

# 求解 Web 服务组合 QoS 优化的多属性决策及自适应遗传算法

鲁城华<sup>1,2</sup> 寇纪淞<sup>1</sup>

(天津大学管理与经济学部 天津 300072)<sup>1</sup> (天津财经大学珠江学院 天津 301811)<sup>2</sup>

**摘要** 随着面向服务计算(Service-oriented Computing, SOC)的不断发展,基于服务质量(Quality of Service, QoS)的 Web 服务组合研究成为了必然趋势。鉴于 QoS 属性的多维性及相互矛盾性,提出将基于 QoS 的 Web 服务组合优化问题转化为多属性决策问题进行求解。采用折中系数对每个组合服务实例到正负理想点的距离进行累加求和,最终得到一组最优服务排序结果,用户可以根据自身偏好进行选择。传统的多属性决策方法无法有效地处理大规模的组合服务搜索空间,因此,为了有效地解决 Web 服务组合优化这一 NP 难题,提出一种结合多属性决策方法和自适应遗传算法的新型优化算法来解决该问题。实验采用真实的 QoS 综合服务数据集进行验证,实验结果表明,该方法能够在较短时间内找到全局近似最优解,且解集的排序结果接近于实际的最优服务排序。同时,该方法对于解决大规模的 Web 服务组合优化问题具有良好的可伸缩性。

**关键词** Web 服务组合, QoS, 多属性决策, 遗传算法

中图分类号 TP301 文献标识码 A DOI 10.11896/j.issn.1002-137X.2019.02.029

## Multi-attribute Decision Making and Adaptive Genetic Algorithm for Solving QoS Optimization of Web Service Composition

LU Cheng-hua<sup>1,2</sup> KOU Ji-song<sup>1</sup>

(College of Management and Economics, Tianjin University, Tianjin 300072, China)<sup>1</sup>

(College of Pearl River, Tianjin University of Finance and Economics, Tianjin 301811, China)<sup>2</sup>

**Abstract** With the increasing of service-oriented computing, the research on Web service composition based on quality of service (QoS) becomes an inevitable trend. With respect of the multi-dimensional nature and mutual contradiction, this paper transformed the optimization of Web service composition based on QoS into the problem of multi-attribute decision making to resolve it. The distances of each solution to the positive ideal solution (PIS) and the negative ideal solution (NIS) were summed up by means of a compromise coefficient. Finally, a set of ranked Web services were provided to users for a flexible choice. The traditional multi-attribute decision making method can not effectively solve the large-scale search space of Web service composition. Therefore, in order to solve the NP-hard problem of Web service composition optimization better, this paper developed an approach combining the multi-attribute decision making and adaptive genetic algorithm (MADMAGA). The experiments were conducted on a real and comprehensive QoS dataset. The experimental results indicate that the method can find the globally optimal solution in a short period of time. The ranking result of solutions is close to the true sort. Moreover, the proposed method has better scalability for solving the large-scale problem of Web service composition optimization.

**Keywords** Web service composition, Quality of service, Multi-attribute decision making, Genetic algorithm

## 1 引言

随着云计算时代的到来,众多企业和个人将其资源以 Web 服务的形式发布到互联网上进行共享,例如各种软件资源和计算资源<sup>[1]</sup>。Google, Amazon, Microsoft, Salesforce 等提供了大量的 Web 服务。

通常,单个 Web 服务无法很好地满足用户需求,需要将多个服务进行组合<sup>[2-3]</sup>,如一个旅游计划服务包括酒店预定、

车票预定、行程安排、天气情况查询以及车辆租赁等子服务。随着 Web 服务数量的剧增,很多功能相同或相似的服务相继出现,组合服务中的每个子任务都可以由多个候选服务完成。因此,用户在提交服务请求时,除了功能需求,还包括非功能需求,即服务质量,例如服务价格、响应时间、可靠性和可用性等<sup>[4]</sup>。比如,用户希望服务价格不超过 \$ 100、响应时间低于 1h、可靠性高于 90% 等。

因此,出现了一个新的研究课题,基于 QoS 的 Web 服务

到稿日期:2018-03-30 返修日期:2018-05-19 本文受国家自然科学基金重点项目(71631003),国家自然科学基金面上项目(71101103)资助。  
鲁城华(1985-),女,博士生,主要研究方向为管理信息系统、数据挖掘,E-mail:iamluchenghua@sina.com;寇纪淞(1947-),男,博士,教授,主要研究方向为信息系统与信息管理等,数据挖掘与知识管理,E-mail:jkskou@tju.edu.cn(通信作者)。

组合,即为组合服务的每个子任务选择合适的 Web 服务,使整个组合服务的 QoS 值满足用户需求。假设一个 Web 服务组合计划由  $n$  个子任务构成,每个子任务可以由  $m$  个功能相同的候选服务完成,则将会有  $m^n$  个组合服务供用户选择。随着子任务和候选服务数量的不断增加,组合 Web 服务的数量呈指数级增长,因此基于 QoS 的 Web 服务组合问题成为了一个 NP 难题。

在实际应用中,用户面临的往往是多维 QoS 属性,因此,为了有效解决 Web 服务组合优化这一 NP 难题,本文提出了一种结合多属性决策方法和自适应遗传算法的新型优化算法。首先,根据 QoS 属性构建相应的多属性决策矩阵,建立求解 Web 服务组合问题的多属性决策模型。在模型的创建过程中,分别考虑了不同 QoS 的主客观权重对组合服务质量的影响。针对组合服务寻优计算,本文提出了一种自适应遗传算法。该方法一方面利用了遗传算法的全局寻优性,另一方面提出了一种局部搜索策略来加快算法的收敛速度,提高计算效率。最后,通过计算组合服务到正理想点(Positive Ideal Solution, PIS)和负理想点(Negative Ideal Solution, NIS)的距离累加值,得到一组最优组合服务排序。无论用户是 PIS 偏好还是 NIS 偏好,都可以根据自身的情况进行选择。

相对于传统的 Web 服务组合优化方法,本文方法能够有效处理多个 QoS 属性以及属性之间的冲突性,且具有全局寻优性和较高的计算效率。

本文第 2 节介绍 Web 服务组合的研究现状;第 3 节介绍组合服务的 QoS 计算过程;第 4 节介绍 QoS 多属性决策模型的创建;第 5 节介绍本文所提出的自适应遗传算法;第 6 节对本文方法进行实验验证,并对结果进行了分析;最后总结本文。

## 2 研究现状

随着 Web 服务数量的剧增,基于 QoS 的 Web 服务组合优化成为了研究热点<sup>[5-6]</sup>,很多学者提出了不同的解决方法,包括启发式方法、最优模型方法及智能算法。

启发式方法从 QoS 局部角度选择最优组合服务,具有代表性的是贪婪算法。文献[7]提出了一种在多项式时间内找到近似最优解的启发式 Web 服务组合方法。该方法将 QoS 的全局约束分解为多个局部约束,为组合服务的每个子任务选择最优的候选服务,将局部最优解组合为全局最优解。文献[8]提出将启发式方法和模拟退火方法相结合来解决 QoS 动态服务组合问题,对候选服务进行分类,采用改进的模拟退火算法获取满足用户 QoS 需求的组合服务。文献[9]提出了一种两阶段启发式算法来解决基于 QoS 的 Web 服务组合优化问题,将蚁群算法和遗传算法相结合,通过优化多个目标函数最终得到一组满足 QoS 约束的最优解集。启发式方法有助于解决规模较小的 Web 服务组合问题,而对于 Web 服务组合优化这一 NP 难题,该方法并不能高效地得到最优解,且该方法从局部最优的角度解决 Web 服务组合优化问题,并不能保证得到全局最优解。

为了更好地从全局角度解决 Web 服务组合优化问题,大量基于最优模型的方法被提出<sup>[10-14]</sup>。该方法将 Web 服务组合优化转化为已知的优化模型,比如整数线性规划模型。文

