

A Hybrid Firefly-inspired Approach for Optimal Semantic Web Service Composition

Cristina Bianca Pop, Viorica Rozina Chifu, Ioan Salomie,
Ramona Bianca Baico, Mihaela Dinsoreanu, Georgiana Copil

Department of Computer Science
Technical University of Cluj-Napoca
26-28 Baritiu str., Cluj-Napoca, Romania
Cristina.Pop@cs.utcluj.ro

Abstract. Biological systems exhibit self-optimization and self-organization capabilities in a de-centralized way without the existence of a central coordinator. These capabilities make them a good source of inspiration for designing meta-heuristics for solving optimization problems in computer science. Inspired from biology, in this paper we propose a hybrid firefly method for selecting the optimal solution in semantic Web service composition. In our approach, the search space of the selection method is represented by an Enhanced Planning Graph structure which encodes all the Web service composition solutions for a given user request. As selection criteria we have considered the *QoS* attributes of the services involved in the composition as well as the semantic similarity between them. For the evaluation of the proposed selection method we have implemented an experimental prototype and carried out experiments on a scenario from the trip planning domain.

Keywords: semantic Web service composition, firefly-based search, genetic operators, *QoS*, semantic similarity.

1 Introduction

The selection of the optimal solution in semantic Web service composition can be seen as an optimization problem which requires specific selection techniques that provide the desired results in an efficient way. Recent research studies demonstrated that principles inspired by the biological systems have lead to the design of efficient techniques that can be used to solve optimization problems. These biologically-inspired techniques are advantageous since they are capable of converging towards the optimal or a near-optimal solution in a short time without processing the entire search space. Such meta-heuristics include Ant Colony Optimization [1], Particle Swarm Optimization [2], or Artificial Immune Systems [3]. These meta-heuristics have been successfully applied to the problem of selecting the optimal solution in Web service composition [4, 5, 6].

In this paper we present a hybrid method for selecting the optimal composition solution that combines an extended version [7] of the Firefly Search algorithm [9]

with genetic operators. The search space for the hybrid method is represented by an Enhanced Planning Graph (EPG) which encodes all the service composition solutions for a given user request. In our approach, a user request is described in terms of functional and non-functional requirements. To identify the optimal solution encoded in the EPG, we define a fitness function which uses the *QoS* attributes and the semantic quality of the services involved in composition as selection criteria. The proposed selection method was tested on a scenario from the trip planning domain.

The paper is structured as follows. Section 2 presents the formal models for representing semantic Web service composition. Section 3 details the hybrid firefly method for selecting the optimal solution in Web service composition. Section 4 presents the experimental framework. We end our paper with conclusions.

2 Semantic Web Service Composition Model

In our approach the Web service composition is modeled by using an *Enhanced Planning Graph (EPG)* structure [8]. This graph actually represents the search space of the hybrid firefly selection method. The *EPG* is obtained by mapping the classical AI planning graph to the semantic Web service composition domain and by adding new domain related structures to ease the service composition representation and service selection. The construction of the *EPG* is an iterative process which is applied at the semantic level by considering the ontology concepts that annotate the services functionality and their input/output parameters. In each iteration, a new layer consisting of a tuple (A_i, L_i) is added to the graph where: (1) A_i contains clusters of services whose inputs are provided by the services from the previous layers and (2) L_i contains clusters of service parameters. A cluster of services groups services which provide similar functionality, while a cluster of service parameters groups similar input and output parameters. The first graph layer is represented by the tuple (A_0, L_0) , where A_0 is an empty set of service clusters and L_0 contains the user-provided input parameters. The construction of the *EPG* ends either when the user requested outputs are contained in the current set of literals or when the graph reaches a fixed point. Reaching a fixed point means that the sets of actions and literals are the same for the last two consecutive generated layers. A composition solution encoded in the *EPG* consists of a set of services, one from each cluster from each *EPG* layer.

3 The Hybrid Firefly Selection Method

The hybrid method for selecting the optimal solution in semantic Web service composition combines a firefly algorithm [7] with principles from evolutionary computing. We have proposed such a hybrid method to maintain a good balance between exploration and exploitation thus eliminating the problem of local optimum stagnation.

3.1 Problem Formalization

The firefly meta-heuristic relies on a set of artificial fireflies which communicate with each other to solve optimization problems. The behavior of artificial fireflies is modeled according to the behavior of fireflies in nature, which search for a mating partner by emitting a flashing light. In this section we present how we mapped the concepts of the firefly meta-heuristic to the problem of Web service composition.

Just as the real fireflies search for a mating partner by means of flashing lights, we have a number of artificial fireflies which search for the optimal service composition solution. Thus, we map the attraction behavior of fireflies to the problem of selecting the optimal service composition as follows: (i) a firefly becomes an artificial firefly, (ii) the position of a firefly becomes a service composition solution, (iii) the brightness of a firefly becomes the quality of a composition solution evaluated with a multi-criteria fitness function, (iv) the attractiveness between two fireflies becomes the similarity between two composition solutions, (v) the movement of a firefly is mapped to a modification of the firefly's current composition solution, (vi) the environment in which fireflies fly is mapped to the *EPG*.

We formally define an artificial firefly as follows:

$$firefly = (sol, score) \quad (1)$$

where *sol* is a service composition solution and *score* is the quality of *sol*.

A service composition solution is defined as:

$$sol = \{solElem_1, \dots, solElem_n\} \quad (2)$$

where: (i) *solElem_i* is a solution element composed of a set of services, one service from each cluster of layer *i*; and (ii) *n* is the total number of layers in the *EPG*.

To evaluate a composition solution, we define a *fitness function QF* which considers the *QoS* attributes of the associated services as well as the semantic quality of the connections between these services:

$$QF(sol) = \frac{w_{QoS} * QoS(sol) + w_{Sem} * Sem(sol)}{(w_{QoS} + w_{Sem}) * |sol|} \quad (3)$$

where: (i) *QoS(sol)* [8] is the *QoS* score of the composition solution *sol*; (ii) *Sem(sol)* [8] is the semantic quality score of the solution *sol*; (iii) *w_{QoS}* and *w_{Sem}* are the weights corresponding to user preference related to the relevance of *QoS* and semantic quality.

3.2 The Hybrid Selection Algorithm

A prerequisite of the hybrid selection method is to establish the number of fireflies that will be used in the search process so as to obtain the optimal solution in a short time interval and without processing the entire search space. We have defined the number of fireflies based on the total number of solutions encoded in the *EPG*:

$$noF = Round(\sqrt[n]{noSol}) \quad (4)$$

where: (i) *noSol* is the number of possible composition solutions encoded in the *EPG*; and (ii) *n* is a positive integer determined experimentally.

The inputs of the selection algorithm (*Algorithm_1*) are: (i) the *EPG* resulted from the Web service composition process, (ii) the weights w_{Sem} and w_{QoS} which state the relevance of a solution's semantic quality compared to its *QoS* quality, and (iii) a number *noF* (formula 4) of artificial fireflies used to search for the best composition. The algorithm returns the optimal or a near-optimal composition solution.

Algorithm_1: Hybrid_Firefly_Web_Service_Selection

Inputs: *EPG*; *noF*; w_{QoS} ; w_{Sem} ; *noF*

Output: $fSol_{best}$

begin

$FSOL = \{ \}$

for $i = 1$ **to** *noF* **do** $FSOL = FSOL \cup \text{Generate_Random_Solution}(EPG)$

repeat

for $i = 1$ **to** m **do**

for $j = 1$ **to** m **do**

if ($QF(FSOL[i]) < QF(FSOL[j])$) **then**

$r = \text{Compute_Distance}(FSOL[i], FSOL[j])$

$FSOL[i] = \text{Crossover}(FSOL[i], FSOL[j], r)$

$u = \text{Generate_Random_Vector}(|FSOL[i]|)$

$FSOL[i] = \text{Mutation}(FSOL[i], u)$

end if

end for

end for

$fSol_{best} = \text{Get_Best_Solution}(FSOL)$

$SOL_{best} = SOL_{best} \cup fSol_{best}$

$u = \text{Generate_Random_Vector}(|fSol_{best}|)$

$FSOL = \text{Modify_Best_Firefly}(FSOL, u)$

until (**Stopping_Condition**())

return $\text{Get_Best_Solution}(SOL_{best})$

end

In the first step of the selection algorithm each firefly is associated with a randomly generated composition solution (*Generate_Random_Solution*). These initial solutions are further improved in an iterative process which stops when the best solution has been the same over the last *noIt* iterations (*Stopping_Condition*).

In each iteration, if the score of the solution associated to a firefly is better than the score of the solution associated to another firefly it means that the latter firefly will be attracted towards the first one and thus it will have its solution improved. The steps for improving the solution associated to the less bright firefly are the following:

- 1) The distance r between the two composition solutions is computed (*Compute_Distance*) as the difference of their scores.
- 2) A crossover operator is applied between the two composition solutions in a number of points depending on the value of the distance r (*Crossover*). As a result of the crossover operation, two new solutions will be obtained and the one having the highest score according to the QF function will be kept.

- 3) A mutation operation (*Mutation*) is performed on the best solution obtained within crossover to introduce diversity. In the mutation process, a mutation vector is randomly generated (*Generate_Random_Vector*) to specify the points where services will be replaced with other services from the same clusters.

After all solutions have been processed, the best one is determined (*Get_Best_Solution*), added to the set of the best composition solutions and then mutated (*Modify_Best_Firefly*) according to a randomly generated mutation vector (*Generate_Random_Vector*). This last mutation is performed to enlarge the search space and to avoid the stagnation in a local optimal solution.

4 Experimental Framework

To validate our selection approach we have implemented an experimental framework whose architecture is presented in Figure 1.

