

# **Agrupamiento de servicios web usando el algoritmo de colonia de hormigas**

## ***Maricela Claudia Bravo Contreras***

Universidad Autónoma Metropolitana, Av. San Pablo #180, Col. Reynosa Tamaulipas, C.P. 02200,  
México D.F., Teléfono: 5318 9532 ext. 149  
*mcbc@correo.azc.uam.mx*

## ***Román Anselmo Mora Gutiérrez***

Universidad Autónoma Metropolitana, Av. San Pablo #180, Col. Reynosa Tamaulipas, C.P. 02200,  
México D.F., Teléfono: 5318 9532 ext. 142  
*rmg@correo.azc.uam.mx*

## ***Roberto Alfonso Alcántara Ramírez***

Universidad Autónoma Metropolitana, Av. San Pablo #180, Col. Reynosa Tamaulipas, C.P. 02200,  
México D.F., Teléfono: 5318 9532 9040 ó 9030  
*raar@correo.azc.uam.mx*

## **Resumen**

Existen múltiples servicios Web publicados en servidores o repositorios de servicios Web, dichos repositorios ofrecen mecanismos de búsqueda y selección de los servicios a los solicitantes que tienen necesidades específicas de dichos servicios; sin embargo, los métodos de búsqueda que ofrecen estos repositorios están generalmente basados en coincidencia de textos o palabras clave proporcionadas por los usuarios lo que ocasiona que las solicitudes de servicios devuelvan servicios que no son relevantes con respecto al criterio de la búsqueda. Una forma de mejorar la búsqueda de servicios es mejorando la organización de los servicios mediante su agrupamiento automático basado en los criterios de selección que comúnmente buscan los solicitantes de servicios Web. En este artículo se describe un enfoque innovador para el agrupamiento

de servicios Web basado en la adaptación del algoritmo de Optimización de Colonia de Hormigas (ACO, por sus siglas en inglés). Se describen una serie de experimentos orientados a determinar estadísticamente cuál es la mejor medición de similitud.

**Palabra(s) Clave(s):** agrupamiento de servicios Web, optimización de colonia de hormigas, repositorios de servicios Web.

## **1. Introducción**

En las últimas décadas ha proliferado el uso de servicios Web, esto se debe principalmente a las facilidades que ofrecen para compartir recursos de software reutilizables en ambientes distribuidos independientemente de la plataforma que se esté utilizando. La arquitectura de servicios Web consiste de una serie de protocolos, lenguajes y estándares interoperables basados en el lenguaje XML que son ampliamente utilizados en redes públicas como Internet. Para la descripción de las interfaces de programación de los servicios Web se emplea el lenguaje WSDL<sup>1</sup>, el cual es un estándar de la W3C que permite a los proveedores de servicios publicar sus servicios y a los clientes solicitantes de servicios realizar búsquedas y utilizar la descripción del servicio seleccionado para crear un objeto de llamada remota.

Existen muchas descripciones de servicios (WSDL) publicados en repositorios y en servidores Web, la mayoría de estos repositorios ofrecen mecanismos de búsqueda de servicios basados en palabras clave, los cuales utilizan métodos sintácticos para el descubrimiento de servicios. La principal desventaja de los métodos sintácticos es que en muchas ocasiones devuelven servicios que son irrelevantes para el solicitante. Para resolver este problema existen dos enfoques de solución principales: el enriquecimiento semántico de la descripción de los servicios y el mejoramiento en la organización de los repositorios de servicios Web. En este artículo se describe una propuesta de solución relacionada con el segundo enfoque.