献[10]将 Web 服务组合转化为整数线性规划模型,结合多目标方法,将 5 个 QoS 属性同时进行优化,最终得到满足用户需求的服务。文献[11]将 Web 服务组合优化转化为模糊线性规划问题,考虑了用户的期望和偏好信息,以便更好地找到满足用户需求的组合服务。文献[12]采用 3 种约束因子来调节服务组合的效果,包括严格约束、适中约束和松弛约束,将不同的 QoS 属性按照其权值综合计算,得到满足全局约束的最优服务,并提供给用户。最优模型方法从全局角度出发,通过模型优化得到组合服务最优解。然而,随着 Web 服务数量的剧增,其时间复杂度将以指数级增长。此外,这种方法忽略了 QoS 之间的冲突性,比如价格和响应时间,价格越高,响应时间通常越低,反之亦然。其计算结果具有一定的片面性且容易丢失大量的决策信息。该方法需要用户提供相应的 QoS 约束,而用户可能并不熟悉 Web 服务环境,因此无法准确地根据其需求提供相应的 QoS 约束值,从而导致计算结果不能更好地满足用户需求。

为了更好地解决 Web 服务组合优化这一 NP 难题,大量智能算法被提出,例如遗传算法<sup>[15-16]</sup>、粒子群优化算法<sup>[17]</sup>及蚁群算法<sup>[18]</sup>。文献[16]提出采用遗传算法解决动态 Web 服务组合问题,考虑了服务组合权重的不固定性,采用罚函数策略动态调整权重值,有利于更好地找到符合用户需求的组合服务。文献[17]采用粒子群优化算法解决动态 Web 服务组合问题,首先剔除冗余候选服务,然后通过提高粒子群多样性来增强算法的全局搜索能力以得到最优组合服务。文献[18]提出采用改进的蚁群算法对多个 QoS 属性进行优化,最终得到一组候选服务组合供用户选择。其中,在解决组合优化问题时遗传算法被认为是最有效的算法,它能够从所有可能的组合中搜索到满足全局约束的解,因此在解决 Web 服务组合优化问题时被广泛使用。

不同于以往的将多个 QoS 累加为单一值的方法,一些研究人员尝试从多重 QoS 角度出发对候选服务进行评估。Web 服务组合问题被转化为多属性决策问题(Multi-attribute Decision Making, MADM)<sup>[19-21]</sup>或者多目标优化问题(Multi-objective Optimization Problem, MOP)<sup>[22-24]</sup>。MADM 方法主要包含层次分析法、理想点法(Technique for Order Preference by Similarity to Ideal Solution, TOPSIS)及折衷比例法。文献[19]将层次分析法用于 Web 服务排序,解决了取值和属性类型不同的 QoS 间如何成对比较的问题,提高了服务排序结果。文献[20]采用模糊多属性决策方法解决 Web 服务组合优化问题,对语义 QoS 值进行去模糊处理,然后采用 TOPSIS 方法对服务进行排序从而得到最优解。文献[22]和文献[23]提出采用多目标粒子群方法来解决 Web 服务组合优化问题,同时对多个 QoS 进行优化,最终得到一组帕累托最优解集供用户选择。MADM 方法能够有效地解决多个 QoS 属性的问题,同时考虑了属性之间的冲突性;MOP 方法能够从多维角度为用户提供 QoS 信息,以使用户选择。然而,这两种方法均存在一定的弊端。MADM 方法属于枚举类方法,需要对海量的备选服务方案进行排序和择优,这对于 Web 服务组合这一 NP 难题来说计算效率较低;此外, MADM 方法计算所得的结果并不一定是全局最优,无法保证用户的全局

QoS 约束得到满足。MOP 方法能够有效地解决大规模的 Web 服务组合优化问题,然而当 QoS 个数增多时,算法的复杂度会大幅度增高,导致计算效率降低。

为了更好地解决基于 QoS 的 Web 服务组合优化问题,本文提出一种结合多属性决策方法和自适应遗传算法的新型优化算法 MADMAGA。同时,为了提高计算效率,本文提出一种局部搜索策略来加快算法的收敛速度,最后能够得到一组最优解集,以供用户根据自身需求进行灵活选择。

### 3 组合 Web 服务 QoS 计算

一个组合 Web 服务是由多个子任务通过一定的工作流模式组合而成,每个子任务对应一组功能相同但 QoS 属性值不同的候选服务。Web 服务组合的目标就是从这些候选服务中依次选择一组服务进行组合,从而满足用户的功能需求以及相应的 QoS 约束要求。

图 1 给出了一个组合服务实例,该实例包含  $n$  个子任务  $\{S_1, S_2, \dots, S_n\}$ ,每个子任务可以由一组候选服务  $\{s_{11}, s_{12}, \dots, s_{1m}\}$  来完成。组合服务包含不同的工作流模式,最基本 4 种工作流是顺序、选择、并行和循环<sup>[25]</sup>,如图 2 所示。

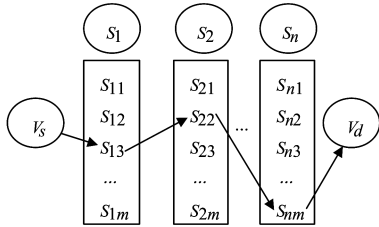


图 1 组合服务实例

Fig. 1 Example of composite service

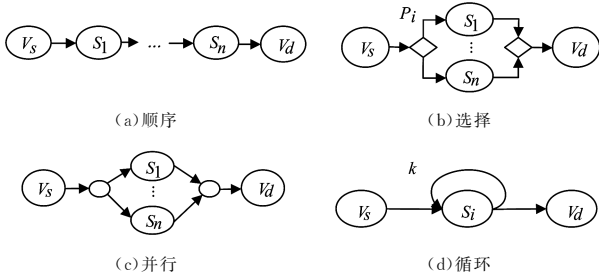


图 2 4 种基本工作流

Fig. 2 Four basic workflows

组合服务中 4 种基本工作流的详细描述如下。

1) 顺序:如图 2(a)所示,在顺序性的组合服务工作流中,按顺序依次执行每个子任务。

2) 选择:如图 2(b)所示,在选择性的组合服务工作流中存在多个选择分支,每个分支的任务以概率  $P_i$  被选择性地执行。

3) 并行:如图 2(c)所示,在并行性的组合服务工作流中存在多个并行的分支,每个并行分支的任务同时被执行,然后再汇入下一个任务中。

4) 循环:如图 2(d)所示,在循环性的组合服务工作流中,某个任务被循环执行  $k$  次之后再继续往下执行其他任务。

价格、响应时间、可靠性和可用性是 4 种最常见的 QoS

属性。组合 Web 服务的 QoS 值是由构成该服务的各个原子服务的 QoS 值聚合而成。不同的 QoS 在不同的工作流下具有不同的聚合公式<sup>[13]</sup>,如表 1 所列。

表 1 不同工作流模式下的 QoS 聚合函数

Table 1 Aggregation functions of QoS attributes for different workflows

QoS	顺序	选择	并行	循环
价格	$\sum_{i=1}^n q(s_i)$	$\sum_{i=1}^n p_i \cdot q(s_i)$	$\sum_{i=1}^n q(s_i)$	$k \cdot q(s_i)$
响应时间	$\sum_{i=1}^n q(s_i)$	$\sum_{i=1}^n p_i \cdot q(s_i)$	$\max_{i=1, \dots, n} \{q(s_i)\}$	$k \cdot q(s_i)$
可靠性	$\prod_{i=1}^n q(s_i)$	$\sum_{i=1}^n p_i \cdot q(s_i)$	$\prod_{i=1}^n q(s_i)$	$q(s_i)^k$
可用性	$\prod_{i=1}^n q(s_i)$	$\sum_{i=1}^n p_i \cdot q(s_i)$	$\prod_{i=1}^n q(s_i)$	$q(s_i)^k$

