We are IntechOpen, the world's leading publisher of Open Access books
Built by scientists, for scientists

6,900
Open access books available

186,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Bio-Inspired Hybrid Algorithm for Web Services Clustering

Maricela Bravo, Román A. Mora-Gutiérrez
and Luis F. Hoyos-Reyes

Abstract

Web services clustering is the task of extracting and selecting the features from a collection of Web services and forming groups of closely related services. The implementation of novel and efficient algorithms for Web services clustering is relevant for the organization of service repositories on the Web. Counting with well-organized collections of Web services promotes the efficiency of Web service discovery, search, selection, substitution, and invocation. In recent years, methods inspired by nature using biological analogies have been adapted for clustering problems, among which genetic algorithms, evolutionary strategies, and algorithms that imitate the behavior of some animal species have been implemented. Computation inspired by nature aims at imitating the steps that nature has developed and adapting them to find a solution of a given problem. In this chapter, we investigate how biologically inspired clustering methods can be applied to clustering Web services and present a hybrid approach for Web services clustering using the Artificial Bee Colony (ABC) algorithm, K-means, and Consensus. This hybrid algorithm was implemented, and a series of experiments were conducted using three collections of Web services. Results of the experiments show that the solution approach is adequate and efficient to carry out the clustering of very large collections of Web services.

Keywords: artificial bee colony, K-means, Consensus, hybrid algorithms, Web services clustering, semantic similarity measures

1. Introduction

Web services clustering is the task of selecting and extracting the features of a collection of Web services, discovering the similarities between them to form groups or classes considering those features. The implementation of novel and efficient algorithms for the automatic clustering of Web services is relevant for the organization of large collections of services in private or public network such as the Internet. Having a directory of Web services organized in groups according to one or more characteristics represents an advantage during the search, selection, invocation, substitution, and composition of Web services.

Methods inspired by nature using biological analogies have been adapted for clustering problems, among which genetic algorithms, evolutionary strategies, and algorithms that imitate the behavior of some animal species have been implemented. Living beings such as animals and plants and even the climate exhibit
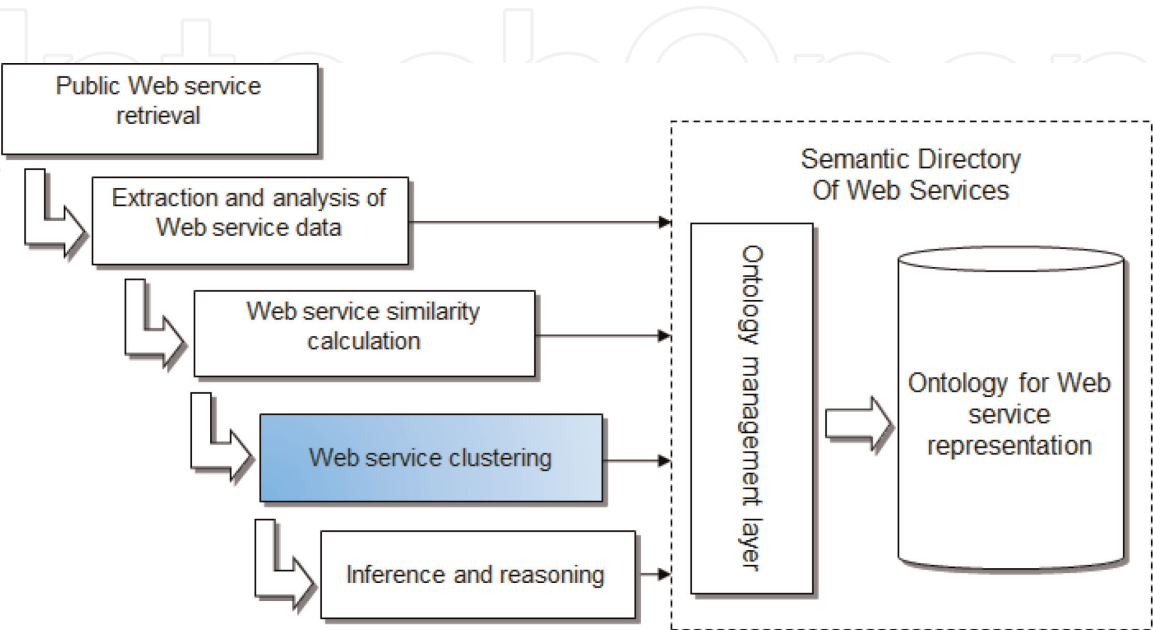
extraordinary, complex, and fascinating natural phenomena. Of particular interest is the intelligent behavior of some animal species to find a solution to solve a problem and maintain the perfect balance of the environment surrounding.

This is the main idea of computation inspired by nature, that is, to imitate the steps that nature has developed and adapt them to find a solution of a problem, thus converting it into a bio-inspired algorithm. This is the main reason for the implementation of an algorithm that exploits the collective intelligence of a Bee Colony as an alternative for clustering. This chapter presents an innovative approach for Web services clustering using a hybrid algorithm based on the Artificial Bee Colony, K-means, and Consensus.

The work described in this chapter is part of a research project whose objective is to design and implement a semantic directory of Web services using efficient, fully automated methods to allow the organization, composition, and classification of Web services with a semantic approach.