---

<sup>1</sup> <http://www.w3.org/TR/wsdl>

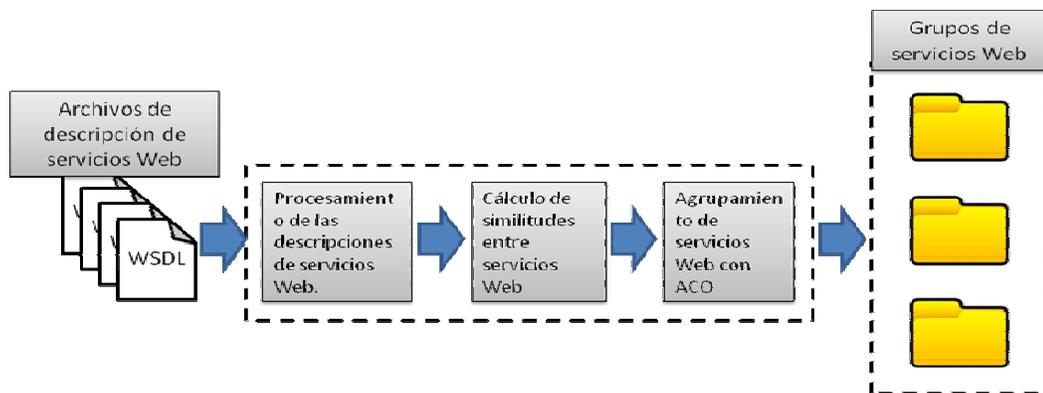
La mejora en la organización de los repositorios de servicios Web tiene como objetivo crear grupos o clases de servicios relacionados a través de sus características comunes, por lo tanto se emplean algoritmos de clasificación o agrupamiento basados en mediciones de similitud. Para el caso de los servicios Web, lo que se busca es encontrar similitudes entre las operaciones y los parámetros de entrada y salida, esto con el fin de agrupar servicios que ofrezcan operaciones con plantillas similares. De esta manera, una vez que los servicios han sido re-organizados o agrupados de acuerdo a sus similitudes se facilitará la búsqueda y selección de servicios. Cuando un solicitante de servicios requiere encontrar alguna descripción de servicio, proporcionará una especificación indicando la operación y los parámetros de entrada y salida que le interesan, con esta información el buscador del repositorio calculará las similitudes del servicio solicitado con la lista de servicios ofrecidos por el repositorio, al encontrar el más cercano devolverá la lista de servicios que se encuentran dentro del grupo al que pertenece. El número de servicios devueltos puede acotarse mediante un valor de referencia que especifique el grado de similitud deseado.

En este artículo se describe un algoritmo de agrupamiento basado en el algoritmo de Optimización de Colonia de Hormigas (ACO, por sus siglas en inglés). El agrupamiento tiene como objetivo crear grupos de servicios Web similares de acuerdo a criterios específicos. El objetivo de tener grupos es ofrecer acceso rápido y eficiente a los servicios [6]. ACO fue propuesto por primera vez por en 1992 by Marco Dorigo et al. [1]. El algoritmo ACO es una meta-heurística que resuelve problemas de optimización, consiste de un conjunto de agentes de software llamados "hormigas" que cooperan para resolver problemas de optimización [7].

El algoritmo ACO toma como entrada una colección de servicios Web y genera como resultado una serie de grupos o clases. El algoritmo ACO fue optimizado para minimizar la diferencia entre servicios dentro de cada grupo y para maximizar la diferencia entre los diferentes grupos.

## 2. Diseño de la solución

El objetivo de este proyecto de investigación es mejorar la organización de los repositorios de servicios Web mediante el agrupamiento automatizado de los servicios, en particular se propone la adaptación de un algoritmo biológicamente inspirado que genere un conjunto de grupos, cada uno conteniendo un conjunto de servicios estrechamente relacionados. Para lograr el agrupamiento se requiere en primera instancia el cálculo de mediciones de similitud entre servicios Web, específicamente se mide la similitud entre las características relevantes descritas en los servicios Web como son: los nombres de las operaciones, los parámetros de entrada y de salida, donde cada uno de éstos consiste de un conjunto de parámetros descritos con su nombre y tipo de dato. El algoritmo biológicamente inspirado que se adaptó fue el ACO, el cual se optimizó buscando maximizar la similitud de servicios al interior del grupo, y maximizar la diferencia entre los diferentes grupos encontrados por el algoritmo. En esta sección se describe la arquitectura de la solución global que fue implementada, la cual consiste de las siguientes tres fases: extracción y procesamiento de los datos de las descripciones de los servicios, cálculo de las mediciones de similitud entre los distintos elementos descritos en todas las descripciones de servicios, y el agrupamiento de los servicios Web. La Figura 1 muestra la arquitectura global de la solución implementada.