表 1 中,  $q(s_i)$  表示原子服务  $s_i$  的 QoS 值,  $k$  表示循环工作流中任务的迭代次数,  $p_i$  表示选择性工作流中选择分支的概率。需要注意的是,价格和响应时间的聚合函数为累加型,可靠性和可用性的聚合函数在顺序、并行、循环工作流中是累积型,选择性工作流的 4 个 QoS 聚合函数是相同的。这种 QoS 聚合方式同样可以应用于其他 QoS 属性,对组合服务来说,其 QoS 值可以由多种工作流聚合而成。

### 4 QoS 多属性决策模型的创建

用户提交的服务请求通常包含多个 QoS 约束,比如价格、响应时间、可靠性、可用性、吞吐量等。本节将采用 MADM 方法对多个 QoS 属性进行计算。

QoS 属性主要包括两种类型:成本型和效益型。前者主要包括响应时间、价格等属性,后者主要包括可靠性、可用性等属性。当服务的效益型 QoS 值越高、成本型 QoS 值越低时,该服务的性能就越好。基于 QoS 的 Web 服务组合优化的目标是尽量最大化效益型 QoS 的值,而最小化成本型 QoS 的值。

假设一个原子服务  $s$  包含  $x$  个效益型属性和  $y$  个成本型属性,如式(1)所示:

$$q(s) = \langle q_{b_1}(s), q_{b_2}(s), \dots, q_{b_x}(s), q_{c_1}(s), q_{c_2}(s), \dots, q_{c_y}(s) \rangle \quad (1)$$

其中,  $q(s)$  表示原子服务  $s$  的 QoS 属性集合,  $q_{b_i}$  表示效益型属性,  $q_{c_i}$  表示成本型属性。

同样,组合服务  $cs$  的 QoS 模型如式(2)所示:

$$Q(cs) = \langle Q_{b_1}(cs), Q_{b_2}(cs), \dots, Q_{b_x}(cs), Q_{c_1}(cs), Q_{c_2}(cs), \dots, Q_{c_y}(cs) \rangle \quad (2)$$

多属性决策模型包括 4 个部分:候选组合服务集合、QoS 属性向量、QoS 属性权重向量、组合服务 QoS 矩阵,如式(3)所示:

$$MADM = (CS, Q, W, DM) \quad (3)$$

其中,  $CS = \{CS_1, CS_2, \dots, CS_l\}$  表示  $l$  个候选服务的集合;  $Q = \langle Q_{b_1}, Q_{b_2}, \dots, Q_{b_x}, Q_{c_1}, Q_{c_2}, \dots, Q_{c_y} \rangle$  表示服务的所有 QoS 属性;  $W = \langle w_{b_1}, w_{b_2}, \dots, w_{b_x}, w_{c_1}, w_{c_2}, \dots, w_{c_y} \rangle$  表示每个 QoS 属性的权重,这个值将在 4.2 节中进行计算;  $DM = [d_{ij}]_{l \times (x+y)}$  表示一个大小为  $l \times (x+y)$  的多属性决策矩阵,元素  $d_{ij}$  表示第  $i$  个候选服务在第  $j$  个 QoS 上的值,该矩阵的表示如下:

$$DM = [d_{ij}]_{l \times (x+y)}$$

$$= \begin{matrix} & Q_{b_1} & Q_{b_2} & \cdots & Q_{b_x} & Q_{c_1} & Q_{c_2} & \cdots & Q_{c_y} \\ CS_1 & \left[ \begin{matrix} d_{1b_1} & d_{1b_2} & \cdots & d_{1b_x} & d_{1c_1} & d_{1c_2} & \cdots & d_{1c_y} \end{matrix} \right] \\ CS_2 & \left[ \begin{matrix} d_{2b_1} & d_{2b_2} & \cdots & d_{2b_x} & d_{2c_1} & d_{2c_2} & \cdots & d_{2c_y} \end{matrix} \right] \\ \vdots & \left[ \begin{matrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{matrix} \right] \\ CS_l & \left[ \begin{matrix} d_{lb_1} & d_{lb_2} & \cdots & d_{lb_x} & d_{lc_1} & d_{lc_2} & \cdots & d_{lc_y} \end{matrix} \right] \end{matrix}$$

$$i=1,2,\dots,l; j=b_1, b_2, \dots, b_x, c_1, c_2, \dots, c_y \quad (4)$$

根据矩阵  $DM$ , 将其每一行的候选服务进行比较, 选择出一组排序的最优组合服务提供给用户进行选择。下文将详细介绍该多属性决策矩阵的计算过程。

#### 4.1 构建标准化决策矩阵

在实际应用中, 不同的 QoS 属性具有不同的数据类型, 例如, 可靠性和可用性属于比率型 QoS, 价格和响应时间属于数值型 QoS。比率型 QoS 的变化范围是有限的, 通常为  $[0, 100\%]$ , 而数值型 QoS 可能存在不同的数据范围, 比如响应时间, 有的用户的响应时间的范围可能是  $[0\text{ s}, 1\text{ s}]$ , 而有的用户响应时间的范围可能是  $[10\text{ s}, 20\text{ s}]$ 。取值范围的不同会对组合服务的 QoS 计算产生一定影响, 因此本文将对数值型的 QoS 数据做规范化处理。本文将采用线性归一化方法将不同范围的 QoS 数据转换到  $[0, 1]$  之间, 如式(5)所示:

$$r_{ij} = \frac{d_{ij}}{\sum_{i=1}^l d_{ij}} \quad (5)$$

经过线性归一化处理之后, 矩阵  $DM$  被转化为标准化决策矩阵  $R$ , 矩阵中的每一列值之和为 1, 如式(6)所示:

$$R = [r_{ij}]_{l \times (x+y)}$$

$$= \begin{matrix} & Q_{b_1} & Q_{b_2} & \cdots & Q_{b_x} & Q_{c_1} & Q_{c_2} & \cdots & Q_{c_y} \\ CS_1 & \left[ \begin{matrix} r_{1b_1} & r_{1b_2} & \cdots & r_{1b_x} & r_{1c_1} & r_{1c_2} & \cdots & r_{1c_y} \end{matrix} \right] \\ CS_2 & \left[ \begin{matrix} r_{2b_1} & r_{2b_2} & \cdots & r_{2b_x} & r_{2c_1} & r_{2c_2} & \cdots & r_{2c_y} \end{matrix} \right] \\ \vdots & \left[ \begin{matrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{matrix} \right] \\ CS_l & \left[ \begin{matrix} r_{lb_1} & r_{lb_2} & \cdots & r_{lb_x} & r_{lc_1} & r_{lc_2} & \cdots & r_{lc_y} \end{matrix} \right] \end{matrix}$$

$$i=1,2,\dots,l; j=b_1, b_2, \dots, b_x, c_1, c_2, \dots, c_y \quad (6)$$

线性归一化处理不会改变 QoS 的特性, 也就是说, 前  $m$  个依然是效益型属性, 后  $n$  个是成本型属性。

#### 4.2 对标准化决策矩阵进行加权

不同的用户对 QoS 属性有不同的偏好和重视程度, 因此将对其进行赋权处理, 如式(7)所示:

$$v_{ij} = \omega_j \cdot r_{ij} \quad (7)$$

其中,  $\omega_j$  代表第  $j$  个 QoS 属性的权重, 本文将根据用户的主观权重和由决策矩阵计算出的客观权重来共同计算  $\omega_j$  值。所有 QoS 属性的权值如式(8)、式(9)所示,  $\omega_s$  代表用户的主观权重,  $\omega_o$  代表客观权重。

$$\omega_s = \langle \omega_{sb_1}, \omega_{sb_2}, \dots, \omega_{sb_x}, \omega_{sc_1}, \omega_{sc_2}, \dots, \omega_{sc_y} \rangle \quad (8)$$

$$\omega_o = \langle \omega_{ob_1}, \omega_{ob_2}, \dots, \omega_{ob_x}, \omega_{oc_1}, \omega_{oc_2}, \dots, \omega_{oc_y} \rangle \quad (9)$$