Figure 1 shows the general architecture of the semantic directory and a methodology to construct and manage the directory. The methodology consists of the following phases:

1. *Public Web service retrieval* aims at searching over the Internet to find and copy Web service descriptions (files formatted in WSDL description language) in a local file directory. Web service retrieval is executed through crawlers designed specifically to parse and identify links to files in WSDL.

2. *Extraction and analysis of Web services* consists of parsing every Web service description file and the extraction of specific data, such as: method names, input and output parameters, and port types to facilitate the automatic invocation. Extracted data is stored in an ontology model. For this phase, we use a tool which transforms WSDL files into an ontological representation.

3. *Web service similarity calculation* is an important phase of the methodology, because classification and clustering of Web services requires the calculation of distances between services. In order to calculate similarities, different measures can be implemented and combined to obtain better results.

Figure 1.
Semantic directory of Web services.

4. *Web service classification and clustering* consists of selecting and extracting the characteristics of a collection of Web services and discovering the similarities between them to form groups or classes considering those characteristics. Therefore, this phase depends on the results of the previous phases. In particular, in this chapter a hybrid clustering algorithm based on the Artificial Bee Colony algorithm, the K-means, and Consensus is described.

5. *Inference and reasoning* represents the supporting mechanisms to exploit and utilize the enriched Web service ontologies.

2. Related work

Web service classification and clustering is a topic addressed form different perspectives such as statistical, stochastic, and novel approaches based on bio-inspired algorithms. Additionally, semantic approaches to describe, discover, and invoke Web services have been studied to propose novel clustering algorithms. In this section a revision of related work is presented considering two trends: reported work that address clustering and classification of Web services and clustering approaches based on bio-inspired algorithms. **Table 1** presents the main characteristics of related work.

In 2009, Liang et al. [1] proposed a method for Web service categorization considering keywords and semantic relations between elements of the description. Their proposed methodology involves preprocessing WSDL documents, rough clustering by labeling Web services with class tag, and fine clustering.

In 2009, Platzer et al. [2] described a scalable approach for clustering very large service repositories. They use a statistical clustering algorithm enhancing a vector space to support the search of services related to a given query.

In 2012, Pop et al. [3] presented two approaches for service clustering, one inspired by the behavior of the birds and other inspired by the behavior of ants. They implemented methods to evaluate the semantic similarity between services.

In 2013, Du et al. [4] presented an approach for clustering Web services based on functional similarity and refinement of clusters using a concept position vector.

In 2014, Wu et al. [5] presented an approach which consists of three modules: data preprocessing, Web service tag recommendation, and Web services clustering. The first module consists of building a content vector formed with nouns, verbs, or adjectives. Authors use different features and different approaches for similarity computation. For content use the normalized Google distance, for data types and messages they use a basic match similarity, and for tag similarity they apply the Jaccard coefficient.

In 2014, Prakash and Singh [6] compared the performance of evolutionary algorithms: Genetic Algorithm, Differential Evolution, Particle Swarm Optimization, and Artificial Bee Colony for clustering three real and one synthetic data sets.

In 2017, Sahoo [7] presented a two-step ABC algorithm for data clustering problems. Authors addressed the three problems of the ABC algorithm such as initial positions of food sources, solution search equation, and abandoned food location.

In 2018, Kotekar and Kamath [8] described a Web services clustering approach based on Cat Swarm Optimization (CSO), which emulates the social behavior of cats in nature.

Automated Web services clustering is useful to facilitate service search, service discovery, service composition, and service substitution. Of particular interest is the representation of Web services through ontologies because the purpose of this work is the automatic organization of any collection (public or private) of Web service in ontologies and their semantic enrichment by classification and clustering.

Work	Input data	Clustering approach	Use of ontologies	Similarity approach	Number of Web services used	Service repository	Benefits	Limitations
Liang et al. [1]	WSDL documents	Incremental K-means algorithm Bisecting K-means	No	Tree-based structure matching	352	Xmethods.com Bindingpoint.com Webservice list.com Xignite.com	This approach clusters Web service documents	The similarity approach is not semantic, and the clustering method is not bio-inspired
Platzer et al. [2]	WSDL documents	Statistical clustering analysis	No	Euclidean distance	275	Xmethods.com	This approach clusters Web service documents	The clustering method is not based on novel bio-inspired algorithms. Similarity measure is not semantic
Pop et al. [3]	WSDL documents extracted from OWL-TC4	Particle swarm and ant-based service clustering	Yes	Semantic similarity by evaluating the Degree of Match (DoM)	894	SAWSDL-TC collection	The solution approach is very similar to the approach described in this chapter. The main difference is on the algorithms utilized	The semantic similarity does not use a lexical database to improve similarity measures
Du et al. [4]	WSDL documents extracted from OWL-TC4	Bottom-up hierarchical clustering	No	Semantic similarity based on WordNet	1075	OWL-TC4 collection	This work is closely related with the approach presented in this chapter	The clustering method is not based on novel bio-inspired algorithms
Wu et al. [5]	WSDL documents	K-means	No	The similarity integrates all the feature measures using a weighed sum	15,968	Seekda	This approach clusters Web service documents	The clustering method is not based on bio-inspired algorithms. Similarity measure is not semantic
Prakash and	Data is based on three real and	Genetic Algorithm, Differential Evolution, Particle Swarm	No	Not specified	Not for Web services	No	The clustering approach is based on novel bio-inspired algorithms.	The clustering method is not applied to Web services