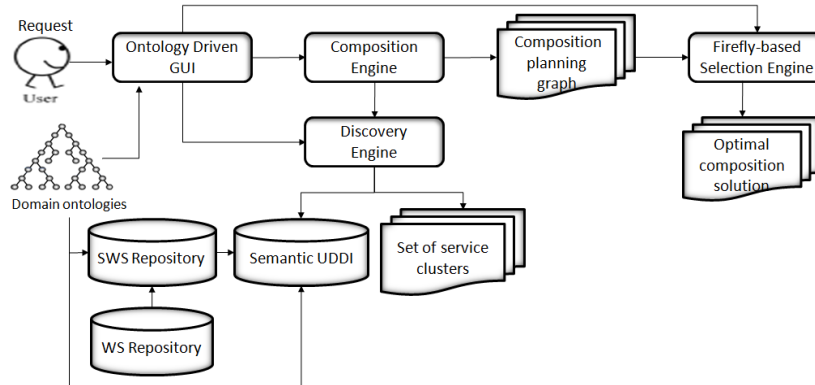


Fig. 1. The architecture of the experimental framework.

The ontology driven graphical user interface guides the users in the processes of searching and composing Web services by providing a controlled language that uses the ontology concepts. The SWS Repository is a repository of semantically annotated services based on the on the Domain Ontology. The Semantic UDDI extends the classical UDDI structure by (i) storing semantic Web service descriptions and (ii) providing means to semantically inquire the UDDI repository. The Discovery Engine receives a service request from a user or from the Composition Engine. In the case of a user request, the Discovery Engine provides an ordered set of atomic Web services which match the request. In the case of a service request from the Composition Engine, the Discovery Engine provides a set of atomic Web services organized in clusters. To satisfy the service requests, the Discovery Engine inquires the Semantic UDDI. The Composition Engine is the component responsible for the construction of the EPG. The Composition Engine interacts with the Discovery Engine which provides the appropriate service clusters. The EPG is then used by the Firefly-based selection engine which provides the optimal or a near optimal composition solution according to the user preferences. The framework has been tested on a scenario from the trip planning domain for which the EPG is organized on 3 layers consisting of 51

services grouped in 11 clusters. This EPG encodes 13996800 composition solutions. To evaluate the hybrid firefly selection algorithm we have performed 100 runs of the algorithm and considered the following criteria: the average number of processed solutions, the average simulation time, and the standard deviation. By analyzing the experimental results we conclude that the hybrid selection algorithm returns the optimal or a near-optimal solution (the average standard deviation is 0.089) on average in 8 seconds by processing around 0.015 % of the search space.

5 Conclusions

This paper proposed a firefly-inspired method for selecting the optimal or a near optimal solution in semantic Web service composition. The selection method has been applied on Enhanced Planning Graph which encodes the set of composition solutions for a given user request. By combining the firefly-based selection approach with genetic operators we ensure a good balance between exploration and exploitation thus avoiding the problem of stagnation in a local optimum. To demonstrate the feasibility of our approach, we have implemented an experimental which has been tested on a scenario from the trip planning domain.

References

1. Dorigo, M., Birattari, M., Stützle, T.: Ant Colony Optimization - Artificial Ants as a Computational Intelligence Technique, *IEEE Computational Intelligence Magazine*, 1(4), 28-39 (2006).
2. Kennedy, J., Eberhart, R.C.: Particle swarm optimization, *The Proceedings of the IEEE International Conference on Neural Networks*, Piscataway, NJ. 1942-1948 (1995).
3. Castro, L., von Zuben, F.: Learning and Optimization using the Clonal Selection Principle. *Proc. Of the IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems*, vol. 6, n. 3, pp. 239-251 (2002)
4. Gao, Y., Bin, N. J., Z., Lei, Y., Qiang, G., Yu, D.: Immune Algorithm for Selecting Optimum Services in Web Service Composition, *The Wuhan University Journal of Natural Sciences*, 11(1), 221-225 (2006).
5. Li, W., Yan-xiang, H.: A Web Service Composition Algorithm Based on Global QoS Optimizing with MOCACO, *Algorithms and Architectures for Parallel Processing, LNCS*, vol. 6082, 218-224 (2010).
6. Ming, C., Zhen-wu, W.: An Approach for Web Services Composition Based on QoS and Discrete Particle Swarm Optimization, *Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing*, 37-41 (2007).
7. Lukasik S., Zak S.: Firefly Algorithm for Continuous Constrained Optimization Tasks, *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems, LNCS vol. 5796*, 97-106 (2009).
8. Pop, C. B., Chifu, V. R., Salomie, I., Dinsoreanu, M.: Immune-Inspired Method for Selecting the Optimal Solution in Web Service Composition, *LNCS*, vol. 6162, 1-17 (2010).
9. Yang, X.S.: *Nature-Inspired Metaheuristic Algorithms*, Luniver Press (2008).