本文将采用信息熵法<sup>[26]</sup>来计算 QoS 的客观权重, 如式(10)所示:

$$\omega_{oj} = \frac{1 - E_j}{\sum_{k=b_1, b_2, \dots, b_x, c_1, c_2, \dots, c_y} (1 - E_k)}$$

$$E_j = -\frac{1}{\ln l} \sum_{i=1}^l r_{ij} \ln r_{ij}, j=b_1, b_2, \dots, b_x, c_1, c_2, \dots, c_y \quad (10)$$

其中,  $E_j$  代表第  $j$  个 QoS 属性的信息熵, 其含义是所有候选

服务第  $j$  个 QoS 属性值的相似度。当  $r_{ij}$  的值同时为  $1/l$  时,  $E_j$  具有最大值,  $E_j$  值越大, 各候选服务在第  $j$  个 QoS 属性上的差异就越小。这进一步说明该属性对于区分候选服务所起到的作用非常小, 因此, 应该赋予这个属性非常小的权重, 反之亦然。信息熵法可以直接从决策矩阵中计算得到 QoS 的客观权重, 计算简便且不需要额外的决策变量, 对 QoS 的微小变动也具有很强的鲁棒性, 比较适合计算客观权重值。

主观权重可以反映用户的主观判断和偏好, 但会因为用户的主观随意性或者缺乏经验而带来片面影响。客观权重虽然是基于完善的数学理论计算得到的, 但却忽略了用户的主观判断和偏好。考虑到这一点, 本文将主客观权重相结合, 用参数  $\lambda$  来平衡两种权重的比例,  $\lambda$  值越大, 用户的主观权重的影响就越大。为了便于计算, 本文取经验值  $\lambda = 0.5$ , 如式(11)所示:

$$\omega = \lambda \omega_s + (1 - \lambda) \omega_o, \lambda \in [0, 1] \quad (11)$$

通过对 QoS 属性赋权之后, 将标准化决策矩阵转化为赋权标准化决策矩阵  $V$ :

$$V = [v_{ij}]_{l \times (x+y)}$$

$$= \begin{matrix} & Q_{b_1} & Q_{b_2} & \cdots & Q_{b_x} & Q_{c_1} & Q_{c_2} & \cdots & Q_{c_y} \\ CS_1 & \left[ \begin{matrix} v_{1b_1} & v_{1b_2} & \cdots & v_{1b_x} & v_{1c_1} & v_{1c_2} & \cdots & v_{1c_y} \end{matrix} \right] \\ CS_2 & \left[ \begin{matrix} v_{2b_1} & v_{2b_2} & \cdots & v_{2b_x} & v_{2c_1} & v_{2c_2} & \cdots & v_{2c_y} \end{matrix} \right] \\ \vdots & \left[ \begin{matrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{matrix} \right] \\ CS_l & \left[ \begin{matrix} v_{lb_1} & v_{lb_2} & \cdots & v_{lb_x} & v_{lc_1} & v_{lc_2} & \cdots & v_{lc_y} \end{matrix} \right] \end{matrix}$$

$$i=1,2,\dots,l; j=b_1, b_2, \dots, b_x, c_1, c_2, \dots, c_y \quad (12)$$

#### 4.3 正负理想点的定义

本节将根据决策矩阵  $V$  中每一列的极值来定义正负理想点。正理想点 PIS 也称为最优方案, 表示为  $v^+ = \langle v_{b_1}^+, v_{b_2}^+, \dots, v_{b_x}^+, v_{c_1}^+, v_{c_2}^+, \dots, v_{c_y}^+ \rangle$ 。负理想点 NIS 也称为最劣方案, 表示为  $v^- = \langle v_{b_1}^-, v_{b_2}^-, \dots, v_{b_x}^-, v_{c_1}^-, v_{c_2}^-, \dots, v_{c_y}^- \rangle$ 。其中,  $v_j^+$  和  $v_j^-$  ( $j=b_1, b_2, \dots, b_x, c_1, c_2, \dots, c_y$ ) 的定义如式(13)所示:

$$v_j^+ = \begin{cases} \max(v_{ij}), & j=b_1, b_2, \dots, b_x \\ \min(v_{ij}), & j=c_1, c_2, \dots, c_y \end{cases} \quad (13)$$

$$v_j^- = \begin{cases} \min(v_{ij}), & j=b_1, b_2, \dots, b_x \\ \max(v_{ij}), & j=c_1, c_2, \dots, c_y \end{cases}$$

当第  $j$  个 QoS 为效益型属性时,  $v_j^+$  取第  $j$  列的最大值; 如果是成本型属性, 则  $v_j^+$  取第  $j$  列的最小值,  $v_j^-$  与  $v_j^+$  相反。在现实中, 最优方案和最劣方案并不存在, 或者说至少不会同时存在。因此, 本文中 PIS 和 NIS 方案只是用来评价候选服务的相对优劣程度。

#### 4.4 计算候选服务到正负理想点的欧氏距离

本文将通过计算候选组合服务到正负理想点的欧氏距离来判断解的优劣性。采用欧氏距离的优点是可以扩展到  $n$  维空间, 因此, 该方法比较适合解决多维 QoS 属性的 Web 服务组合问题。

每个组合服务都可以看作是  $x+y$  维空间中的点, 一个点到 PIS 的欧氏距离越近, 该服务就越接近最优方案; 反之, 该点到 NIS 的欧氏距离越近, 该服务就越接近最劣方案。每个点到  $v_j^+$  和  $v_j^-$  的欧氏距离的计算式如下:

$$ED(CS_i, v^+) = \sqrt{\sum_{j=b_1, b_2, \dots, b_x, c_1, c_2, \dots, c_y} (v_{ij} - v_j^+)^2}$$

$$ED(CS_i, v^-) = \sqrt{\sum_{j=b_1, b_2, \dots, b_p, c_1, c_2, \dots, c_p} (v_{ij} - v_j^-)^2}$$

$$i=1, 2, \dots, l \quad (14)$$

可以发现,欧氏距离没有固定的取值范围,它的范围受 QoS 属性个数的影响。为了便于比较,本文将欧氏距离规范化到 $[0, 1]$ 之间。首先,需要找出解空间中的点到 PIS 和 NIS 的距离极值,即极大值和极小值。计算过程如式(15)所示:

$$\begin{cases} ED_{p^+} = \min_{1 \leq i \leq l} (ED(CS_i, v^+)) \\ ED_{p^-} = \max_{1 \leq i \leq l} (ED(CS_i, v^+)) \\ ED_{n^+} = \max_{1 \leq i \leq l} (ED(CS_i, v^-)) \\ ED_{n^-} = \min_{1 \leq i \leq l} (ED(CS_i, v^-)) \end{cases} \quad (15)$$

其中, $ED_{p^+}$ 和 $ED_{p^-}$ 分别是解空间中的所有解到 PIS 的极小值和极大值; $ED_{n^+}$ 和 $ED_{n^-}$ 分别是解空间中的所有解到 NIS 的极大值和极小值。然后,将式(14)中每个解的欧氏距离转化到 $[0, 1]$ 之间,如式(16)所示:

$$\begin{cases} NED(CS_i, v^+) = \frac{ED_{p^-} - ED(CS_i, v^+)}{ED_{p^-} - ED_{p^+}} \\ NED(CS_i, v^-) = \frac{ED(CS_i, v^-) - ED_{n^-}}{ED_{n^+} - ED_{n^-}} \end{cases} \quad (16)$$

其中, $NED(CS_i, v^+)$ 和 $NED(CS_i, v^-)$ 分别是解到 $v_j^+$ 和 $v_j^-$ 的标准化欧氏距离。可以发现,这两个值越大,解到 PIS 的距离就越小,而到 NIS 的距离就越大。为了便于比较和分析,本文将这两个值采用一个系数 $\sigma$ 折合成一个值, $\sigma \in [0, 1]$ ,如式(17)所示:

$$\gamma(CS_i) = \sigma \cdot NED(CS_i, v^+) + (1 - \sigma) \cdot NED(CS_i, v^-) \quad (17)$$