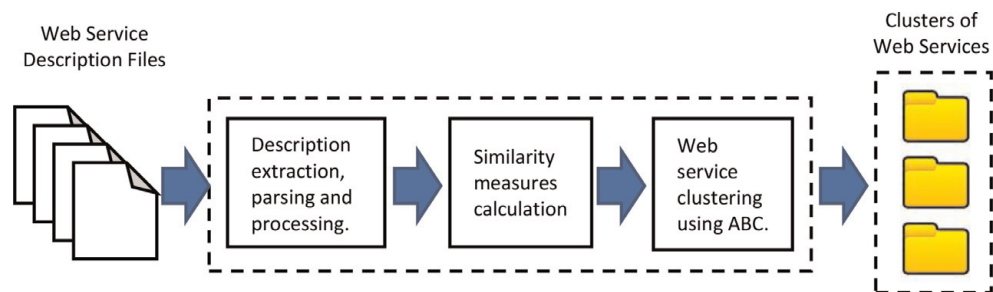| Work	Input data	Clustering approach	Use of ontologies	Similarity approach	Number of Web services used	Service repository	Benefits	Limitations
Singh [6]	one synthetic data sets	Optimization, and Artificial Bee Colony						
Sahoo [7]	Data is downloaded from UCI repository	Two-step ABC algorithm	No	Euclidean distance	Not for Web services	Data sets are downloaded from the UCI repository	The clustering approach is based on novel bio-inspired algorithms	The clustering method is not applied to Web services. Similarity measure is not semantic
Kotekar and Kamath [8]	WSDL documents extracted from OWL-TC4	Cat Swarm Optimization (CSO)	No	Euclidean distance and TF-IDF	1083	OWL-TC4 collection	The clustering approach is based on novel bio-inspired algorithms	Similarity measure is not semantic

Table 1.
Comparison of related work.

3. Clustering process

Clustering of Web services consist of partitioning the set of Web services in the collection into an appropriate number of clusters based on a similarity measure. Therefore, services in the same cluster are more similar than the services in the different clusters [6].

In this section the clustering approach implemented is described. This process has as input a collection of Web services formatted according to Web Service Description Language (WSDL). This collection of Web services is processed utilizing specific parsers to extract the most important data of the service description, which are the method names and input and output parameters. The detailed process is described in the following subsections (**Figure 2**).
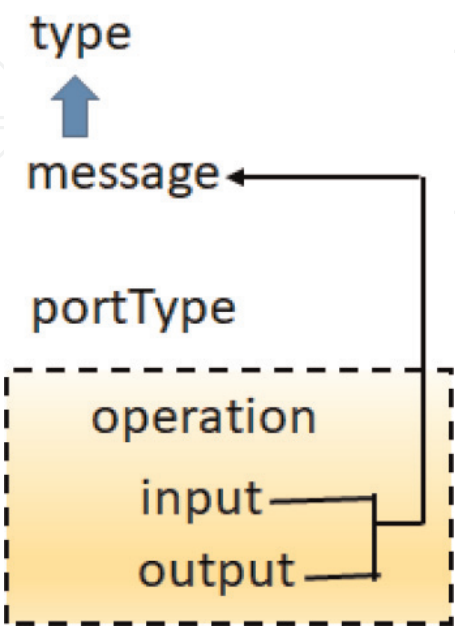
3.1 Extracting and parsing

Every Web service description includes the definition of the programming interfaces to be invoked remotely. **Figure 3** shows the abstract service interface

Figure 2.
Clustering process of Web services.

Figure 3.
Web service interface definition.

Semantic similarity measure	Description	Approach	Advantage
Lesk [10]; Banerjee and Pedersen [11]	Address word sense disambiguation by counting overlaps between dictionary definitions	Corpus-based approach	It is not a syntactic technique, and it is not dependent on global information
Wu and Palmer [12]	Path length to the root node from the least common super-concept of two concepts	Taxonomic-based approach	It is a syntactic-semantic technique
Resnik [13]	Evaluate the semantic similarity in a taxonomy, based on the notion of information content	It is a hybrid approach that combines corpus-based statistical methods with knowledge-based taxonomic information	It is a semantic technique
Jian and Conrath [14]	It combines a lexical taxonomy structure with corpus statistical information	It is a hybrid approach that combines taxonomic-based approach with corpus-based approach	Based on the tests reported, this combined approach outperforms other computational models
Lin [15]	This measure uses the amount of information needed to state the commonality between the two concepts and the information needed to describe these terms	Information content measure (corpus-based)	It is a universally applicable similarity measure, independent of domain or form of knowledge representation
Hirst Onge [16]	This measure states that two lexicalized concepts are semantically close if their synonyms are connected by a path that is not too long, and it is not changing its direction frequently	Information content measure that uses lexical chains as a context	It is a semantic and context-based technique
Leacock and Chodorow [17]	This measure finds the shortest path length between two concepts, and scales that value by the maximum path length in the is-A hierarchy in which they occur	This is an information content measure that adds topical to local context using a statistical classifier	It is a semantic and context-based technique

Table 2.
Summary of similarity measurements.

definition; from this, the elements extracted for similarity calculation are the name of the operations and their associated input and output parameters.