**Figura 1. Arquitectura de la solución implementada para el agrupamiento de servicios Web mediante el algoritmo ACO.**

### **3. Procesamiento de las descripciones de servicios Web**

Un servicio Web es la interface de un componente de software reutilizable que es descrito mediante el lenguaje estándar WSDL, es publicado en repositorios de servicios Web o en cualquier servidor Web y es descubrible para su invocación remota. Existen varios repositorios de servicios Web públicos, por ejemplo:

- a) El directorio de Servicios Web SOAP soportado por Membrane<sup>2</sup>, el cual es un registro público de servicios no basado en UDDI, este registro ofrece mecanismos de invocación remota de servicios mediante un cliente SOAP y la verificación constante de la disponibilidad de los servicios.
- b) El repositorio de servicios Web Visual Web Service<sup>3</sup>, el cual permite a los usuarios visualizar la interface de cualquier servicio listado en el repositorio a través de la URL del WSDL.
- c) El repositorio de servicios Web XMethods<sup>4</sup> publica una lista de servicios Web ofrecidos por múltiples proveedores. Este repositorio no ofrece monitoreo en tiempo real de los puntos de acceso a los servicios ni de su funcionalidad. Sin embargo, si verifica que los archivos WSDL de cada servicio se mantengan accesibles y técnicamente viables.
- d) ProgrammableWeb<sup>5</sup> es un directorio de Interfaces de Programación de Aplicaciones (APIs) basados en Internet, el cual realiza búsquedas y descubrimiento de APIs para uso en la Web y aplicaciones móviles.

El primer paso para el procesamiento de las descripciones de servicios Web es descargar descripciones de servicios (archivos WSDL), abrir los archivos y extraer los datos de las operaciones de los servicios y de los parámetros de entrada y salida. Para

---

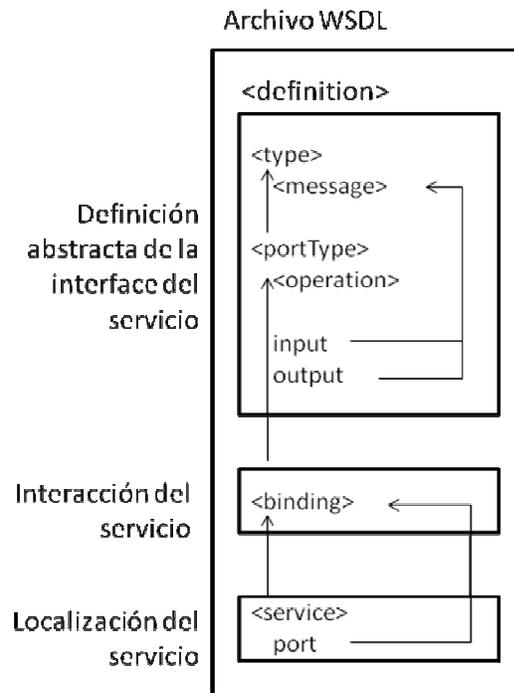
<sup>2</sup> <http://www.service-repository.com/>

<sup>3</sup> <http://www.visualwebservice.com>

<sup>4</sup> <http://www.xmethods.com>

<sup>5</sup> <http://www.programmableweb.com>

extraer la información del servicio el programa debe localizar la etiqueta que corresponde con el nombre de la operación. Específicamente, los archivos WSDL describen la interface de programación de un servicio mediante las siguientes etiquetas: *<definitions>* es la etiqueta principal de toda descripción del servicio, esta etiqueta se divide en una sección abstracta y una sección concreta. La sección abstracta se utiliza para definir los tipos de puertos mediante la etiqueta *<portType>*, la cual define un conjunto de operaciones, los cuales a su vez describen la entrada y la salida de cada operación. Con el propósito de calcular la similitud semántica de todas las operaciones de los servicios se extraen los nombres de las operaciones y los nombres de los mensajes de entrada y salida de cada operación. La Fig. 2 muestra la estructura general de la descripción de servicios Web con WSDL. La Fig. 3 muestra un extracto de descripción del servicio <http://wsf.cdyne.com/WeatherWS/Weather.asmx?WSDL> descargado del repositorio de servicios Visual Web Service.