系数 $\sigma$ 代表了决策者不同的风险态度,如果 $\sigma$ 的取值较大,则说明决策者希望组合方案到 PIS 的距离越近,即风险厌恶态度;反之,则说明决策者是风险偏好态度, $\gamma(CS_i)$ 的结果取决于组合方案到 NIS 的距离。最终得到 $\gamma(CS_i)$ 取值最大的解即为最优解,但是,由于 QoS 属性之间往往存在冲突性,最优方案不一定在所有的 QoS 属性上都优于其他方案。因此,本文提供给用户的是一组按照 $\gamma(CS_i)$ 值排序的方案集,用户可以根据自身偏好自由选择。

## 5 求解 Web 服务组合 QoS 优化的自适应遗传算法 MADMAGA

传统的多属性决策方法在寻找最优解时采用的是枚举法,因此只适用于解决有限个离散解的情况。面对大规模的 Web 服务组合问题,本文提出一种自适应遗传算法进行全局动态寻优。

### 5.1 MADMAGA 的基本思想

MADMAGA 同时优化两个种群,主种群 C 和外部种群 E, E 包含了最优个体,即精英个体。从 C 和 E 中分别获取父代个体来生成后代。本文采用了强精英参与策略,主种群中的最优个体能够加入到下一代进化中。

同时,本文提出了一种新的种群更新策略,根据当前种群生成新的种群。有些 QoS 之间存在冲突性,比如价格和响应时间,价格越高,响应时间通常越低。在实际应用中,如果能够适当地提高价格来大幅度降低响应时间,那么这种组合方案通常被认为是可行的。因此,本文在寻找最优解的过

程中引入了 $\epsilon$ 支配关系<sup>[27]</sup>,以保证解的多样性,从而满足用户多样化的需求。在按照适应度对个体进行排序之前,先对解到正负理想点的距离进行 $\epsilon$ 支配关系的判断,以此进行个体的初步筛选。然后再根据个体的适应度值进行排序,每代取前 $k$ 个个体加入到外部种群中。迭代结束之后,再将外部种群的个体进行适应度值排序,将前 $k$ 个最优个体输出给用户。

此外,为了加快算法的收敛速度,提高计算效率,本文提出一种自适应的局部搜索策略。如果每代生成的个体经过支配关系的初步筛选并排序之后最优个体数目小于 $k$ ,则启动局部搜索策略。局部搜索在外部种群中进行,将每个个体的某个基因位进行一次变异,以便在最优个体的邻域内发现其他较优个体。

### 5.2 遗传操作

遗传算法模仿自然界生物进化机制,是一种随机全局搜索和优化的智能算法。遗传操作的设计有利于促进算法的进化,包括精英参与策略、交叉和变异策略。下文将详细介绍本文遗传算法中的编码及进化策略。

#### 5.2.1 染色体编码

在 MADMAGA 中,一个染色体或者个体代表一个候选组合服务,本文将采用实数编码策略来对染色体进行编码。假设候选服务包含 $n$ 个子任务,则一个染色体包含 $n+3$ 个基因位。前 $n$ 个基因位表示所选中的候选服务,用整数表示,第 $i$ 个基因位是所选中的 Web 服务在候选服务集中的排序值。后 3 个基因位分别是该组合服务到正负理想点的标准化欧氏距离以及两者的折中值。染色体的编码方式如下:

$S_1$	$S_2$	...	$S_{n-1}$	$S_n$	$NED(CS, v^+)$	$NED(CS, v^-)$	$\gamma(CS)$
-------	-------	-----	-----------	-------	----------------	----------------	--------------

#### 5.2.2 适应度函数

在遗传算法中,个体的适应度值代表了其对环境的适应能力,适应度值越高越可能繁殖较优后代。在 MADMAGA 算法中,个体的适应度值即为该组合服务到正负理想点的折中值,如式(18)所示:

$$F(CS_i) = \gamma(CS_i) \quad (18)$$

可以发现,适应度值越高的个体,其距离正理想点越近,距离负理想点越远,越能满足用户的需求。

#### 5.2.3 强精英参与策略

本文采用了强精英参与策略,首先分别从当前种群和精英种群中获取父代个体,每代进化得到的后代取排名前 $k$ 的精英个体进入外部种群。当外部种群的个体数目达到限定值 100 时,新个体的进入需采取排序选择策略,排名 100 以后的个体被淘汰进入当前种群。当前种群数目如果超过 100,同样进行排序选择。这种策略有助于提高解的质量,加快收敛。

#### 5.2.4 交叉和变异

交叉操作能够生成新的个体,使算法在整个搜索空间中寻找到全局最优解,本文采取的交叉算是均匀交叉。均匀交叉是指两个个体的每个基因都以相同的交叉概率进行交换,从而生成两个新个体。具体操作如下:首先随机生成一串屏蔽字 $W = W_1 W_2 \dots W_n$ ,其中 $W_i = 0$ 或者 1,由于遗传操作只对染色体的前 $n$ 个基因位进行,因此屏蔽字的长度为组合服务的子任务数 $n$ 。假设两个父代个体是 $p_1$ 和 $p_2$ ,新生成的个

体为  $c_1$  和  $c_2$ 。如果  $W_i=0$ , 则  $c_1$  和  $c_2$  的第  $i$  个基因值分别继承父代  $p_1$  和  $p_2$  相应位置的基因值; 如果  $W_i=1$ , 则  $c_1$  的第  $i$  个基因值继承父代  $p_2$  相应位置的基因值,  $c_2$  的第  $i$  个基因值继承父代  $p_1$  相应位置的基因值。

变异操作是通过对该体的某些基因位上的值做变动来增加种群的多样性。本文采用柯西变异算子, 如式(19)所示:

$$c_i = p_i + C(0, t) \quad (19)$$

$$f_i(x) = \frac{1}{\pi} \cdot \frac{t}{t^2 + x^2}, -\infty < x < \infty$$

其中,  $p_i$  是父代个体  $p$  的第  $i$  个基因位;  $c_i$  是所生成的子代个体  $c$  的第  $i$  个基因位;  $C(0, t)$  是由以 0 为中心、尺度参数为  $t$  的柯西分布函数所生成的随机数;  $t$  为大于 0 的尺度参数, 本文中  $t$  的取值为 1。

### 5.3 MADMAGA 算法设计

MADMAGA 算法的伪代码如算法 1 所示, 主要包括 5 个阶段: 1) 初始化阶段(第 1—2 行); 2) 选择-生成阶段(第 4—18 行); 3) 种群更新阶段(第 19—20 行); 4) 局部搜索阶段(第 21—23 行); 5) 最终排序阶段(第 25 行)。

#### 算法 1 MADMAGA 算法

1. 初始化主种群 C
2. 从主种群 C 中生成外部种群 E,  $E \leftarrow \text{UpdateE}(C)$
3. For  $i=1$  to  $nGen$  do
4.   For  $j=1$  to  $N_C$ , do
5.      $r_c = \text{rand}(1)$
6.     If  $r_c \leq P_c$ , then
7.       从 C 和 E 中随机选择两个父代个体  $p_1$  和  $p_2$  进入交配池
8.       If  $p_1 = p_2$ , then
9.         从 E 中重新选择一个个体
10.      End If
11.      对  $p_1$  和  $p_2$  进行交叉操作生成新个体  $c_1$  和  $c_2$
12.      End If
13.       $r_m = \text{rand}(1)$
14.      If  $r_m \leq P_m$ , then
15.       从 E 中随机选择一个个体  $p_3$  进入交配池
16.       对  $p_3$  进行变异操作生成新个体  $c_3$
17.      End If
18.    End For
19. 更新当前种群 C
20.  $E \leftarrow \text{UpdateE}(C)$ , 将 C 中前  $k$  个最优个体加入 E
21. If  $\text{Num}(\text{UpdateE}(C)) < k$ , then
22.    启动局部搜索, 将 E 中个体的某个基因位进行变异并更新 E
23. End If
24. End For
25. 对 E 进行非支配筛选并排序, 输出前  $k$  个最优个体

