数，而每一维的范围代表该维所对应的 Web 子服务所对应候选服务的个数。在每次迭代中，粒子通过跟踪个体极值 $p$best 与全局极值 $g$best 来更新自己的速度和位置，产生新一代的粒子，实现了在解空间的并行搜索。算法停止时，得到了一个粒子集合，集合中粒子就是满足约束条件的服务组合流程集。具体流程见算法 1。

**算法 1** IDMPSO

输入：种群 $P_1$ 及辅助种群 $P_2$ 的规模，进化代数 $T$。
输出：Pareto 最优解集 $P^*$。
begin
a) t = 0
b) $P_1(t)$, $P_2(t)=\varnothing$
c) Initialize($P_1(t)$)
d) OptimalSelect($P_1(t)$, $P_2(t)$)
e) while(t≤T)
f) SelectBestPosition($P_1(t)$, $P_2(t)$)
g) $P_1(t)$ = Update($P_1(t)$)
h) SelPbest($P_1(t)$)
i) $P_2(t+1)$ = OptimalSelect($P_1(t)$, $P_2(t)$)
j) t = t + 1
k) goto e)
l) $P^*=P_2(t)$
m) output($P^*$)
end

其中：步骤 c)用来初始化粒子群；步骤 d)是利用种群 $P_1$ 及辅助种群 $P_2$ 将初始化的粒子群中的 Pareto 最优解放入到辅助种群 $P_2(t)$ 中；步骤 f)为种群 $P_1(t)$ 中的每个粒子选择全局最优位置，具体见 2.1 节；步骤 g)根据速度与位置的更新公式对种群 $P_1$ 中的粒子进行更新，检查更新后粒子的速度与位置是否在限定的范围内，同时更新粒子的个体极值；步骤 h)是确定每个粒子自身的最优位置；步骤 i)利用种群 $P_1$ 及辅助种群 $P_2$ 进行优良解的保持，具体见 2.2 节；步骤 l)将容器 $P_2$ 中的 Pareto 最优解赋值给 $P^*$；步骤 m)输出用户选择的最优解。

### 2.1 Pareto 最优解保留

在动态的环境中，由于算法本身也具有随机性，这就可能导致在优化过程中产生的 Pareto 最优解丢失，引入了辅助种群来对当前产生的 Pareto 最优解实施保留。在进化过程中，通过将每次产生的 Pareto 最优解放入辅助种群来进行保留。Pareto 最优解保留的具体过程详见算法 2。其中 ParetoNum( )用来计算粒子群中的最优解的个数，Maxcapacity 是辅助种群 $P_2(t)$ 的最大容量，SelectPareto( )用来取出粒子群的最优解，DensityDis( )用来计算种群中每个粒子的密集距离[4]，MateSelect($P_2(t)$)、SelectPareto($P_1(t)$)用来将辅助种群 $P_2(t)$ 中的粒子与从 $P_1(t)$ 中的最优非劣粒子混合放入临时容器中，$P_{\mathrm{temp}}(t)\setminus \min(\mathrm{DensityDis}(P_{\mathrm{temp}}(t)))$ 表示从临时容器 $P_{\mathrm{temp}}(t)$ 中删除它里面的密集距离最小的粒子。

**算法 2** Pareto 最优解保留(OptimalSelect)

输入：辅助种群 $P_2(t)$ 和第 $t$ 代种群 $P_1(t)$。
输出：第 $t+1$ 代辅助种群 $P_2(t+1)$。
begin
$P_{\mathrm{temp}}(t)=\varnothing$, num = 0, k = 0
if($P_2(t)=\varnothing$)
{if(ParetoNum($P_1(t)$)≤Maxcapacity)
$P_2(t+1)$ = SelectPareto($P_1(t)$)
else
{$P_{\mathrm{temp}}(t)$ = SelectPareto($P_1(t)$)
while(num < (ParetoNum($P_1(t)$) − Maxcapacity))
{$P_{\mathrm{temp}}(t)=P_{\mathrm{temp}}(t)\setminus\min(\mathrm{DensityDis}(P_{\mathrm{temp}}(t)))$
num = num + 1}
$P_2(t+1)=P_{\mathrm{temp}}(t)$}
$P_{\mathrm{temp}}(t)=\varnothing$}
else
{$P_{\mathrm{temp}}(t)$ = MateSelect($P_2(t)$, SelectPareto($P_1(t)$))
if(ParetoNum($P_{\mathrm{temp}}(t)$)≤Maxcapacity)
$P_2(t+1)$ = SelectPareto($P_{\mathrm{temp}}(t)$)
else
{while(k < ParetoNum($P_{\mathrm{temp}}(t)$) − Maxcapacity)
{$P_{\mathrm{temp}}(t)=P_{\mathrm{temp}}(t)\setminus\min(\mathrm{DensityDis}(P_{\mathrm{temp}}(t)))$
k = k + 1}
$P_2(t+1)=P_{\mathrm{temp}}(t)$}
Output($P_2(t+1)$)
end

### 2.2 全局最优粒子选取

在对种群中的粒子进行速度更新时，需要首先确定该粒子对应的全局最优位置。在这里本文采用如下策略：在迭代前期，对种群中的每个粒子，计算容器中各个粒子的密集距离，并将其作为轮盘赌算法的适应值，从容器中为粒子选择一个全局最优位置。重复步骤，为种群中的每个粒子指定一个全局最优位置，这样可以提高粒子的全局搜索能力。在迭代后期，对种群中的每个粒子，首先计算该粒子与辅助种群中每个粒子在目标空间上的欧几里德距离，并选择辅助种群中与该粒子的欧几

里德距离最小的粒子为该粒子的全局最优粒子。重复上述步骤,可以给种群中的每个粒子指定其全局最优粒子。具体的算法见算法 3。其中 selgbestbasedwheel( )表示利用轮盘赌算法从辅助种群 $P_2(t)$ 中选择一个粒子,OJMDistan( * , * )表示计算两个粒子之间的欧几里德距离,Min( )是取里面的最小值。

**算法 3**　全局最优粒子选取(SelectBestPosition)

输入:最大迭代次数 Step,当前迭代次数 $i$,阈值 $p$(这里取 0.9),当前种群 $P_1(t)$。

输出:更新后的种群 $P_1(t)$。

```
begin
  foreach( a ∈ P1 ( t ) )
  if( ( i/Step) < p)
  a. bestPosition = selgbestbasedwheel( )
  else
  foreach( b ∈ P2 ( t ) )
  a. bestPosition = Min( OJMDDistan( a , b) )
  output( P1 ( t ) )
end
```

## 3　实验及分析

实验环境为 100 Mbps 的局域网,微机的配置为 Pentium 1.7 GHz CPU/2 GB 内存,Windows XP,程序用 C 语言实现。假定 Web 组合服务由 10 个 Web 子服务串联而成,每个服务节点有 10 个候选服务可供选择,每个候选服务的 QoS 参数采用随机方法在一定的范围内生成,粒子速度、位置的更新采用带惯性权重的粒子群算法[5],$w$ 从 0.9 线性下降到 0.4。

### 3.1　算法的有效性

该实验通过算法执行的 CPU 开销来验证算法的有效性。实验分别考虑了粒子群规模为 10、20 和 30,迭代次数为 100、200、300 和 400 的几种情况。对于每一种情况,IDMPSO 算法分别运行 20 次取平均值,结果如图 1 所示。图中横坐标代表迭代次数,纵坐标代表算法执行的时间开销。由图 1 可以看出,随着粒子群规模的增加和迭代次数的增加,CPU 运行时间并没有大量增加,可以满足求解大部分服务组合情况的需要。

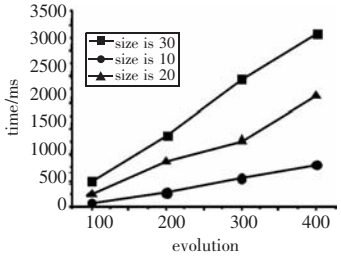


图 1　算法平均执行时间

### 3.2　算法的可行性

本文采用密集距离来进行 Pareto 最优解保留,并提出了一种新的全局最优粒子的选择方法。在这里与文献[6]中提到的 CMPSO 算法进行比较。这里假定迭代次数为 400,粒子群规模分别为 10、20 和 30。对于每一种情况,算法分别运行 20 次取平均值,结果如图 2 所示。由于优良解保持策略和全局最优粒子选取策略的引入,算法 IDMPSO 的时间开销要大于算法 CMPSO 的时间开销。在粒子群规模为 50,迭代次数为 500 的情况下的最优解分布情况如图 3 所示。从图 3 以看出,算法 IDMPSO 的最优解的分布情况要优于算法 CMPSO。尽管算法 IDMPSO 比算法 CMPSO 的时间开销要大一些,但是算法 IDMPSO 能得出比算法 CMPSO 更优的解,并且分布得比较均匀,由此证明了该算法的可行性。