4. Semantic similarity measures

Measuring the similarity between two concepts is not a new topic. Throughout the last decades, many measures of similarity have been reported using different perspectives: syntactic, semantic, contextual, etc. In this work, we use a set of semantic similarity measurements based on WordNet.[1] Computing similarity

---

[1]https://wordnet.princeton.edu/

between all Web services in the collection is a process executed in pairs. Let *W* be the tuple that represents all Web services in the collection as follows:

$$W = \langle P, I, O \rangle \tag{1}$$

where *P*, represents all operation names; *I,* is the set of input parameters; *O*, is the set of output parameters.

In particular, in this work the similarity measures were applied only on the operation names. Therefore, the similarity calculation takes as input a matrix of all operation names in the collection of Web services, that is, as follows:

$$(\text{Let}) \; P = \{p_1, p_2, p_3, ..., p_n\} \tag{2}$$

$$\text{Input Matrix} = \left\{ \left(p_i, q_i \right) \in P \times P, 1 \leq i \leq n \right\} \tag{3}$$

Eight measures that exploit WordNet database were used to calculate the semantic similarity between Web service operations. WordNet is a lexical database available online; it is organized into five categories: nouns, verbs, adjectives, adverbs, and function words [9]. The utilization of WordNet for semantic similarity measurements is a good approach in contrast with the traditional syntactic similarity approaches, specifically in the case of service operations, as they normally include a verb indicating the main functionality of the operation method.

Additionally, an application programming interface (API) that implements a large collection of semantic similarity measures (140 methods) is available WNetSSAPI[2]. A deeper analysis and comparison of similarity measures is out of the scope of this work. **Table 2** shows a summary of the semantic similarity measures used.

With these measures, all service operations are compared, and a set of eight matrixes are created with the distances between them. **Figure 4** shows an example of the calculation of the eight similarities with operation names.

SimilarityBetween(ChangeAngleUnit , ChangeAreaUnit) =

==> Jacquard Syntactic Similarity = 0.5

==> Wu Palmer Semantic Similarity = 0.5994745994745996

==> Lin Semantic Similarity = 0.4535231912829227

==> Path Semantic Similarity = 0.33355780022446685

==> Lesk Semantic Similarity = 0.6666666666666666

==> HirstStOnge Semantic Similarity = 1.0

==> JiangConrath Semantic Similarity = 0.2985112762922702

==> LeacockChodorow Semantic Similarity = 0.8320633585750906

==> Resnik Semantic Similarity = 0.5883605316331607

Figure 4.
Example of the calculation of semantic similarities.
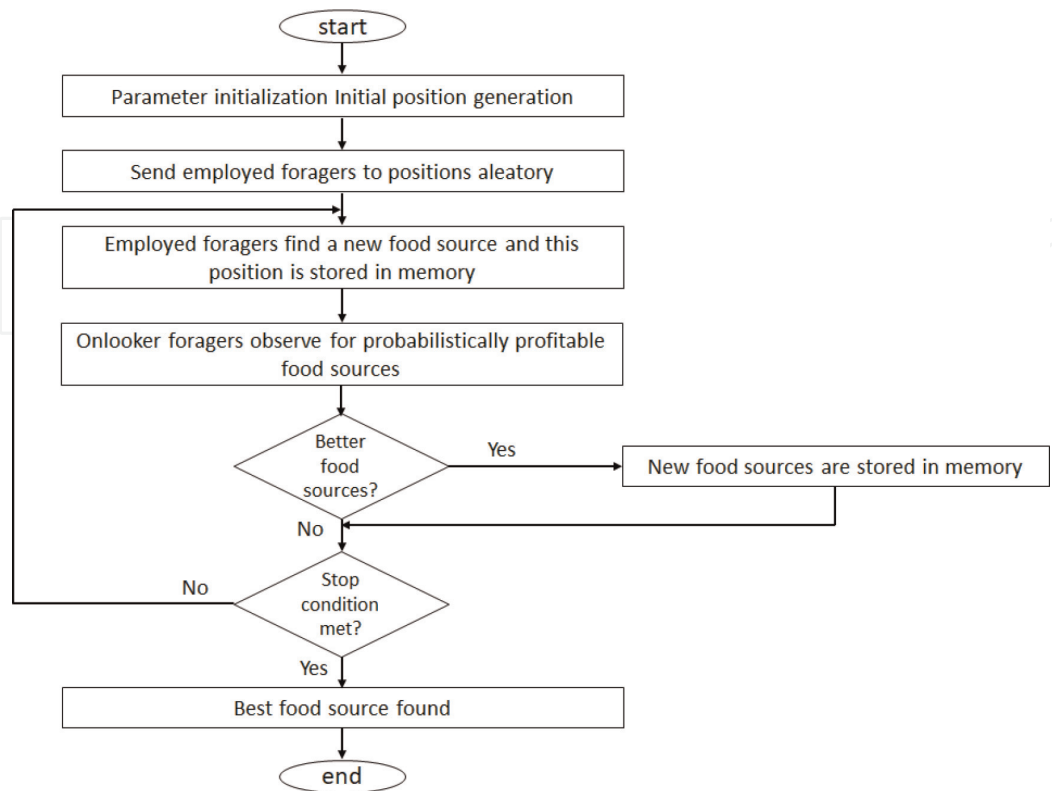
_____

[2]http://wnetss-api.smr-team.org/

5. Artificial Bee Colony algorithm

The Artificial Bee Colony (ABC) algorithm is an optimization technique that simulates the foraging behavior of honey bees and has been successfully applied to various practical problems and is a nature-inspired and swarm intelligence method that has been applied in different scenarios with good results. The ABC algorithm was proposed in 2005 by Karaboga [18, 19]; accordingly, the collective intelligence model of the Bee Colony consists of:

a. *Employed foragers* which are bees assigned (employed) to a particular food source and are exploiting it. They carry information about the food source, distance and direction to the nest and the profitability of the source, and are capable of sharing this information.

b. *Unemployed foragers* are bees that are continuously searching for food sources. These unemployed bees are subdivided into *scouts*, bees that search on the surrounding environment for new food sources, and *onlookers*, bees that wait in the nest.

