An Adapted Ant-Inspired Algorithm for Enhancing Web Service Composition

Fadi Dahan, King Saud University, Riyadh, Saudi Arabia
Khalil El Hindi, King Saud University, Riyadh, Saudi Arabia
Ahmed Ghoneim, King Saud University, Riyadh, Saudi Arabia

ABSTRACT

Web Service Composition (WSC) provides a flexible framework for integrating independent web services to satisfy complex user requirements. WSC aims to choose the best web service from a set of candidates. The candidates have the same functionality and different non-functional criteria such as Quality of Service (QoS). In this work, the authors propose an ant-inspired algorithm for such problem. They named it Flying Ant Colony Optimization (FACO). Flying ants inject pheromone not only on the nodes on their paths but also on neighboring nodes increasing their chances of being explored in future iterations. The amount of pheromone deposited on these neighboring nodes is inversely proportional to the distance between them and the nodes on the path. The authors believe that by depositing pheromone on neighboring nodes, FACO may consider a more diverse population of solutions, which may avoid stagnation. The empirical experiments show that FACO outperform Ant Colony Optimization (ACO) for the WSC problem, in terms of the quality of solutions but it requires slightly more execution time.

KEYWORDS

Ant Colony Optimization (ACO), Flying Ant Colony Optimization (FACO), Service-Oriented Computing (SOC), Web Service (WS), Web Service Composition (WSC)

1. INTRODUCTION

Web services (WS’s) are used in Service-Oriented Computing (SOC) to support low-cost, rapid development, code reuse, platform independent and easier to maintain applications. WS’s are self-contained and self-described applications. They interact with other applications over the web (Gustavo, Casati, Kuno, & Machiraju, 2004; Papazoglou, 2008) to perform sophisticated tasks. WS providers create services, and then store their description into repository to make them available for potential users.

However, it may be difficult or impossible to find an individual (or atomic) WS that performs all tasks needed by a business process. Therefore, WS’s may need to be composed to fulfill the requirements of a business process (Rodriguez-Mier, Mucientes, Vidal, & Lama, 2012). This process is called Web Service Composition (WSC).
From an enterprise perspective, WSC helps to develop a new business application at low cost and risk (Sheng et al., 2014). Business process can be described using two ways (Andrews et al., 2003): executable business processes (EBP) where the business process is defined in detail, and abstract business processes (ABP) where the message exchanged by the parties are specified.

In this work, we use ABP, which is modeled as n tasks/levels, T=[t_1,t_2,…,t_n]. These tasks are formed as a workflow. For each task, a set of different WS’s, called Candidate Web Services (CWS), are available from different providers. These WS’s perform a similar functionality, and meet different non-functional criteria. The aim is to select a good combination of the WS’s that meets the user non-functional requirements such as cost and availability.

Tasks and CWS can be presented as a directed acyclic graph (see Figure 1), where the first task is the start state and the last task is the end state. Assuming that there are m CWS’s for each task, there are m^n available paths, each represents a potential solution for the problem.

The problem is challenging because in addition to the WS functional requirements we need to consider several non-functional requirements defining the QoS properties. These are cost, response time, availability and reliability.

Clients get the services from the service registry over the World Wide Web based on the WS description (Menasce, 2002). Many providers publish their WS’s over the registry; these services may be identical in terms of functionality. In such cases, we may want to select the WS’s that meet our QoS criteria. QoS can be divided into two groups deterministic and nondeterministic (Liu, Ngu, & Zeng, 2004). Deterministic attributes refer to attributes with known values before invocation, such as cost or security protocols. Nondeterministic attributes refer to the uncertain values that cannot be determined in advanced such as response time, availability and reliability. Additionally, these criteria affect the WS performance in different ways because of their objective value. For some criteria such as cost and response time, the desired value is the minimum, for others such as availability and reliability, the desired value is the maximum.

Section 2 defines the QoS-aware WSC problem. Section 3, reviews related works that use ACO for the WSC problem. In Section 4, we present FACO algorithm. Section 5 presents the experimental results. The last section is conclusion section.

![Figure 1. Modeling for CWS that attached to each task](image-url)
Ontology-Driven Interactive Visualization of Film Production Complexity Using a Visual Language
www.igi-global.com/article/ontology-driven-interactive-visualization-of-film-production-complexity-using-a-visual-language/153669?camid=4v1a

The Berlin SPARQL Benchmark
www.igi-global.com/chapter/berlin-sparql-benchmark/55042?camid=4v1a