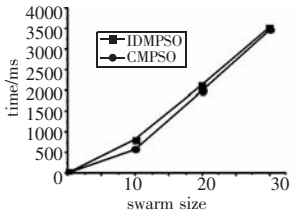


图 2　CMPSO 与 IDMPSO 的平均执行时间

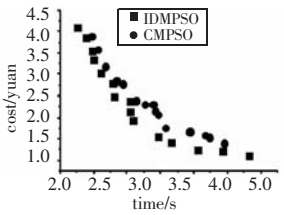


图 3　CMPSO 与 IDMPSO 的优化解集

## 4　结束语

服务选择是服务组合的一个关键问题。针对现有的服务选择算法的不足,本文建立了服务选择问题的多目标优化模型,并提出 Web 服务组合中基于 QoS 的改进型粒子群算法。通过粒子的密集距离来进行优良解的保持,在迭代的前期,对种群中的每个粒子,通过将容器中每个粒子的密集距离作为轮盘赌算法的适应值,为每个粒子选择一个全局最优粒子;在迭代的后期,通过计算种群中每个粒子与容器中每个粒子的欧几里德距离,并将其最小距离者作为其全局最优粒子。最后通过不同参数和不同策略的对比实验,证明了算法的有效性和可行性。今后需要对 IDMPSO 的性能进行进一步的评价,并需要对 IDMPSO 算法动态终止条件、全局最优粒子的寻找方法等方面进行深入研究。

**参考文献:**

[1] 刘莉平,陈志刚,刘爱心. 基于粒子群算法的 Web 服务组合研究[J]. 计算机工程,2008,34(5):104-106,112.

[2] 刘树磊,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] GAREY M R,JOHNSON D S. Computers and intractability:a guide to the theory of NP completeness[M]. New York:FREEMAN W H and Company,1979.

[4] 雷德明,吴智铭. 基于个体密集距离的多目标进化算法[J]. 计算机学报,2005,28(8):1320-1326.

[5] 纪震,廖惠连,吴清花. 粒子群算法及应用[M]. 北京:科学出版社,2009.

[6] 郑友莲,樊俊青. 基于密集距离的多目标粒子群优化算法[J]. 湖北大学学报:自然科学版,2008,30(2):141-144,191.

[7] CASATI F,ILNICKI S,JIN Li-jie,*et al*. eFlow:a platform for developing and managing composition e-services,Technical Report HPL-2000-36[R]. [S. l. ]:HP Laboratories Palo Alto,2000.

[8] ZHAO Jun-feng,XIE Bing,ZHANG Lu,*et al*. A Web services composition method supporting domain feature[J]. Chinese Journal of Computers,2005,28(4):731-738.

[9] LIANZHAO Z,BOUALEM B,ANNE H H, *et al*. QoS-aware middleware for Web services composition[J]. IEEE Trans on Software Engineering,2004,30(5):311-327.

[10] EBERHART R, SHI Yu-hui. Particle swarm optimization:developments,applications and resources[C]//Proc of Congress on Evolutionary Computation. Washington:IEEE Computer Society,2001:81-86.

[11] YU Tao,LIN K J. The design of QoS broker algorithms for QoS-capable Web services[C]//Proc of IEEE International Conference on e-technology e-commerce and e-service. Washington:IEEE Computer Society,2004:17-24.

[12] LIU Yu-tu,ANNE H H,ZENG L Z. QoS computation and policing in dynamic Web service selection[C]//Proc of the 13th International Word Wide Web Conference on Alternate Track Papers & Posters. New York:ACM Press,2004:66-73.

[13] AGGARWAL R,VERMA K,MILLER J,*et al*. Constraint driven Web service coposition in METEOR-S[C]//Proc of IEEE International Conference on Services Computing. [S. l. ]:Springer-Verlag,2004:23-30.

[14] BOSMAN P A N. The balance between proximity and diversity in multiobjective evolutionary algorithms[J]. IEEE Trans on Evolutionary Computation,2003,7(2):174-188.

[15] 龚小勇,朱庆生,武春岭. Web 服务组合中基于 QoS 的改进型遗传算法[J]. 计算机应用研究,2008,25(10):2922-2924,2961.

[16] 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.