在初始化阶段, 首先随机生成  $N_C$  个个体组成主种群 C, 本文中主种群的规模为 100, 然后提取 C 中所有的非支配个体并按照适应度值进行排序, 取前  $k$  个个体组成外部种群 E, 种群更新过程  $\text{UpdateE}()$  如算法 2 所示。在选择-生成阶段, 从 C 和 E 中分别随机选择一个父代个体, 对其进行交叉和变异操作生成后代个体, 其中  $P_c$  和  $P_m$  分别表示交叉概率和变异概率。

### 算法 2 UpdateE(C)算法

1. Procedure UpdateE(C)
2. 计算 C 中个体到正负理想点的标准化欧氏距离以及折中值
3. 根据两个标准化欧氏距离判断个体之间的  $\epsilon$  支配关系, 选出所有非支配个体
4. 将非支配个体按照适应度值进行排序
5. 选出前  $k$  个个体加入外部种群 E 中
6. End Procedure

种群更新阶段是将每代所生成的个体中的前  $k$  个最优个体加入到外部种群 E 中。如果最优个体数目不足  $k$  个, 则启动局部搜索策略, 加快算法收敛。最终排序阶段将外部种群 E 中的个体进行非支配筛选并按适应度值排序, 输出前  $k$  个个体给用户进行选择。

## 6 实验分析

为了评价本文所提出的 MADMAGA 算法, 设计了 3 组实验。首先, 验证了在对个体进行初步筛选时判断  $\epsilon$  支配关系中  $\epsilon$  的取值对解的影响。其次, 将 MADMAGA 与传统多属性决策方法 TOPSIS 的计算性能进行了对比分析。最后, 评估了 MADMAGA 与 TOPSIS 的可扩展性。实验使用 Matlab R2010b 编程, 运行环境为 Windows 7, Intel Pentium 4×3.4 GHz, 内存 4 GB。

### 6.1 实验配置

#### 6.1.1 数据集

本文采用一个通用的综合服务数据集 QWS\_Dataset\_v2<sup>[28]</sup>, 该数据集包含了真实世界中 2507 个 Web 服务的 9 个 QoS 属性的值, 主要包括 response time, availability, throughput, successability, reliability, compliance, best practices, latency 以及 documentation, 其既包括效益型属性, 也包括成本型属性。关于 QWS\_Dataset\_v2 数据集的详细介绍请参考文献[28]。

组合服务中的选择、并行和循环都可以转化为顺序结构, 因此本文实验将主要考虑由顺序结构组成的组合服务流程。本文预先设定一个组合服务流程, 假设其包含  $N_T$  个子任务, 每个子任务对应的候选服务数为  $N_S$ 。

#### 6.1.2 参数设置

实验中所用到的主要参数如表 2 所列。

表 2 参数设置及含义

参数	描述	参数取值
C	主种群	—
E	外部种群	—
$N_C$	主种群的规模	100
$N_E$	外部种群的规模	100
$nGen$	进化代数	1000
$N_T$	组合服务流程中的子任务数	2~10
$N_S$	每个子任务绑定的候选服务数	10~50
$P_c$	交叉概率	0.9
$P_m$	变异概率	0.05~0.1
$k$	每代选取的精英个体数	10

#### 6.1.3 评价指标

本文在评价 MADMAGA 算法计算得到的最优解集时, 将其与最常用的多属性决策方法对比标准进行排序相似性对比,

该方法为简单线性加权法 (Simple Additive Weighting, SAW)<sup>[29]</sup>。实验主要采用了 5 个相关性评价指标: Precision, Recall, PCC (Pearson's Correlation Coefficient), SRCC (Spearman's Rank Correlation Coefficient), NDCG (Normalized Discounted Cumulative Gain)。Precision 和 Recall 分别指 MADMAGA 对 SAW 排名前 10 的个体的查准率和查全率; PCC 和 NDCG 均用来度量 SAW 中标准排名的个体与其在 MADMAGA 中排名的接近程度; SRCC 用来衡量 SAW 与 MADMAGA 方法中排名前 10 的个体的排名相似性。Precision 指标和 Recall 指标的计算公式如式(20)所示:

$$Precision = \frac{R_{SAW}}{All_{MADMAGA}}, Recall = \frac{R_{SAW}}{All_{SAW}} \quad (20)$$

其中,  $R_{SAW}$  代表 MADMAGA 所找到的属于 SAW 排名前 10 的个体数目,  $All_{MADMAGA}$  代表 MADMAGA 找到的所有解,  $All_{SAW}$  代表 SAW 排名前 10 的个体。

$$PCC = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}} \quad (21)$$

其中,  $n$  代表所评价的数据个数,  $X_i$  和  $Y_i$  分别表示两组需要比较的数据,  $\bar{X}$  和  $\bar{Y}$  分别表示这两组数据的平均值。

$$NDCG_k = \frac{DCG_k}{IDCG_k}, DCG_k = \sum_{i=1}^k \frac{2^{R_i} - 1}{i \log_2(i+1)} \quad (22)$$

其中,  $R_i$  表示两者排名的相关性等级, 取值为  $[0, 3]$  之间的整数。当第  $i$  个方案在两者中排名完全一致时, 取  $R_i$  的值为 3; 当两者排名之差的绝对值在  $[1, 3]$  之间时, 取  $R_i$  的值为 2; 当两者排名之差的绝对值在  $[4, 6]$  之间时, 取  $R_i$  的值为 1; 当两者排名之差的绝对值大于 6 时, 取  $R_i$  的值为 0。IDCG 是在完美排序下  $k$  具有的最大 DCG 值, 本文中取  $k$  的值为 10。

$$SRCC = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)} \quad (23)$$

其中,  $n$  代表所评价的数据个数,  $d_i$  代表两组数据的秩次值之差。在 SRCC 的计算过程中, 首先分别对两组数据从大到小进行排序, 对排序后的每个数据标注上秩次值, 该值就是此数据在原来排序中的位置值, 对于相同的数据则取其平均位置值作为秩次值。

## 6.2 不同的 $\epsilon$ 取值对 MADMAGA 有效性的影响

由于 QoS 的冲突性, 本文在 MADMAGA 中引入了  $\epsilon$  支配关系的判断, 以提高解的多样性。然而,  $\epsilon$  取值过大容易导致大量解的丢失。因此, 本实验将分别设定不同的  $\epsilon$  值来分析其对算法有效性的影响。

$\epsilon$  取值分别为 0, 0.0002, 0.002, 0.02, 0.2, 为了便于分析, 本实验不考虑折中系数  $\sigma$  的影响, 将其取值为 0.5, 其他参数设定如下:  $N_T = 10, N_S = 10, P_m = 0.05$ 。

为了评价本文方法所得解集的质量, 将计算结果与 SAW 方法的标准排名结果进行排序相似性分析, 采用 5 个相关性评价指标进行度量, 分别为 Precision, Recall, PCC, SRCC, NDCG。将本文方法迭代运行 1000 次, 取 10 次运行结果的平均值。图 3 给出了在不同的  $\epsilon$  取值下 MADMAGA 在 5 个评价指标上的计算结果。

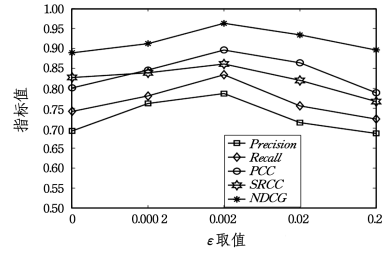


图 3 不同  $\epsilon$  取值下 MADMAGA 和 SAW 最优解的排序相似性对比  
Fig. 3 Sort similarity comparison about optimal solutions of MADMAGA and SAW with different values of  $\epsilon$

从实验结果可以看出, 不同的  $\epsilon$  取值情况下, 5 个相关性指标值均不同,  $\epsilon$  取值从 0 增加到 0.002 时, 5 个指标值均呈现不同程度的升高, 说明解的质量越来越好。然而, 随着  $\epsilon$  取值的继续增大, 5 个指标均呈现不同程度的下降趋势。该结果说明,  $\epsilon$  取 0.002 对本文算法比较适宜, 若过大则容易导致解的丢失, 从而使得最终解集的质量降低。