The ABC algorithm has different modes of behavior:

a. Exploration is the task executed by unemployed bees to find new food sources.

b. Exploitation is the task executed by employed bees on a food source.

c. Recruitment is the action that an unemployed bee executes with forager bees to exploit a food source.

d. Abandonment of a nectar source occurs when a better food source is found.

Figure 5.
ABC algorithm general workflow.

An important behavior of *employed* and *onlooker* bees is their capacity of sharing information (memory) to choose and adjust the food source value. This value depends on the proximity to the nest, the richness or concentration of honey energy [18]. The exchange of information occurs during the waggle dance at the hive. *Onlooker* foragers watch numerous dances at the dancing area and decide to employ themselves at the most profitable food source. When an *onlooker* forager recruit starts searching and locates the food source, then it utilizes its own capability to memorize the location and starts exploiting it. The *onlooker* forager becomes an *employed* forager. In the ABC algorithm the set of possible solutions represent the food sources, and the food source value represents the quality of the solution. A general representation of the ABC workflow algorithm is presented in **Figure 5**.

6. Hybrid algorithm description

A hybrid algorithm was proposed to make the ABC auto-adjustable during each iteration to decide the number of clusters by incorporating K-means and a Consensus method. In particular, K-means is used to select the elements inside each generated cluster to decide centroids for similarity calculations. The solution of the algorithm is represented as a vector of size n (number of Web services to cluster) where each position of the element in the vector is the group to which it belongs to.

6.1 Objective function

The objective function of this hybrid algorithm is shown in Eq. (4):

$$\text{Min} \sum_{\substack{i=1 \\ x \in c_i \\ y \in c_i}}^{C} d(x_i, y_i) \tag{4}$$

where d, distance between centroid of cluster y_i and a service x_i; y_i, centroid of cluster i; x_i: one of the services included in cluster i. No group of services should be empty, and there should be no intersection between groups.

6.2 Filtering similarities

The first stage of the hybrid algorithm consists of filtering of the eight matrices that contain the information of similarities between Web services. The filtering consists of discarding values that exceed the limits allowed and established by the similarity measures, as a result of this filtering, new matrices are generated with a degree of 95% certainty in the measurements. Eq. (5) shows the filtering calculation:

$$X - 1.96 \frac{\sigma}{\sqrt{N}} \leq \mu \leq X + 1.96 \frac{\sigma}{\sqrt{N}} \tag{5}$$

where X, average matrix; 1.96, table value; σ, standard deviation; N, element of the similarity matrix; μ, average similarity.

6.3 Food source representation

After the filtering process, all obtained data is stored in an average matrix (food sources) discarding the positions that contain null or zero information; that is,

the average matrix was calculated considering only those values that were filtered and accepted as feasible values that contribute with information to the hybrid algorithm.

6.4 Bee's representation

Using the average matrix a set of arrays is generated representing the bees and other important information as the number of groups and centroids. **Table 3** shows the structure of the solution generated.

a. *Max group*. This hybrid algorithm does not require the user to indicate how many groups it should generate; the algorithm as it iterates determines how many groups to generate based on the results obtained on the previous iteration and applying the Consensus method. Initially, the *i-th* bee will generate a random number of groups, based on a discrete uniform distribution with limits 2 to N/2; in the subsequent iterations of the algorithm, the *i-th* bee will produce a random number γ (based on a normal distribution of the weighted variance of the Max group determined by the colony in the previous iteration), then a simple rounding will be applied to γ.

b. *Centroids*. Next, the *i-th* bee must determine the centroids of the γ groups involved in the classification. The centroid sub-vector is formed by N integer elements, where x_{ijk} is zero if it is not considered as a centroid for any group, in case $x_{ijk} = a$ implies that the *j-th* Web service is centroid of the *k-th* group. Initially such values are assigned randomly; later by applying Eq. (6), the values of the sub-vector are obtained:

$$X'_{new} = \text{round}\left(X'_i - \phi(X'_i - X'_s)\right) \qquad (6)$$

where X'_{new}, new vector; X'_i, first vector generated by the algorithm; ϕ, aleatory number between 0 and 1

a. β represents the sum of the similarity between centroids of the groups, while α is the summation of the similarity between members of each group to the corresponding centroid. The objective is to minimize β and maximize α simultaneously.

b. For each bee the *assessment* function is calculated using $f(x) = \frac{1}{\beta} + \alpha$; the objective is to obtain the highest $f(x)$ value. During each iteration, the $f(x)$ value is stored in the solution vector.

c. Normalization is used to determine the quality of the food source found. Normalization is calculated with $n_i = \frac{f(x_i)}{\sum_{i=1}^{N} f(x_i)}$. The bee will abandon the food source if there is a better food source in the near surrounding.

