Applied Soft Computing 39 (2016) 124–139

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Comparative analysis of multi-objective evolutionary algorithms for QoS-aware web service composition

Marcel Cremene [a,*], Mihai Suciu [b], Denis Pallez [c], D. Dumitrescu [b]

[a] Technical University of Cluj-Napoca, Romania
[b] Babes-Bolyai University of Cluj-Napoca, Romania
[c] Université Nice Sophia-Antipolis, France

ARTICLE INFO

Article history:
Received 2 May 2015
Received in revised form
26 September 2015
Accepted 3 November 2015
Available online 14 November 2015

Keywords:
Service composition
Quality of service
Real world services
Multi-objective optimization
Pareto set
Differential evolution

ABSTRACT

Web service composition combines available services to provide new functionality. The various available services have different quality-of-service (QoS) attributes. Building a QoS-optimal web service composition is a multi-criteria NP-hard problem. Most of the existing approaches reduce this problem to a single-criterion problem by aggregating different criteria into a unique global score (scalarization). However, scalarization has some significant drawbacks: the end user is supposed to have a complete a priori knowledge of its preferences/constraints about the desired solutions and there is no guarantee that the aggregated results match it. Moreover, non-convex parts of the Pareto set cannot be reached by optimizing a convex weighted sum. An alternative is to use Pareto-based approaches that enable a more accurate selection of the end-user solution. However, so far, only few solutions based on these approaches have been proposed and there exists no comparative study published to date. This motivated us to perform an analysis of several state-of-the-art multi-objective evolutionary algorithms. Multiple scenarios with different complexities are considered. Performance metrics are used to compare several evolutionary algorithms. Results indicate that GDE3 algorithm yields the best performances on this problem, also with the lowest time complexity.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Web services are very popular and widespread in enterprise distributed environments, being a central aspect in Software Engineering. The most significant advantages of the service-oriented approach are: reusability, interoperability across different platforms, location independence, security support, dynamic search/connectivity, and compatibility within the Cloud Computing paradigm.

New services can be created by combining existing ones. The process is called *service composition* or *service orchestration* [1]. A *composite service* is a service obtained by composition/orchestration and the way of composing services – the service architecture – is generally described by a *workflow*.

Each service is characterized by a certain functionality (logic/functional aspect) and a set of non-functional aspects called *quality of service* (QoS) attributes. For instance, the function of a service may be to provide a map for a given address and a QoS attribute may be the associated response time. QoS-aware service composition is discussed in the next section.

1.1. QoS-aware web service composition

Different services offering the same functionality may have different QoS attributes [2]. QoS-aware service composition is an NP-hard [3] multi-objective optimization problem (MOP) well known in Service Oriented Computing (SOC) [4,5]. Each QoS attribute may be associated to an objective/criterion. Some objectives cannot be simultaneously optimized: a service may have a lower cost but a higher response time whereas another one may be more expensive but faster.

The global QoS of a composite service depends on the QoS of each of the services used in the composition. Therefore, the problem consists in selecting, among all the available services, those ensuring the optimal composite service QoS. User preferences should be also taken into account: some users may prefer cheaper services while other users may prefer high quality services.

Numerous approaches have been proposed for solving this problem (e.g., [6–15]). Exhaustive search algorithms try to find the optimal solution whereas metaheuristic algorithms are only looking for a 'good' solution that may be obtained in a reasonable time frame. Different optimization algorithms types have been used: integer programming, linear programming, dynamic programming, tabu search, local search, evolutionary algorithms, and hybrid algorithms. An overview may be found in [13].

1.2. Existing approaches

Existing QoS-aware service composition approaches may be categorized in three classes:

- *Scalarization-based* approaches. A global evaluation function aggregates different QoS attributes into a unique, global score. This aggregation procedure is also called *scalarization*. Despite the fact that QoS-aware service composition is a multi-objective problem, almost all existing approaches use scalarization in order to

* Corresponding author. Tel.: +40 742676404.
E-mail address: cremene.marcel@gmail.com (M. Cremene).

http://dx.doi.org/10.1016/j.asoc.2015.11.012
1568-4946/© 2015 Elsevier B.V. All rights reserved.

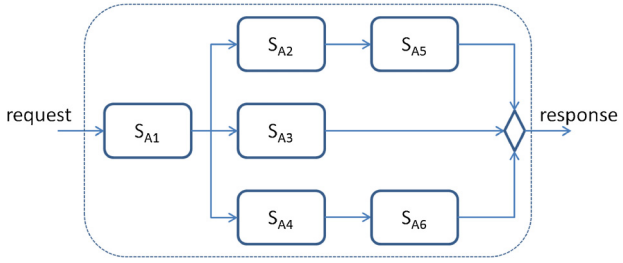

Fig. 1. A workflow example describing a composite service composed of 6 services.

reduce this problem to a single-objective one. This is, however, a very particular approach that has some important drawbacks discussed in detail in Section 3.1.3.

- *Pareto-based* approaches. Multi-objective (MO) algorithms are used to identify multiple solutions. Very few existing Pareto-based proposals have been found in the literature for this specific problem. Pareto-based approaches offer a higher accuracy in selecting a solution in accordance with user preferences.
- *Hybrid* methods. Some recent proposals combining both Pareto and scalarization-based techniques exist. Since they are also based on scalarization they present the same drawbacks already discussed.