分别从 5 个指标来看, 当  $\epsilon = 0.002$  时, Precision 和 Recall 的值均大于 0.75, 说明本文方法至少可以找到平均 7.5 个在 SAW 中排名前 10 的个体。PCC 值大于 0.85, NDCG 值在 0.9 以上, 说明 MADMAGA 与 SAW 方法找出的前 10 个解在排名上显著正相关, 且排名靠前的方案的相关性更高。SRCC 值大于 0.85, 说明本文方法与 SAW 方法排名前 10 的相同个体在其排序上也具有非常高的相似性。

综合来看, 当  $\epsilon = 0.002$  时, MADMAGA 的综合性能最优, 在以下实验中将使用此值。

## 6.3 MADMAGA 与 TOPSIS 的计算性能分析

传统的多属性决策方法 TOPSIS 属于枚举型方法, MADMAGA 属于进化寻优类算法。本实验将两者进行对比, 以 SAW 方法为对比标准, 计算两种方法的解集排序与 SAW 的解集排序在 5 个相关性指标上的值。

本实验的参数设定与 6.2 节相同, 其中,  $\epsilon$  取最优值 0.002, 折中系数  $\sigma$  分别取 0, 0.1, 0.2, ..., 1.0, 实验结果同样取 10 次运行结果的平均值。TOPSIS 方法的 Precision, Recall, PCC, SRCC, NDCG 分别为 0.673, 0.738, 0.845, 0.826, 0.877, 本文方法的计算结果如图 4 所示。

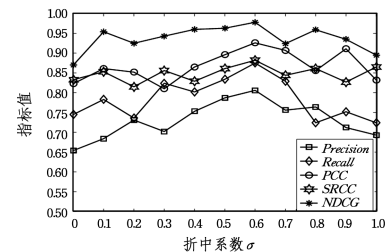


图 4 MADMAGA 和 TOPSIS 的最优解的排序相似性对比  
Fig. 4 Sort similarity comparison about optimal solutions of MADMAGA and TOPSIS

从图 4 可以看出, 无论  $\sigma$  取何值, MADMAGA 在各个指标上的表现基本优于 TOPSIS, 但个别点除外, 比如  $\sigma = 0, 1.0$  时, 其某些指标值略低于 TOPSIS 方法。当  $\sigma = 0.6$  时, 本文方法在各个指标值上均达到最优, 说明大部分决策者持风险

厌恶态度,即希望组合方案到 PIS 的距离越近越好。实验结果表明,MADMAGA 对于解决 Web 服务组合的排序选择问题比传统的多属性决策方法 TOPSIS 更加有效。此外, $\sigma$  的取值过大或过小都会对结果产生一定的影响,0.6 是较为适宜的取值,以下实验将采用此值。

#### 6.4 MADMAGA 与 TOPSIS 的可扩展性对比分析

本文提出的算法有利于解决 Web 服务组合优化这一 NP 难题。首先分析 MADMAGA 的计算复杂度,在初始化阶段完成之后,算法的时间主要消耗在迭代寻优部分,即选择-生成-种群更新-局部搜索阶段。本文中,算法每运行一次将迭代生成  $N_C$  个新个体。在选择-生成阶段,染色体的交叉和变异操作与任务数  $N_T$  有关,该阶段的计算复杂度为  $O(N_C N_T)$ 。在种群更新阶段,计算复杂度主要表现在后代个体的支配筛选和排序上,算法的复杂度为  $O(2N_C^2)$ 。在局部搜索阶段,计算复杂度依赖于外部种群规模  $N_E$ ,为  $O(N_E)$ 。因此,算法整体的计算复杂度为  $O(N_C N_T + 2N_C^2 + N_E)$ 。 $N_E$  的值不断增加,最大不超过 100,因此,算法的最大计算复杂度是当  $N_E = N_C$  时的取值,即  $O(N_C(N_T + 1) + 2N_C^2)$ 。综上,MADMAGA 的计算复杂度主要依赖于主种群的规模以及任务节点数。

TOPSIS 方法是枚举型方法,其计算复杂度与组合服务的备选方案数有关,即  $O(N^2)$ 。 $N$  为备选方案数,该值与任务节点数  $N_T$  和每个子任务的候选服务数  $N_S$  有关,随着两者的增加, $N$  呈指数级增长。

为了对比 MADMAGA 与 TOPSIS 的可扩展性,本文设计了两组实验。第一组实验是当组合服务中的任务数  $N_T$  逐渐增长时,将 MADMAGA 和 TOPSIS 的运行时间进行对比。 $N_T$  的取值从 2 逐渐增加到 10,间隔为 1。每个子任务绑定的候选服务数  $N_S$  设为固定值 10。第二组实验是当任务数  $N_T$  保持不变,候选服务数  $N_S$  从 10 增加到 50 时对两种方法的运行时间进行分析。此时, $N_T$  取值为 5, $N_S$  递增间隔为 5。由于随着  $N_T$  和  $N_S$  的增加,备选组合方案会呈指数级增长,因此,为了提高算法的性能,本实验调整遗传操作中的变异概率  $P_m$ ,即  $P_m$  随着  $N_T$  和  $N_S$  的增加以 0.005 的幅度增加。实验分别运行 10 次并取结果的平均值,计算结果如图 5 和图 6 所示。

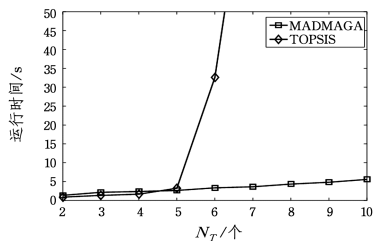


图 5 随着  $N_T$  的增加 MADMAGA 和 TOPSIS 的运行时间对比

Fig. 5 Comparison of running time for MADMAGA and TOPSIS with increasing  $N_T$

由图 5 可知,当任务数  $N_T$  小于 5 时, TOPSIS 方法较为有效,此时服务规模小于  $10^5$ 。由此说明,当问题规模较小时,传统方法更加有效。然而,当任务数从 5 增加到 6 时, TOPSIS 方法的运行时间显著增加。随着任务数的继续增加, TOPSIS 方法已经失效,而本文方法的运行时间呈现出均

匀平稳的变化趋势。根据图 6 所示的时间变化曲线,随着  $N_S$  的增加, MADMAGA 的运行时间增长得非常平缓,相对于图 5,增长趋势较为缓慢。 TOPSIS 方法的运行时间依旧增加得比较明显。

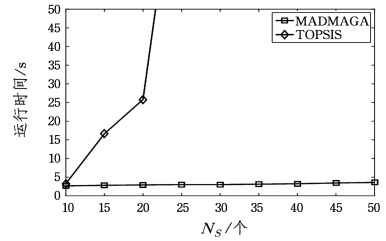


图 6 随着  $N_S$  的增加 MADMAGA 和 TOPSIS 的运行时间对比  
Fig. 6 Comparisons of running time for MADMAGA and TOPSIS with increasing  $N_S$

复合增长率(Compound Growth Rate, CGR)可以反映出的一组数据的增加幅度,因此,本实验计算了两种方法运行时间的 CGR 值。随着  $N_T$  增加, MADMAGA 和 TOPSIS 的 CGR 值分别为 19.30% 和 182.13%。随着  $N_S$  的增加, MADMAGA 和 TOPSIS 的 CGR 值分别为 3.75% 和 146.01%。计算结果说明,当任务数保持不变时,随着候选服务数目的增多, MADMAGA 的运行时间差异不大,说明 MADMAGA 方法能够独立于候选服务数目的变化。当任务数增加时, MADMAGA 的运行时间有一定幅度的增加,但增幅不大。 TOPSIS 方法在两种情况下的时间增幅均很大。

通常,组合服务中的任务数变化幅度不会太大,而  $N_S$  的取值则是千变万化的。实验结果说明,在面对组合服务搜索空间的剧增时, MADMAGA 方法依然能够在多项式时间内快速地找到全局最优解,这充分说明 MADMAGA 方法具有显著的可扩展性。

**结束语** 本文提出了一种结合多属性决策和自适应遗传算法的优化方法 MADMAGA 来解决基于 QoS 的 Web 服务组合优化问题,考虑了 QoS 之间的矛盾性以及多维性。首先根据不同的 QoS 属性创建多属性决策模型,将候选组合服务到正负理想点的距离折中计算;然后采用一种自适应的遗传算法进行寻优求解,最终得到一组最优的组合服务排序结果给用户,用户可以根据自身偏好进行服务选择。实验结果表明,本文方法优于传统的多属性决策方法,能够有效地处理大规模的组合服务搜索空间,且具有良好的可扩展性。

随着 Web 服务数量的增多,对组合服务来说,其每个子任务的候选服务来自于不同的服务提供者。因此,候选服务之间通常是独立的。本文的研究是基于已知固定的 QoS 值,在实际应用中,服务提供者和用户所处的地理位置和软硬件环境也会对服务质量产生一定影响。因此, QoS 值会存在个性化以及动态性的特征。今后将进一步研究环境因素对 Web 服务组合的影响以及基于动态 QoS 的 Web 服务组合。

#### 参考文献

- [1] XU L, LI Y H, CHEN L, et al. A Testing Method for Web Services Focusing on User Requirements [J]. Chinese Journal of Computers, 2014, 37(3): 512-521. (in Chinese)

- 许蕾,李言辉,陈林,等.一种面向用户需求的 Web 服务测试方法[J].计算机学报,2014,37(3):512-521.
- [2] WU Y P,BAO W D,ZHANG W M,et al. Web Service Composition Systems Survey [J]. Computer Science,2011,38(9):1-4. (in Chinese)  
武云鹏,包卫东,张维明,等. Web 服务组合系统研究综述[J]. 计算机科学,2011,38(9):1-4.
- [3] WANG P W,DING Z J,JIANG C J,et al. Constraint-Aware Approach to Web Service Composition [J]. IEEE Transactions on Systems Man & Cybernetics Systems,2017,44(6):770-784.
- [4] ROUACHED M,SALLAY H. A semantic QoS-aware web services composition framework [J]. International Journal of Business Information Systems,2017,17(1):94.
- [5] JATOTH C,GANGADHARAN G R,BUYYYA R. Computational Intelligence based QoS-aware Web Service Composition: A Systematic Literature Review [J]. IEEE Transactions on Services Computing,2017,PP(99):1.
- [6] BENSLIMANE S M,HUCHARD M,et al. QoS-aware optimal and automated semantic web service composition with user's constraints[J]. Service Oriented Computing & Applications, 2017,11(2):1-19.
- [7] LI J,ZHAO Y,LIU M,et al. An adaptive heuristic approach for distributed QoS-based service composition[C]// ISCC'10 Proceedings of the IEEE Symposium on Computer and Communications. 2010:687-694.
- [8] ZHANG K,GAO H H,ZHU Y H,et al. QoS Dynamic Web Services Composition Method Based on Improved Simulated Annealing Algorithm[J]. Journal of Applied Sciences, 2017, 35(5): 570-584. (in Chinese)  
张康,高洪皓,朱永华,等.一种基于改进模拟退火算法的 QoS 动态服务组合方法[J].应用科学学报,2017,35(5):570-584.
- [9] WANG L,ZHAO S S. Research on the Two-stage Heuristic Algorithm Based Web Service Composition Optimization [J]. Electronic Technology,2012(10):19-24. (in Chinese)  
王雷,赵山山.基于两阶段启发式算法的 Web 服务组合优化[J].电子技术,2012(10):19-24.
- [10] LI J,QIAO R,LIU Z Z. Solution of Web Service Composition Scheduling Problem Combining with Game Theory and Multi-objective MILP [J]. Computer Engineering,2016,42(1):11-17. (in Chinese)  
李靖,乔蕊,刘志中.结合对策论与多目标 MILP 的 Web 服务组合调度问题求解[J].计算机工程,2016,42(1):11-17.
- [11] WANG P,CHAO K M,LO C C. On optimal decision for QoS-aware composite service selection[J]. Expert Systems with Applications,2010,9(6):440-449.
- [12] LUO Y S,YANG K,TANG Q,et al. A multi-criteria network-aware service composition algorithm in wireless environments [J]. Computer Communications,2012,35(15):1882-1892.
- [13] MARDUKHI F,NEMATBAKHSH N,ZAMANIFAR K,et al. QoS decomposition for service composition using genetic algorithm[J]. Applied Soft Computing,2013,13(7):3409-3421.
- [14] ANGARITA R,RUKOZ M,CARDINALE Y. Modeling dynamic recovery strategy for composite web services execution [J]. World Wide Web-internet & Web Information Systems,2016, 19(1):1-21.
- [15] GAO H,YAN J,MU Y. Trust-oriented QoS-aware composite service selection based on genetic algorithms[J]. Concurrency & Computation Practice & Experience,2014,26(2):500-515.
- [16] WU Q L,ZHOU T H. Research on Quality of Service-based Dynamic Web Service Composition Method [J]. Computer Application and Software,2016,33(5):20-23. (in Chinese)  
吴青林,周天宏.基于服务质量的动态 Web 服务组合方法研究[J].计算机应用与软件,2016,33(5):20-23.
- [17] ZHANG Y P,JING Z H,ZHANG Y W,et al. Dynamic Web Service Composition Based on Discrete Particle Swarm Optimization[J]. Computer Science,2015,42(6):71-75. (in Chinese)  
张燕平,荆紫慧,张以文,等.基于离散粒子群算法的动态 Web 服务组合[J].计算机科学,2015,42(6):71-75.
- [18] WANG L,SHEN J,LUO J. Facilitating an ant colony algorithm for multi-objective data-intensive service provision[J]. Journal of Computer & System Sciences,2015,81(4):734-746.
- [19] TRAN V X,TSUJI H,MASUDA R. A new QoS ontology and its QoS-based ranking algorithm for Web services[J]. Simulation Modelling Practice & Theory,2009,17(8):1378-1398.
- [20] FANG X R. Study on Filter Algorithm of QoS-Based Fuzzy Multi-Attribute Web Service Composition [J]. Applied Mechanics & Materials,2012,182-183:2131-2135.
- [21] YANG J,LI D F,LAI L B. Composite Service Multi-attribute Selection Method Based on Message Negotiation Under the Web Service Environment [J]. Operations Research and Management Science,2015(3):134-141. (in Chinese)  
杨洁,李登峰,赖礼邦. Web 服务环境下基于信息协商的组合服务多属性选择方法[J].运筹与管理,2015(3):134-141.
- [22] WANG L,SHEN J,LUO J. Facilitating an ant colony algorithm for multi-objective data-intensive service provision [J]. Journal of Computer & System Sciences,2015,81(4):734-746.
- [23] LIAO J,LIU Y,WANG J,et al. Lightweight approach for multi-objective web service composition [J]. IET Software,2016, 10(4):116-124.
- [24] SILVA A S D,MEI Y,MA H,et al. Fragment-based genetic programming for fully automated multi-objective web service composition[C]// The Genetic and Evolutionary Computation Conference. 2017:353-360.
- [25] SUN S X. A decomposition-based approach for service composition with global QoS guarantees[J]. Information Sciences,2012, 199(15):138-153.
- [26] WANG T C,LEE H D. Developing a fuzzy TOPSIS approach based on subjective weights and objective weights [J]. Expert Systems with Applications,2009,36(5):8980-8985.
- [27] LAUMANN S M,THIELE L,DEB K,et al. Combining convergence and diversity in evolutionary multiobjective optimization [J]. Evolutionary Computation,2014,10(3):263-282.
- [28] AL-MASRI E,MAHMOUD Q H. Investigating Web Services on the World Wide Web[C]// International Conference on World Wide Web, WWW 2008, Beijing, China, April. DBLP, 2008:795-804.
- [29] WANG H. Robust Control of the Output Probability Density Functions for Multivariable Stochastic Systems [J]. IEEE Transactions on Automatic Control,1999,44(11):2103-2107.