Groups generated	Centroids	β	α	Normalization	Assessment	Max group	Limit
32,231	32,001	1.068	1.55	0.301	0.36	3	5

Table 3.
Composition of the vector with information of generated groups.

Input: Number of bees (B), Filtered similarity matrix, Number of Web services to cluster (M), number of maximum iterations (v), value of φ

Output: Best classification found by the ABC algorithm

1   For each of the N food sources execute algorithm of Fig. 7
2   Calculate normalization value
3   v'=0
4   While v'≤v
5   Compute mean ($\bar{c}$) and variance (S_c) weighed (base on normalization values) of groups used by employed bees at iteration v'-1.
6   For each bee
7   Search for food source at neighborhood of *i* using algorithm of Fig. 8
8   If value of f(x_{new}) is better than f(x)
9   Move employed bee to the new food source
10                                      $limit_i$=0
11   Else
12                                      $limit_i$= $limit_i$+1;
13   End If
14   If ($limit_i$ = g)
15   Aleatory generate a new food source
16   Move the i-th bee to new food source
17   End If
18   End For
19   End While
20   Select bee with best *f(x)*

Figure 6.
ABC algorithm pseudocode.

1   Randomly determine the number of groups to generate (a uniform distributed value between 2 and *M*/2)
2   Randomly determine the centroids of each group. (Construction of the centroid vector)
3   Classify each of the *M* Web services in one of the groups according to the similarity to the centroids (application of the classic *k-means* algorithm)
4   Determine the corresponding assessment value.

Figure 7.
Pseudocode of the "Generation of N initial food sources."

d. Limit is a *counter* that indicates the number of iterations that the employed bee has being exploiting the current food source. The employed bee is obligated to abandon the food source when a *g* number of iterations is achieved (**Figures 6–8**).

1   Let a be an aleatory number with normal distribution, mean ($\bar{c}$) and variance (S_c)

2   The number of groups of the *i-th* bee is the rounded value of a that is within the limits

3   *i'* is a food source different from *i*, which is randomly selected based on the value of normalization.

4   With the centroid sub-vectors of *i* and *i'*

5   Generate x $_{new}$, which is a combination of centroid sub-vectors of *i* and *i'* using Equation (2)

6   Repair x$_{new}$ in such a way that it contains exactly a cells different to zero, each with a different value between *i* and a.

7   For each of the *m* Web services

8   Determine the similarity to the centroids of the *j-th* service.

9   Assign the *j-th* service to the group with the highest similarity.

10   End For

11   Calculate β α and f$\left(x_{new}\right)$

Figure 8.
Pseudocode of "Search for a food source in the neighborhood of i."

7. Experimentation and results

Experiments were carried out with three service collections, which involved grouping 50, 647, and 1027 Web services, respectively. For the creation of the similarity matrices, eight different semantic similarity measures were calculated for each collection of Web services. The resulting similarity matrices were filtered to discard values that exceed the allowed limits.

The determination of the parameters used by the algorithm was performed by brute force, resulting in using 10 and the value of "phi" (φ) set to 0.8; the value of the limit that makes up each vector from the beginning was established with the value of 10.

In order to characterize the behavior of the algorithm, 20 executions were made with 100, 200, and 500 iterations of the algorithm. For each of the executions, the best value of $f(x)$ found by the bees was determined. Subsequently, a statistical analysis of the results found was carried out. **Table 4** shows the average values of $f(x)$ for each of the instances with 100, 200, and 500 iterations, respectively.

Based on the results shown in **Table 4**, it can be affirmed that for the 50 services instance, the best values are found with 500 iterations, while for the instance of 647 services, the best values are obtained with 100 iterations. Finally, for the instance of 1027, it is obtained with 200 iterations.

Matrix size	100 iterations	200 iterations	500 iterations
	$f(x)$	$f(x)$	$f(x)$
50	0.4631	0.5011	0.5169
647	0.4457	0.4152	0.4228
1027	0.4414	0.4782	0.4542