1.3. Paper objective

To the best of our knowledge, there is no Pareto-based approaches comparative analysis available to date targeting the QoS-aware service optimization problem. Hence, this paper analyzes several evolutionary multi-objective algorithms (EMOAs). Some of the most known state-of-the-art EMOAs are evaluated using a dataset collected from real web services. The comparison is based on standard performance metrics and statistical analysis.

1.4. Outline

The next section defines the QoS-aware service optimization problem. Section 3 presents the most relevant existing approaches to address the QoS-aware service composition problem. Section 4 presents a comparative analysis of several state-of-the-art EMOAs. Finally, the last section summarizes the experimental results interpretations and concludes the paper.

2. Problem statement

A composite service may be described by a workflow [16]. A workflow contains activities and composition elements: sequence, parallel, switch (condition) and loop structures. An example including 6 activities: S_{A1} to S_{A6} is depicted in Fig. 1. Executing an activity means invoking a service.

It is necessary to make a distinction between an *abstract service* (denoted S_{Ai}) and a *concrete service* (denoted S_{Ci}). An abstract service is a model describing a functionality – a *service class/type*. Each workflow activity S_{Ai} is associated to an abstract service. A concrete service S_{Ci} is a real web service that may be invoked – an *implementation/instance* of the abstract service class.

A hypothesis is that several alternative concrete services exist for each abstract service. For instance, different booking services may be used for a same flight. Each concrete service is characterized by different QoS attributes. Therefore, it is important to know which concrete service is selected to implement an abstract service. Given l abstract services and k concrete services for each abstract service, there are k^l combinations. This is a combinatorial multi-objective optimization problem. The search space is discrete: a solution may be encoded as a vector of l integer values, each value i indicates a concrete service S_{Ci}, $1 \le i \le k$. Finding the optimal service combination is a non-deterministic polynomial-time hard (NP-hard) problem (as identified in [3–5]). This means that an exhaustive search algorithm is impractical, excepting very simple cases (few abstract and concrete services).

The problem is not only to find the solutions with the best QoS but also to identify the subset of solutions matching the user preferences.

2.1. QoS attributes

Some of the most used QoS attributes for web services are described bellow (e.g., [7,17,18,13,19]):

- *Response time (t)* represents the round-trip time between sending a request to the service and receiving a response;
- *Availability (a)* is defined as being the number of successful invocations over the total number of invocations. It measures the probability that a service is available;
- *Throughput (d)* represents the total number of invocations served by a service for a given period of time;
- *Reliability (r)* represents the services capacity at ensuring reliable message delivery;
- *Cost/price (c)* is sometimes considered as a QoS attribute even if it is not a quality indicator in a strict sense. It represents the cost/price of accessing a service (i.e. the amount of money to pay) for each service request;
- *User rating (u)* is a mean score given by the users about their subjective service usage satisfaction.
- *Security (s)* is a measure of the security level offered by a service.

2.2. QoS composition rules

The QoS attributes of a composite service are obtained by composing the QoS attributes of each of the services used in the composition, as depicted in [17]. Examples of composition rules/operators are proposed in [7,17,5,20,13]. Several of the most used composition rules are depicted in Table 1.

The composite service workflow includes several architectural patterns. Four pattern types are considered: sequence composition, parallel composition, condition, and loop. A specific QoS composition rule is defined for each architectural pattern (see Table 1). For instance, the response time of a parallel composition made of two or more services is given by the highest response time. If several services are invoked sequentially, the global response time is the sum of the time response of each service.

In the case of conditional/switch pattern (XOR), each service is invoked with a probability p_i and the response time is an average based on these probabilities.

A loop structure will multiply the response time by the number of loop cycles. Similar rules are defined for other QoS attributes (see Table 1).

2.3. Multi-objective optimization problems – general concepts

QoS-aware service composition is a multi-objective optimization problem (MOP) since several QoS attributes have to be optimized simultaneously. In this section we formally define a MOP and we introduce its specific terminology.

Let $S \subseteq \mathbb{R}^n$ denote the n-dimensional search space and $F \subseteq S$ the feasible space. When the optimization problem has no constraint, feasible space and search space are identical ($F = S$).

Let $x = (x_1, \ldots, x_n) \in S$, refer to a *decision vector*. Let us consider m objectives defined by a set of real valued functions $f_i : \mathbb{R}^n \to \mathbb{R}$ with $i = 1 \ldots m$.

Function $f(x) : \mathbb{R}^n \to \mathbb{R}^m$, defined as $f(x) = (f_1(x), \ldots, f_m(x))$, is called the *objective vector*.

When solving a MOP, the goal is to find decision vectors $x^* \in F$ which optimize function f, satisfying a set of defined constraints g (a set of k inequality-based functions), and h (a set of q equality-based functions).

دریافت فوری ← متن کامل مقاله

ISIArticles
مرجع مقالات تخصصی ایران

✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
✓ امکان دانلود نسخه ترجمه شده مقالات
✓ پذیرش سفارش ترجمه تخصصی
✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
✓ دانلود فوری مقاله پس از پرداخت آنلاین
✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات