

# **AGRUPAMIENTO DE SERVICIOS WEB MEDIANTE EL ALGORITMO COLONIA ARTIFICIAL DE ABEJAS**

***Maricela Claudia Bravo Contreras***

Universidad Autónoma Metropolitana, Unidad Azcapotzalco  
*mcbc@correo.azc.uam.mx*

***Román Anselmo Mora Gutiérrez***

Universidad Autónoma Metropolitana, Unidad Azcapotzalco  
*rmg@correo.azc.uam.mx*

***Iván Flores Sánchez***

Universidad Autónoma Metropolitana, Unidad Azcapotzalco  
*floressanchezivan@gmail.com*

***Roberto Alfonso Alcántara Ramírez***

Universidad Autónoma Metropolitana, Unidad Azcapotzalco  
*raar@correo.azc.uam.mx*

## **Resumen**

En este artículo se reporta un enfoque híbrido para la clasificación de servicios Web basada en el algoritmo bio-inspirado de Colonia Artificial de Abejas (ABC), K-medias y un método de consenso. La clasificación se realiza mediante la utilización de medidas de similitud semántica, es decir se comparan semánticamente servicios web, obteniendo de cada métrica un valor, que es el valor de similitud de cada servicio entre todos los demás servicios, estos datos son almacenados en diferentes matrices, para su posterior calculo mediante el algoritmo híbrido y obtener una mejor clasificación de los servicios web.

**Palabra(s) Clave(s):** Agrupamiento de servicios Web, colonia artificial d abejas, métodos heurísticos.

## **1. Introducción**

El agrupamiento de servicios Web es la tarea de identificar las características de los servicios y determinar a qué clase o grupo pertenecen. La implementación de mecanismos de agrupamiento automática de servicios Web es relevante porque a través de ésta se pueden organizar de acuerdo a algún criterio seleccionado grandes colecciones de servicios en la Web. Contar con grandes colecciones de servicios organizados y agrupados es un objetivo constante de los directorios de servicios Web públicos, porque solamente a través de un directorio bien organizado se puede facilitar la automatización de la búsqueda, selección, substitución e invocación de cualquier servicio Web.

En lo referente al agrupamiento, en los últimos años se han implementado métodos basados en analogías biológicas para resolver problemas complejos de computación y optimización entre los que destacan algoritmos genéticos, estrategias evolutivas, comportamiento de algunas especies de animales, etcétera. Muestra de ello es la creciente tendencia para construir soluciones inspiradas en la naturaleza para resolver problemas complejos y difíciles. Los seres vivos como los animales y las plantas y el mismo clima exhiben extraordinarios, complejos y fascinantes fenómenos naturales y uno de ellos es la forma de encontrar la mejor solución para resolver el problema y mantener el equilibrio perfecto entre sus componentes o el medio que los rodea.

Esta es la idea principal de la computación inspirada en la naturaleza, es decir, imitar los pasos que ha desarrollado la naturaleza y adaptarlos a un problema, convirtiéndolo así en un algoritmo bio-inspirado [1]. Son muchos los algoritmos bio-inspirados que imitan la naturaleza para resolver problemas de optimización [2].

El empleo de métodos convencionales para resolver problemas de optimización requiere enormes esfuerzos y recursos computacionales, que tienden a fallar en cuanto aumenta el tamaño del problema [3]. Esta es la razón principal de la implementación de un algoritmo que explote la característica de inteligencia colectiva, dicha característica la posee el algoritmo Colonia Artificial de Abejas como alternativa en la optimización de clasificar servicios web.

En este artículo se presenta un enfoque innovador para el agrupamiento de servicios Web que utiliza un algoritmo híbrido basado en el algoritmo de Colonia Artificial de Abejas, K-Medias y Consenso. Para la evaluación del método de agrupamiento se utilizó una colección de 1027 servicios Web públicos con los cuales se realizaron las etapas del procesamiento, clasificación, pruebas y análisis. El algoritmo recibe como entrada una colección de 8 matrices de mediciones de similitud semántica, las cuales se calculan utilizando todas las descripciones de servicios Web.

## **2. Antecedentes**

El trabajo que se reporta en este artículo forma parte de un proyecto de investigación de mayor alcance cuyo objetivo es el de diseñar e implementar un directorio semántico de servicios web y recursos de programación utilizando métodos eficientes, totalmente automatizados, que permitan la organización, composición y clasificación de servicios web con un enfoque semántico. En la figura 1 se muestra la arquitectura general del directorio semántico, el cual está compuesto por módulos cada uno con un objetivo específico, en particular, con este trabajo se contribuyó con el módulo de clasificación y agrupamiento de servicios web y para su funcionamiento hace uso de los módulos que le preceden:

- Módulo de recuperación de servicios Web. Las descripciones de servicios Web son buscados y recuperados de la red mediante el uso de programas autónomos que buscan y recuperan recursos disponibles en la Web (también conocidos como crawlers o arañas Web).
- Módulo de análisis y extracción de datos. Los servicios recuperados son procesados mediante programas que barren el archivo y extraen la información relevante de los servicios. La información específica de los servicios se almacena en varias ontologías de servicios Web.
- Módulo de cálculo de similitudes. Los datos extraídos son utilizados para realizar el cálculo de la similitud semántica utilizando ocho medidas de similitud.

- Módulo de clasificación y agrupamiento. La clasificación y agrupamiento de servicios Web se realiza utilizando diferentes algoritmos, por ejemplo: máquinas de soporte vectorial, redes neuronales, optimización por colonia de hormigas, entre otros. En particular en este artículo se reporta un método de agrupamiento de servicios mediante un algoritmo híbrido derivado del algoritmo Colonia Artificial de Abejas (ABC).

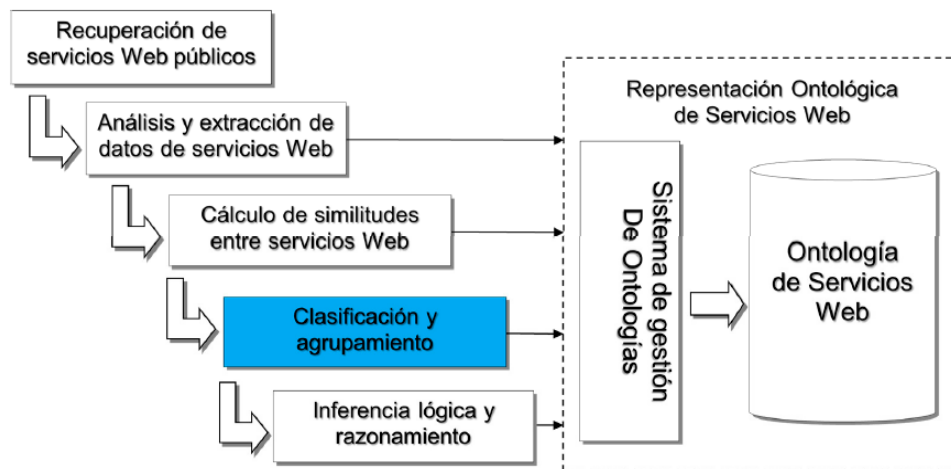


Figura 1 Arquitectura general del directorio semántico de servicios Web.

### 3. Diseño de la Solución

La estructura general de la solución propuesta emplea un algoritmo híbrido y se muestra en la figura 2, cuyo funcionamiento se describe con más detalle en las siguientes secciones.

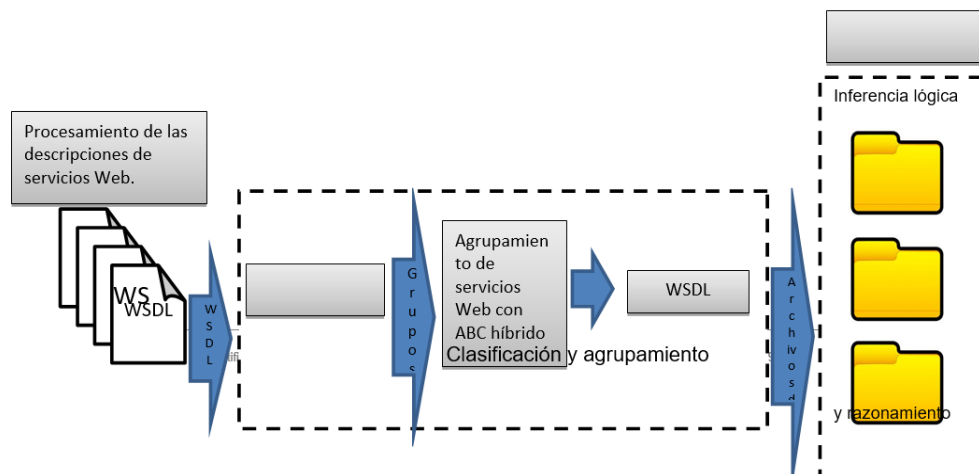


Figura 2 Arquitectura general de la solución propuesta para el agrupamiento de servicios.

El proyecto consta de los siguientes módulos:

- Colección de servicios Web local. Consiste de una gran colección de servicios Web, que se usan de entrada a todo el proceso para ser agrupados, los servicios Web de esta colección fueron previamente recuperados y almacenados localmente mediante arañas Web.
- Módulo de procesamiento de las descripciones de los servicios web. Este módulo realiza la tarea de recuperar y preprocesar toda la información de los servicios web para su posterior agrupamiento.
- Módulo de agrupamiento de los servicios web mediante el algoritmo híbrido. Este módulo determina cuántos grupos deben generarse, determina los centros de los grupos, y agrupa los servicios en función de los centros generados.
- Repositorio para los servicios web agrupados. Este módulo alberga todos los servicios web organizados en grupos, el objetivo de este módulo es obtener un esquema de agrupamiento óptimo.

•

#### **4. Descripción del Algoritmo Colonia Artificial de Abejas**

La idea principal del algoritmo Colonia Artificial de Abejas (ABC) es imitar el comportamiento de búsqueda y recolección de alimento haciendo uso de recursos muy específicos como son:

- Exploración. Acción que realizan las abejas desempleadas para buscar y localizar nuevas fuentes de alimento.
- Explotación. Tarea que llevan a cabo las abejas obreras en una fuente de alimento.
- Reclutamiento. Acción que realizan las abejas desempleadas con las abejas obreras para explotar una fuente de alimento.
- Abandono. Sucede cuando se encuentra una mejor fuente de alimento.
- El algoritmo ABC está compuesto por dos grupos de abejas: las abejas obreras y las abejas en espera. Las abejas en espera se subdividen a su vez en: abejas exploradoras y en abejas espectadoras, estas dos clases de abejas (obrero y espectadora) tienen la característica de elegir y ajustar

mediante su experiencia (memoria) la fuente de alimento. En el algoritmo ABC se asume que cada posición de la fuente de alimento equivale a una solución factible y la cantidad de alimento que se recolecta de la fuente de alimento representa la calidad asociada a la solución [2], la figura 3 resume el algoritmo del ABC.

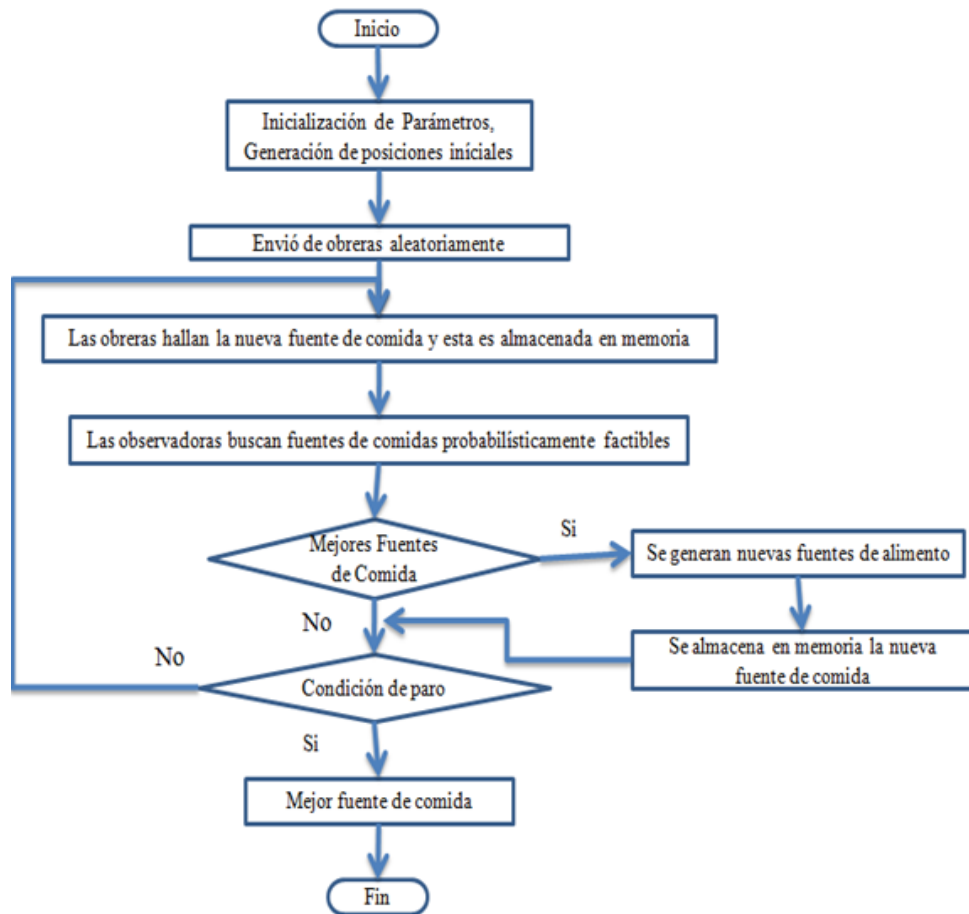


Figura 3 Diagrama general del algoritmo ABC.

La figura 4 muestra la gráfica del comportamiento del algoritmo ABC, donde el color azul representa a las abejas obreras, el color verde representa a las abejas exploradoras, el color púrpura representa la mejor solución obtenida, el color rojo representa el óptimo deseado. La gráfica del lado izquierdo (a) muestra como comienza el algoritmo ABC posicionando a las obreras aleatoriamente, del lado derecho (b) muestra el resultado después de una serie de ejecuciones que la solución factible es alcanzada.

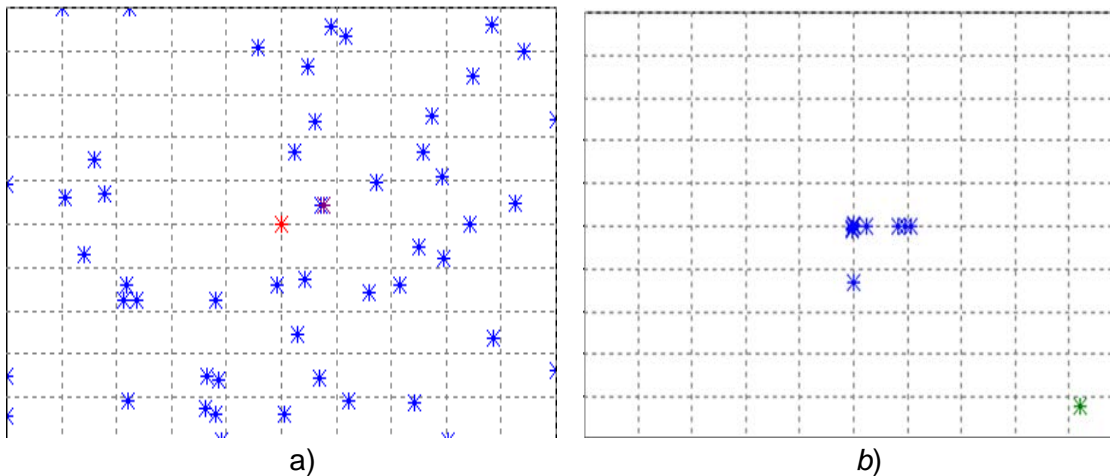


Figura 4 Gráfica del comportamiento del algoritmo ABC.

La combinación de otros métodos de agrupamiento como es el caso del método de consenso, permite al algoritmo bio-inspirado la creación de grupos que se generan desde 2 grupos (valor mínimo de grupos permitidos) hasta el número de servicios Web que se utilizan entre 2 (valor máximo de grupos que se podrán generar). Para el caso del método K-Medias, el algoritmo selecciona elementos dentro de los grupos generados anteriormente convirtiéndolos en centros con los cuales se podrá medir la similitud, también el algoritmo híbrido hace uso de métodos heurísticos al generar números aleatorios que el algoritmo utiliza para el cálculo en la generación de grupos.

La solución se presenta como un vector de entrada de tamaño n (número de servicios Web a agrupar) donde el elemento en el vector, devolverá el grupo al que pertenece. La función objetivo que el algoritmo híbrido emplea está dada por ecuación 1.

$$\text{Min } \sum_{i=1}^c d(x_i, y_i)$$

$x_i \in c_i$   
 $y_i \in c_i$

Donde:

$d$ : Distancia

$y_i$ : Centro del i-ésimo grupo

$x_i$ : Uno de los servicios incluidos en el i-ésimo grupo. Ningún grupo puede estar vacío y no puede existir intersección entre los grupos.

## 5. Descripción del Algoritmo Híbrido

La primera etapa que realiza el algoritmo híbrido es la filtración de las 8 matrices que contienen la información de similitudes de los servicios Web mediante la ecuación 2, descartando valores que excedan los límites permitidos y establecidos por las mismas métricas, se generan nuevas matrices filtradas con un 95% de certeza en las mediciones.

$$X - 1.96 \frac{\sigma}{\sqrt{N}} \leq \mu \leq X + 1.96 \frac{\sigma}{\sqrt{N}} \quad (2)$$

Donde:

$X$  : Matriz Promedio.

1.96 : Constante.

$\sigma$  : Desviación estándar.

$N$  : Elemento de la matriz de similitud.

$\mu$  : Similitud Promedio.

Posteriormente filtradas las 8 matrices todos los datos obtenidos se almacenan en una matriz promedio (fuentes de alimentos) descartando las posiciones que contengan información nula ó 0 es decir se promediaron todos aquellos valores que si fueron filtrados y aceptados como valores factibles cuya información si aporta gran ayuda al algoritmo.

Para después leer la matriz de similitud promedio para generar por primera vez una serie de arreglos (abejas) los cuales tendrán almacenada información, por ejemplo, los grupos generados, los centros seleccionados etcétera (Figura 5).