Table 4.
Average results of collections executing 100, 200, and 500 iterations.
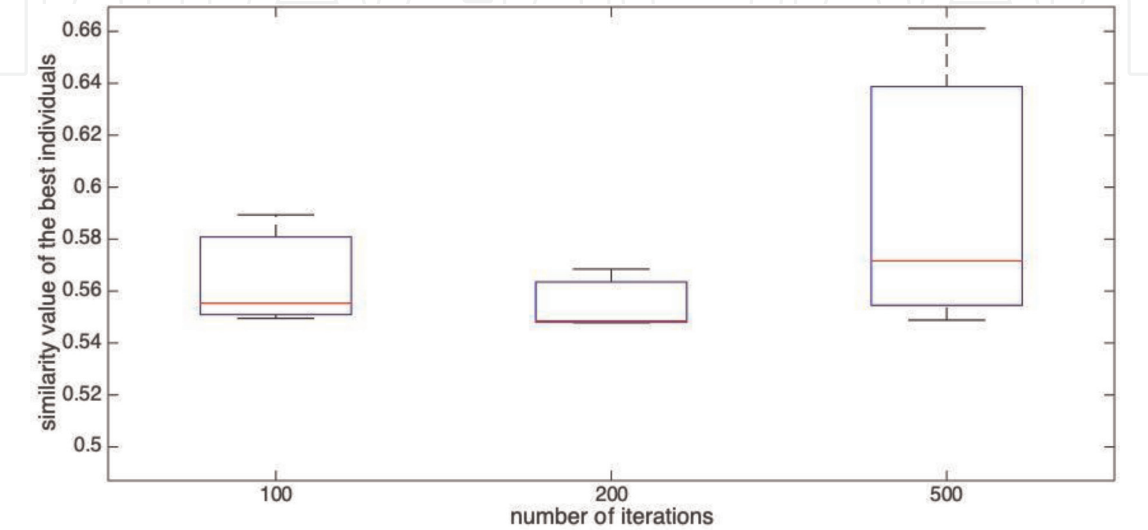
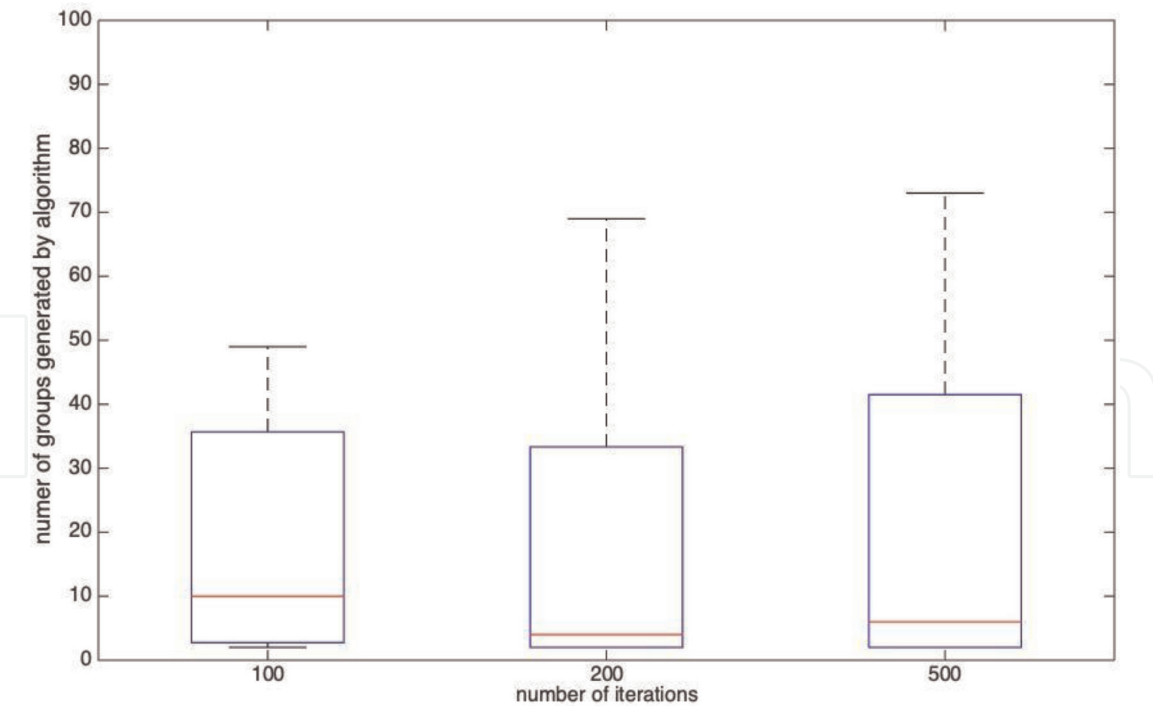| Collection-iterations	Best	Worst	Mean	Variance	Deviation
50–100	0.58936208	0.29274988	0.45901481	0.00593354	0.0770295
50–200	0.54853117	0.44904535	0.50052478	0.00088301	0.02971549
50–500	0.57163535	0.46554965	0.51820976	0.00090081	0.03001343
647–100	0.54947584	0.2153186	0.44401451	0.01027133	0.10134758
647–200	0.54778745	0.20257995	0.40923046	0.01554565	0.12468218
647–500	0.54881541	0.22262872	0.41911288	0.01280278	0.11314939
1027–100	0.55531806	0.27378002	0.44488931	0.00847871	0.09207991
1027–200	0.56846422	0.28919387	0.47529505	0.00772078	0.08786795
1027–500	0.66112025	0.20302145	0.4674717	0.00977301	0.09885856

Table 5.
Summary with the best results of similarity.

Collection-iterations	Best	Worst	Mean	Variance	Deviation
50–100	2	10	4.31578947	6.22105263	2.49420381
50–200	2	4	3	0.89210526	0.94451324
50–500	2	6	3.94736842	2.36578947	1.53811231
647–100	3	49	18.0526316	221.207895	14.8730594
647–200	2	67	18.9473684	305.831579	17.488041
647–500	2	73	23.2105263	426.368421	20.6486905
1027–100	2	110	31.2631579	1066.82895	32.6623475
1027–200	2	69	22.1052632	355.884211	18.8648936
1027–500	2	123	31	1179.21053	34.3396349

Table 6.
Summary with the best results of clusters.

Table 5 shows the results obtained with configurations: the best, worst, average, variance, and standard deviation of $f(x)$ over the 20 executions of the algorithm (**Table 6**).

Figure 9.
Comparison of similarity between the best solutions found with 1027 services.

Figure 10.
Comparison of the number of groups in the best case for 1027 services.

Figures 9 and **10** show a comparison of results obtained from the algorithm with 100, 200, and 500 iterations of the instance of 1027 services. It can be seen that the best values of $f(x)$ are produced with 500 iterations; however the result of the number of groups involved is more stable and better with 200 iterations.

8. Conclusions

This chapter describes a hybrid algorithm for Web services clustering; the approach is based on the ABC optimization algorithm combined with the K-means and Consensus. The clustering process starts from the extraction and processing of WSDL documents, then the calculation of semantic similarities between pairs of Web services operations. The semantic similarity measurements are based on WordNet, combining corpus-based and taxonomic approaches. As a result of the calculations, eight matrices are generated which are the input data set to the hybrid bio-inspired algorithm.

Biologically inspired algorithms offer advantages over conventional clustering methods, such as the ability to find a near optimal solution by updating the candidate solutions iteratively and have self-organizing behavior.

The clustering algorithm designed is based on the behavior of the bees but was improved by incorporating K-means and Consensus so that the algorithm adjusts itself in each iteration. A series of experiments were conducted with the hybrid ABC algorithm, using optimal values for each adjustable parameter. The experiments were carried out with three collections with 50, 647, and 1027 Web services, and the algorithm was executed with variants of 100, 200, and 500 iterations. The hybrid ABC algorithm has shown good results for Web services clustering.

As future work, more combinations of semantic similarities, as well as incorporating more information of the description of the services and the data types incorporated in the XML service definitions.

15

Also the incorporation of other bio-inspired algorithms (swarm intelligence) for the classification and clustering of Web services, such as Particle Swarm Optimization (PSO), Bat Algorithm (BA), and Bird Swarm Algorithm (BSA).

Author details

Maricela Bravo*, Román A. Mora-Gutiérrez and Luis F. Hoyos-Reyes
Autonomous Metropolitan University, Ciudad de México, México

*Address all correspondence to: mcbc@azc.uam.mx

IntechOpen

References

[1] Liang Q, Li P, Hung PC, Wu X. Clustering web services for automatic categorization. In: SCC'09: IEEE International Conference on Services Computing. IEEE; 2009. pp. 380-387

[2] Platzer C, Rosenberg F, Dustdar S. Web service clustering using multidimensional angles as proximity measures. ACM Transactions on Internet Technology. 2009;**9**(3):11

[3] Pop CB, Chifu VR, Salomie I, Dinsoreanu M, David T, Acretoaie V, et al. Biologically-inspired clustering of semantic web services. Birds or ants intelligence? Concurrency and Computation: Practice and Experience. 2012;**24**(6):619-633

[4] Du YY, Zhang YJ, Zhang XL. A semantic approach of service clustering and web service discovery. Information Technology Journal. 2013;**12**(5):967-974

[5] Wu J, Chen L, Zheng Z, Lyu MR, Wu Z. Clustering web services to facilitate service discovery. Knowledge and Information Systems. 2014;**38**(1): 207-229

[6] Prakash J, Singh PK. Evolutionary and swarm intelligence methods for partitional hard clustering. In: 2014 International Conference on Information Technology (ICIT). IEEE; 2014. pp. 264-269

[7] Sahoo G. A two-step artificial bee colony algorithm for clustering. Neural Computing and Applications. 2017; **28**(3):537-551

[8] Kotekar S, Kamath SS. Enhancing web service discovery using meta-heuristic CSO and PCA based clustering. In: Progress in Intelligent Computing Techniques: Theory, Practice, and Applications. Singapore: Springer; 2018. pp. 393-403

[9] Fellbaum C. WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press; 1998

[10] Lesk M. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In: Proceedings of SIGDOC '86. 1986

[11] Banerjee S, Pedersen T. An adapted Lesk algorithm for word sense disambiguation using WordNet. In: Computational Linguistics and Intelligent Text Processing. Berlin Heidelberg: Springer; 2002. pp. 136-145

[12] Wu Z, Palmer M. Verbs semantics and lexical selection. In: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics; 1994. pp. 133-138

[13] Resnik P. Using information content to evaluate semantic similarity. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence. Montreal; 1995. pp. 448-453

[14] Jiang JJ, Conrath DW. Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of International Conference on Research in Computational Linguistics, Taiwan. 1997

[15] Lin D. An information-theoretic definition of similarity. In: Proceedings of the 15th International Conf. on Machine Learning. San Francisco, CA: Morgan Kaufmann; 1998. pp. 296-304

[16] Hirst G, St-Onge D. Lexical Chains as Representations of Context for the Detection and Correction of Malapropisms; 1998. pp. 305-332

[17] Leacock C, Chodorow M. Combining local context and WordNet

similarity for word sense identification. In: WordNet: An Electronic Lexical Database. Vol. 49. 1998. pp. 265-283

[18] Karaboga D. An Idea Based on Honey Bee Swarm for Numerical Optimization. Vol. 200. Technical Report: tr06. Erciyes University, Engineering Faculty, Computer Engineering Department; 2005

[19] Karaboga D, Ozturk C. A novel clustering approach: Artificial Bee Colony (ABC) algorithm. Applied Soft Computing. 2011;**11**(1):652-657