32231	32001	1.068	1.55	0.301	0.36	3	5
Grupos generados		$\beta$	$\alpha$	Normalización	Valoración	Max grupo	Limite

Figura 5 Composición del vector con información de los grupos generados.

En la figura 5 se muestra la estructura de una solución generada por el híbrido propuesto. A continuación se describe a detalle cada uno de los elementos que la conforman:



- a) Max grupo. El algoritmo propuesto no requiere que el usuario a priori indique cuantos grupos debe generar, a medida que itera éste determina cuantos grupos debe generar en función de los resultados obtenidos y la idea de consenso. Inicialmente, la  $i$ -ésima abeja generara un número aleatorio el número de grupos a generar, con base a una distribución uniforme discreta con limites 2 a  $N/2$ ; en las subsecuentes iteraciones del algoritmo, la  $i$ -ésima abeja producirá un número aleatorio  $\phi$  (con base a una distribución normal de media la varianza ponderada del Max grupo determinado por la colonia en la iteración anterior), posteriormente se aplicara un redondeo simple a  $\phi$  (en caso de salir de los límites 2 a  $N/2$  se asignará el valor violado).
- b) A continuación, la  $i$ -ésima abeja debe determinar los centroides de los  $\gamma$  grupos implicados en la clasificación. El sub-vector centroides se conforma por  $N$  elementos enteros; donde  $x_{ijk}$  es cero si no es considerado como centriode para ningún grupo, en caso que  $x_{ijk} = a$  implica que el  $j$ -ésimo servicio web es centriode del  $k$ -ésimo grupo. Inicialmente dichos valores se asignan aleatoriamente.
- c) Posteriormente mediante la ecuación 3 se obtienen los valores del sub-vector.

$$X'_{new} = \text{round} (X'_i - \phi(X'_i - X'_s)) \quad (3)$$

Dónde:

$X'_{new}$ : Nuevo vector generado.

$X'_i$ : Primer vector que genera el algoritmo.

$\phi$ : Numero Aleatorio entre 0 y 1.

Debe hacerse notar, que el vector  $X'_{new}$  es reparado con objeto que sólo  $k$  elementos sean diferentes de cero y todos los grupos tengan un centriode diferente.

- d) El sub-vector grupos generados se construye tomando como base el sub-vector centriode los elementos cero se remplazan por el valor  $k'$  que es el centriode más cercano al servicio web.

- e) es la sumatoria de la similitud entre centroides de los grupos mientras que  $\alpha$  es la sumatoria de la similitud entre miembros de cada grupo al centroide correspondiente. Se busca minimizar  $\beta$  y maximizar  $\alpha$  de manera simultánea.
- f) Para cada una de las abejas se determina el valor de  $f(x)=1/\beta + \alpha$  se busca que este valor de  $f(x)$  sea lo mayor posible. El valor de  $f(x)$  se guarda en el vector la solución en la celda de valoración.
- g) La normalización implica  $n_i = \frac{f(x_i)}{\sum_{i=1}^N f(x_i)}$  este elemento servirá para determinar la calidad de la fuente de alimento encontrada. Una abeja abandonara su fuente de alimento actual si la fuente de alimento explorada en su vecindario es mejor.
- h) Limite es un contador que indica el número de iteraciones que la abeja ha permanecido en la fuente de alimento actual, cuando se llega a un valor  $g$ , y siempre y cuando la abeja no se encuentre en la mejor fuente de alimento encontrada, la  $i$ -ésima abeja es obligada a abandonar su fuente de alimento actual.

En las figuras 6 , 7 y 8 se muestra el pseudocódigo del algoritmo propuesto.

```
Entrada: Número de abejas (N), Matriz de similitud filtrada, Número de servicios web a agrupar (M), número de iteraciones máximas a realizar por el algoritmo (v), valor de  $\Phi$ 
Salida: Mejor clasificación encontrada por el ABC
1 Para cada una de las N fuentes de alimento aplicar el algoritmo de la Fig. 7
2 Calcular el valor de normalización
3  $v=0$ 
4 Mientras  $v \leq v$ 
5 Determinar el promedio ( $\bar{c}$ ) y varianza ( $S_c$ ) ponderado (con base en los valores de la normalización) de los grupos empleados por las abejas en la  $v-1$  iteración.
6 Para cada una de las abejas
7 Buscar una fuente de alimento en el vecindario de  $i$  con base en la Fig. 8
8 Si el valor de  $f(x_{new})$  es mejor que el valor  $f(x)$ 
9 Mover a la abeja obrera a la nueva fuente de alimento
10 limite =0
11 Si no
12 limite = limite +1;
13 Termina Si
14 Si limite = g
15 Aleatoriamente generar una nueva fuente de alimento
16 Mover a la  $i$ -ésima abeja a esa nueva fuente de alimento
17 Termina Si
18 Termina Para
19 Termina mientras
20 Seleccionar la abeja con mejor valor de  $f(x)$ 
```

Figura 6 Pseudocódigo del híbrido generado.

- 1 Determinar aleatoriamente el número de grupos a construir (es un valor uniforme distribuido entre 2 y  $M/2$ )
- 2 Determinar aleatoriamente los centroides de cada grupo. (Construcción del sub-vector centroides)
- 3 Colocar a cada uno de los  $M$  servicios web en uno de los grupos en función de la similitud a los centroides (aplicación del algoritmo de k-centroides clásico)
- 4 Determinar el valor de valoración de correspondiente

Figura 7 Pseudocódigo de la fase “Generación de  $N$  fuentes de alimento iniciales”.

- 1 Sea  $\alpha$  un numero aleatorio con distribución normal con media ( $\bar{x}$ ) y varianza ( $S_c$ )
- 2 El número de grupos a utilizar por la  $i$ -ésima abeja es el valor redondeado  $\alpha$  que se encuentre dentro de los límites
- 3  $i'$  es una fuente de alimento distinta a  $i$  seleccionada aleatoriamente con base en el valor de la normalización.
- 4 Tomar los sub-vectores centroides de  $i$  e  $i'$
- 5  $x_{new}$  es una combinación de los sub-vectores centroides de  $i$  e  $i'$  mediante la ecuación (2)
- 6 Reparar  $x_{new}$  de forma tal que contenga exactamente  $\alpha$  celdas distintas de cero, cada una tome un valor entre  $i$  y  $\alpha$  distinto.
- 7 Para cada uno de los  $m$  servicios web
- 8     Determinar la similitud a los centroides del  $j$ -ésimo servicio.
- 9     Asignar al  $j$ -ésimo servicio el grupo de mayor similitud.
- 10 Termina Para
- 11 Calcular  $\beta$ ,  $\alpha$  y  $f(x_{new})$

Figura 8 Pseudocódigo de la fase “Buscar una fuente de alimento en el vecindario de  $i$ ”.

## 6. Implementación

Se clasificaron tres instancias, las cuales implicaron agrupar de 50, 647 y 1027 servicios web respectivamente. Para la creación de la matriz de similitud filtrada inicialmente se calculó la similitud mediante 8 métricas distintas para cada una de las instancias, posteriormente se utilizó el procedimiento descrito con anterioridad a fin tamizar la información encontrada.

La determinación de los parámetros utilizados por el algoritmo se realizó por fuerza bruta, dando como resultado utilizar 10 y el valor de “phi” ( $\phi$ ) se estableció en 0.8, el valor del límite que conforma cada vector desde un principio se estableció con el valor de 10.

A fin de caracterizar el comportamiento del algoritmo se realizaron 20 ejecuciones con 100, 200 y 500 iteraciones del algoritmo. Para cada una de las ejecuciones se determinó el mejor valor de  $f(x)$  encontrado por las abejas. Posteriormente, se realizó un análisis estadístico de los resultados encontrados

## 7. Resultados

Las tablas 1, 2 y 3 muestran los valores promedios de  $f(x)$  para cada una de las instancias con 100, 200 y 500 iteraciones respectivamente.

Tabla 1 Resultados promedios de las instancias con 100 iteraciones.

Tamaño Matriz	F(x)
50	0.4631
647	0.4457
1027	0.4414

Tabla 2 Resultados promedios de las instancias con 200 iteraciones.

Tamaño Matriz	F(x)
50	0.5011
647	0.4152
1027	0.4782

Tabla 3 Resultados promedios de las instancias con 500 iteraciones.

Tamaño Matriz	500 iteraciones
50	0.5169
647	0.4228
1027	0.4542

Con base en las tablas anteriores, se puede afirmar que para la instancia de 50 servicios los mejores valores se encuentran con 500 iteraciones, mientras que para la instancia de 647 servicios los mejores valores se obtienen con 100 iteraciones. Finalmente, para la instancia de 1027 se obtiene con 200 iteraciones.

A continuación, se muestran los resultados obtenidos en forma de tablas con las mejores configuraciones, también se muestran las gráficas resultado de las mismas. En la tabla 4 se muestran el mejor, el peor, promedio, varianza y desviación estándar de  $f(x)$  sobre las 20 ejecuciones del algoritmo.

En la tabla 5 se muestran el mejor, el peor, promedio, varianza y desviación estándar del número de grupos sobre las 20 ejecuciones del algoritmo.

En las figuras 8 y 9 se muestra un comparativo de resultados obtenidos del algoritmo con 100, 200 y 500 iteraciones del algoritmo en la instancia de 1027 servicios. En ella se puede observar que los mejores valores de  $f(x)$  se producen

con 500 iteraciones, sin embargo, el resultado de número de grupos implicados es más estable y menor con 200 iteraciones.

Tabla 4 Resumen con los mejores resultados de la similitud.

Instancia-iteraciones	Mejor	Peor	Promedio	Varianza	Desviación
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

Tabla 5 Resumen con los mejores resultados de los Grupos.

Instancia-iteraciones	Mejor	Peor	Promedio	Varianza	Desviación
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

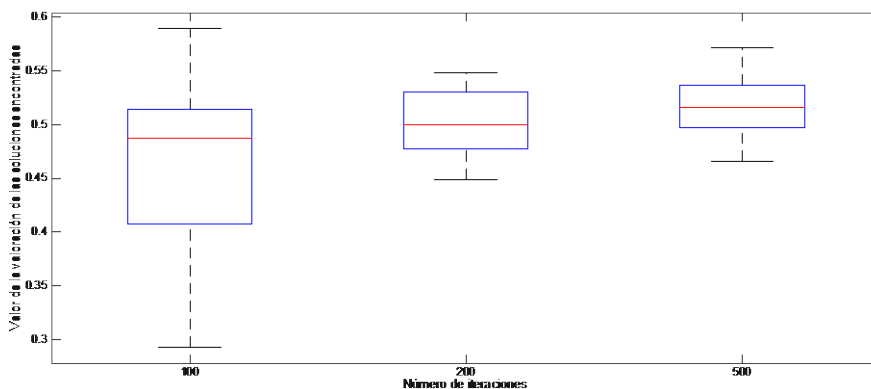


Figura 8 Similitud entre las mejores soluciones encontradas con 1027 servicios.

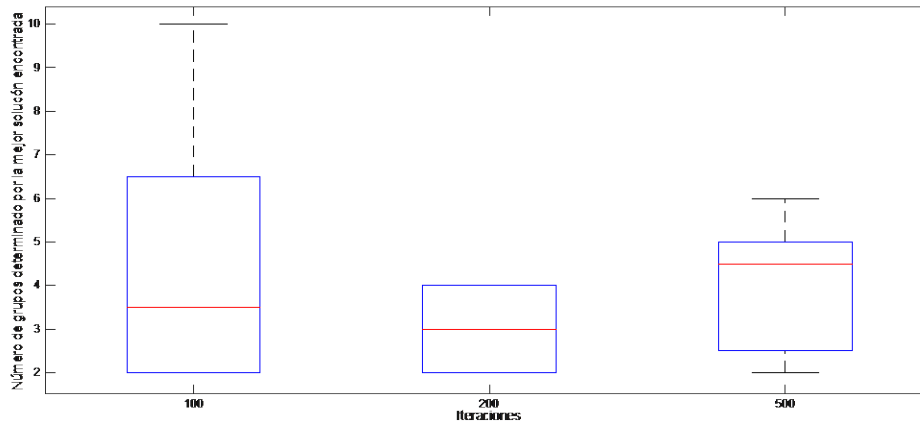


Figura 9 Comparativo del número de grupos en el mejor caso para 1027 servicios.

## 8. Trabajos Relacionados

El tema de agrupamiento y clasificación de servicios Web ha sido estudiado ampliamente desde diferentes perspectivas que abarcan desde enfoques estadísticos, estocásticos y sus variantes. Asimismo, se ha abordado con enfoques novedosos en el ámbito de los servicios web semánticos mediante ontologías y motores de inferencia. Sin embargo, poco se ha hecho para abordar el problema empleando algoritmos híbridos, meta-heurísticas y algoritmos biológicamente inspirados. En esta sección se presenta una revisión de los trabajos relacionados con el tema de clasificación y agrupamiento de servicios Web, así como algunos trabajos que reportan el empleo de algoritmos biológicamente inspirados aplicados en servicios Web.

En 2004 Dong [4] presentó un enfoque de agrupamiento para realizar la búsqueda de servicios Web. Esta búsqueda consistió en dos etapas principales: el usuario proporcionaba palabras clave en un buscador de servicios, con los servicios devueltos por el buscador, el sistema de agrupamiento extraía los conceptos semánticos de las descripciones en lenguaje natural contenidas en las descripciones de los servicios Web. Con ayuda de la co-ocurrencia de términos en los nombres de los parámetros de entrada y salida y en los nombres de las operaciones, el cálculo de similitudes se realizó para emplear el algoritmo de agrupamiento aglomerativo y clasificar los términos en conceptos significativos. Mediante la combinación de las palabras clave originales y los conceptos

extraídos de las descripciones de los servicios la similitud de dos servicios se realiza al nivel de conceptos. Desde el punto de vista del empleo de conceptos, este enfoque es similar al que se describe en este artículo, sin embargo, el algoritmo de agrupamiento que se empleó difiere significativamente.

En 2007 Arbramowicz et. al [5] propuso una arquitectura para el filtrado y agrupamiento de servicios Web. Este enfoque se considera dentro del tema de investigación de los servicios Web semánticos, ya que el autor empleó servicios descritos con el lenguaje OWL-S (Web Ontology Language for Web Services, por sus siglas en inglés). El filtrado de servicios se basa en los perfiles que representan a los usuarios y la información de la aplicación. Para mejorar el filtrado de servicios, se propone el empleo de un método de agrupamiento. El propósito de este enfoque era apoyar el proceso de conciliación entre servicios, ahorrando tiempo de ejecución y mejorando el refinamiento de los datos almacenados.

## **9. Conclusiones**

Las meta-heurísticas representan un aliado poderoso para resolver problemas de optimización en diferentes dominios de aplicación. En este trabajo se presentó una adaptación al algoritmo ABC para obtener un mejor enfoque de agrupamiento de servicios. El algoritmo ABC ha mostrado buenos resultados para el agrupamiento de servicios, más modificaciones son posibles, por ejemplo, la medición de similitudes semánticas, así como incorporar más información de la descripción de los servicios y los tipos de datos. Como trabajo a futuro se implementarán otros algoritmos bio-inspirados para la clasificación y agrupamiento de servicios Web.

## **10. Bibliografía y Referencias**

- [1] D. Floreano, C. Mattiussi, *Bio-Inspired Artificial Intelligence Theories, Methods and Technologies*, The MIT Press Cambridge, Massachusetts London, Englan, 2008.

- [2] B. Xing, J. Wen, "Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms". Springer. Volume 62. 2014.
- [3] S. Binitha, S S. Sathya, "A Survey of Bio inspired Optimization Algorithms". International Journal of Soft Computing and Engineering. Volume 2. Issue 2. May 2012.
- [4] X. Dong, A. Halevy, J. Madhavan, E. Nemes, J. Zhang, "Similarity Search for Web services". In Proceedings of the 30th VLDB Conference. 2004.
- [5] W. Abramowicz, K. Haniewicz, M. Kaczmarek, D. Zyskowski, Architecture for Web services filtering and clustering. In Internet and Web Applications and Services. 2007. Pag. 18.

## **11. Autores**

Dra. Maricela Claudia Bravo Contreras obtuvo el grado de doctor en Ciencias de la Computación en el 2006 por el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET). Desde mayo de 2011 se encuentra trabajando como Profesora Investigadora en el Departamento de Sistemas de la UAM Azcapotzalco. Actualmente es miembro del Área de Investigación en Sistemas de Información Inteligentes ([www.aisii.azc.uam.mx](http://www.aisii.azc.uam.mx)). Sus principales áreas de interés son: composición automatizada y optimización de servicios Web públicos, diseño y desarrollo de Ontologías aplicadas en la solución de problemas de ingeniería; diseño e implementación de aplicaciones de cómputo móvil, cómputo ubicuo y sistemas sensibles al contexto; estudio de sistemas distribuidos adaptativos basados en agentes inteligentes.

Dr. Román Anselmo Gutiérrez obtuvo el grado de Doctor en Investigación de Operaciones en la Universidad Nacional Autónoma de México en 2014. Actualmente es profesor investigador del Departamento de Sistemas de la UAM Azcapotzalco, también es miembro del Área de Investigación de Optimización Combinatoria. Sus principales áreas de interés son: optimización, heurísticas y meta-heurísticas, y sistemas de múltiples agentes.

Ing. Iván Flores Sánchez es graduado de la Licenciatura en Ingeniería en Computación por la Universidad Autónoma Metropolitana.



M.C. Roberto A. Alcántara Ramírez obtuvo el grado de Maestro en Ciencias en Ingeniería Mecatrónica por el T.E.S.E en 2008. Es Ingeniero en Electrónica, por la Universidad Autónoma Metropolitana, unidad Azcapotzalco. Desde 1987 se desempeña como Profesor Investigador de tiempo completo en el Departamento de Electrónica de la UAM-Azcapotzalco. Desde 1998 es miembro del Área de Investigación de Control de Procesos, donde desarrolla trabajos de Investigación en las áreas de Instrumentación, Control de Procesos y Electrónica de Potencia. De 2009 a 2013 fungió como Coordinador de la Licenciatura en Ingeniería Electrónica. Actualmente se desempeña como Jefe del Departamento de Electrónica.