**Figura 2. Etiquetas principales de la descripción de servicios con WSDL**

```
<wsdl:portType name="WeatherHttpGet">
  <wsdl:operation name="GetWeatherInformation">
    <wsdl:documentation>Gets Information for each WeatherID
  </wsdl:documentation>
  <wsdl:input message="tns:GetWeatherInformationHttpGetIn"/>
  <wsdl:output message="tns:GetWeatherInformationHttpGetOut"/>
</wsdl:operation>
```

**Figura 3. Extracto de la descripción de un servicio con WSDL**

#### **4. Cálculo de las mediciones de similitud entre servicios Web**

Un requerimiento importante del algoritmo de agrupamiento es contar con matrices de similitud entre todos los servicios Web. Dado que cada servicio Web describe un conjunto de operaciones, la comparación se debe realizar entre todas las operaciones de la colección de servicios que se esté empleando para el agrupamiento. Normalmente los nombres de las operaciones son textos entre una y siete palabras de longitud. Estos nombres se escriben usando formatos distintos y en muchas de las ocasiones no contienen información relevante. Por lo tanto para obtener la similitud entre todas las operaciones es necesario realizar una fase de pre-procesamiento. Durante la fase de pre-procesamiento lo que se busca es eliminar o reducir los elementos no significativos de los nombres de las operaciones, para lograrlo se obtienen las unidades léxicas que son parte del nombre. Primero se realiza la normalización del texto transformando los nombres a una forma canónica única, por ejemplo: *getFlightPrice*. Entonces se extraen las unidades léxicas del nombre de la operación, para este caso es [*get*] [*Flight*] [*Price*]. Finalmente, se realiza la discriminación de varias palabras que no contienen significados importantes para los nombres de las operaciones, las cuales son: *http*, *for*, *return*, *result*, *soap*. Todas estas palabras son eliminadas previo al cálculo de similitudes semánticas entre los nombres de todas las operaciones.

Para realizar el cálculo de la similitud semántica entre todos los nombres de operaciones se emplearon cinco mediciones de similitud basadas en Wordnet<sup>6</sup>. Las mediciones de similitud que se emplearon son las siguientes:

- a) La medida de similitud semántica de Wu y Palmer [2] considera las profundidades de dos conjuntos de sinónimos en la taxonomía de WordNet, así como la profundidad del elemento común más bajo en la taxonomía.
- b) La medida de similitud propuesta por Lin [3], se basa en una definición de similitud universal en el ámbito de teoría de la información que no es directamente establecida como en definiciones anteriores, más bien se deriva de un conjunto de supuestos.
- c) La medida de similitud de Path [4] se basa en la longitud de la ruta más corta entre dos conjuntos de sinónimos considerando solamente relaciones del tipo "es-un" y escala la longitud de la ruta por la profundidad total de la taxonomía.
- d) La medida de similitud de Lesk [5] propone que el grado de similitud entre dos palabras es proporcional en la medida de superposición de las definiciones del diccionario.

Estas cuatro mediciones de similitud se seleccionaron porque los resultados de las cuatro están normalizados entre 0 y 1, lo que facilita hacer un análisis comparativo entre sus resultados, así como calcular el promedio de todas.

El resultado de la etapa del cálculo de similitudes es un conjunto de matrices con las distancias de todas las operaciones comparadas por pares. En una etapa posterior, durante la calibración del algoritmo de agrupamiento supervisado, se hace un análisis estadístico para determinar cuál de todas las mediciones devuelve mejores resultados de agrupamiento.

---

<sup>6</sup> WordNet es un diccionario de palabras y sinónimos en inglés. Organiza los conceptos en conjuntos de sinónimos llamados *synsets*, proporcionando definiciones cortas y generales, y almacena relaciones semánticas entre los conjuntos de sinónimos.

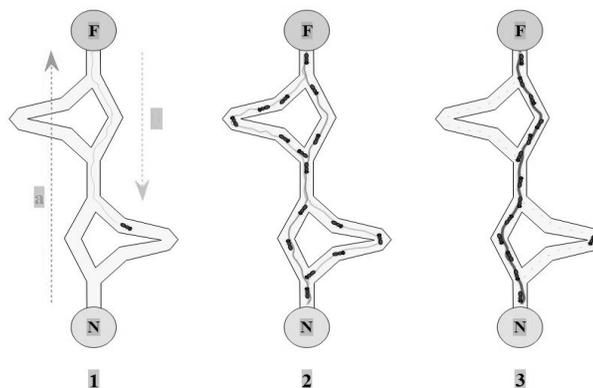
## **5. Agrupamiento de servicios Web mediante ACO**

El algoritmo de agrupamiento que se diseñó e implementó es considerado un híbrido ya que incorpora k-medias y métodos de consenso para determinar el número de grupos (o clases) y que operación pertenece a qué grupo (o clase). En este algoritmo híbrido, el comportamiento de las hormigas es afectado por la presencia y concentración de dos feromonas, las cuales se emplean para definir el número de grupos y los centroides de los grupos.

En este método de agrupamiento híbrido el algoritmo de consenso se incorpora como un elemento de decisión para determinar el número de grupos a generar, el cual se asocia con la primera feromona. Inicialmente, cada hormiga decide construir k grupos, donde k es un número aleatorio. Después la primera feromona es actualizada basada en los resultados que obtengan la mejor solución en cada iteración.

Una variante del algoritmo k-medias se asocia con la segunda feromona, la cual es usada por las hormigas para decidir qué operación debería ser el centroide basada en información previa y las matrices de distancias, cada hormiga decide colocar cada operación en un grupo. La probabilidad de que cada operación sea seleccionada como centroide se actualiza en cada iteración.

Como entrada del algoritmo ACO adaptado se emplean las matrices generadas durante la fase de cálculo de similitudes semánticas. El algoritmo ACO procesa estas matrices iterativamente para encontrar la mejor solución por aproximaciones sucesivas, donde cada hormiga virtual ofrece una solución posible para el agrupamiento. La Fig. 4 muestra el comportamiento general de las hormigas artificiales mientras construyen una solución colectiva.



**Figura 4. Algoritmo ACO que muestra el comportamiento de las hormigas artificiales durante la construcción de una solución.**

El primer paso del algoritmo ACO adaptado es cargar todos los componentes y parámetros de configuración que afectarán el comportamiento de las hormigas artificiales, por ejemplo hay parámetros que permiten ajustar el costo de las trazas de feromonas y la información heurística. El siguiente paso es leer las matrices de similitud para generar un arreglo de distancias medias y la inicialización del conjunto de soluciones, en este caso en particular, el conjunto de solución se integra con todas las hormigas artificiales que ejecutarán el algoritmo.

Cada hormiga artificial construye una solución concurrentemente e independientemente utilizando la información de los parámetros de configuración. Esta solución incorpora dos elementos de información: la varianza y la experiencia adquirida al encontrar la mejor solución. La memoria de la experiencia adquirida es representada mediante matrices. Una matriz de feromonas es utilizada para localizar un servicio dado en un grupo, y una segunda matriz de feromonas es utilizada para correlacionar el número de grupos formados.

Estas matrices representan la memoria de las trazas de feromona producida por las hormigas artificiales durante la construcción de una solución en cada iteración. Cuando todas las hormigas han construido sus soluciones las trazas de feromonas son

actualizadas. Los valores de los rastros de feromonas en las matrices se incrementan por la operación de depósito y se decrementa por la operación de evaporación.

El depósito de feromona se realiza actualizando las matrices que son utilizadas para construir una solución, aumentando la probabilidad de ser seleccionada en el futuro cercano para intensificar la búsqueda en regiones cercanas a la solución. De manera complementaria, la evaporación de feromonas evita la convergencia prematura a soluciones no óptimas, promoviendo la exploración sobre el espacio de búsqueda utilizando información heurística. La información heurística se basa en funciones que minimizan la varianza de la distancia entre grupos y el número de operaciones contenidos en cada grupo.

En cada iteración, se obtiene la mejor hormiga de la colonia, esta hormiga es la que produjo la mejor solución, con la solución de esta hormiga, las matrices de feromonas se actualizan. La mejor hormiga se obtiene ordenando todas las hormigas de acuerdo con su costo y sus soluciones construidas; su contribución es pesada con la actualización del rastro de feromona de acuerdo a su posición [8]. Una vez que este proceso iterativo termina, la mejor solución y la mejor hormiga se devuelve, y sus soluciones son evaluadas.

La Figura 5 muestra el diagrama de flujo del algoritmo ACO adaptado.

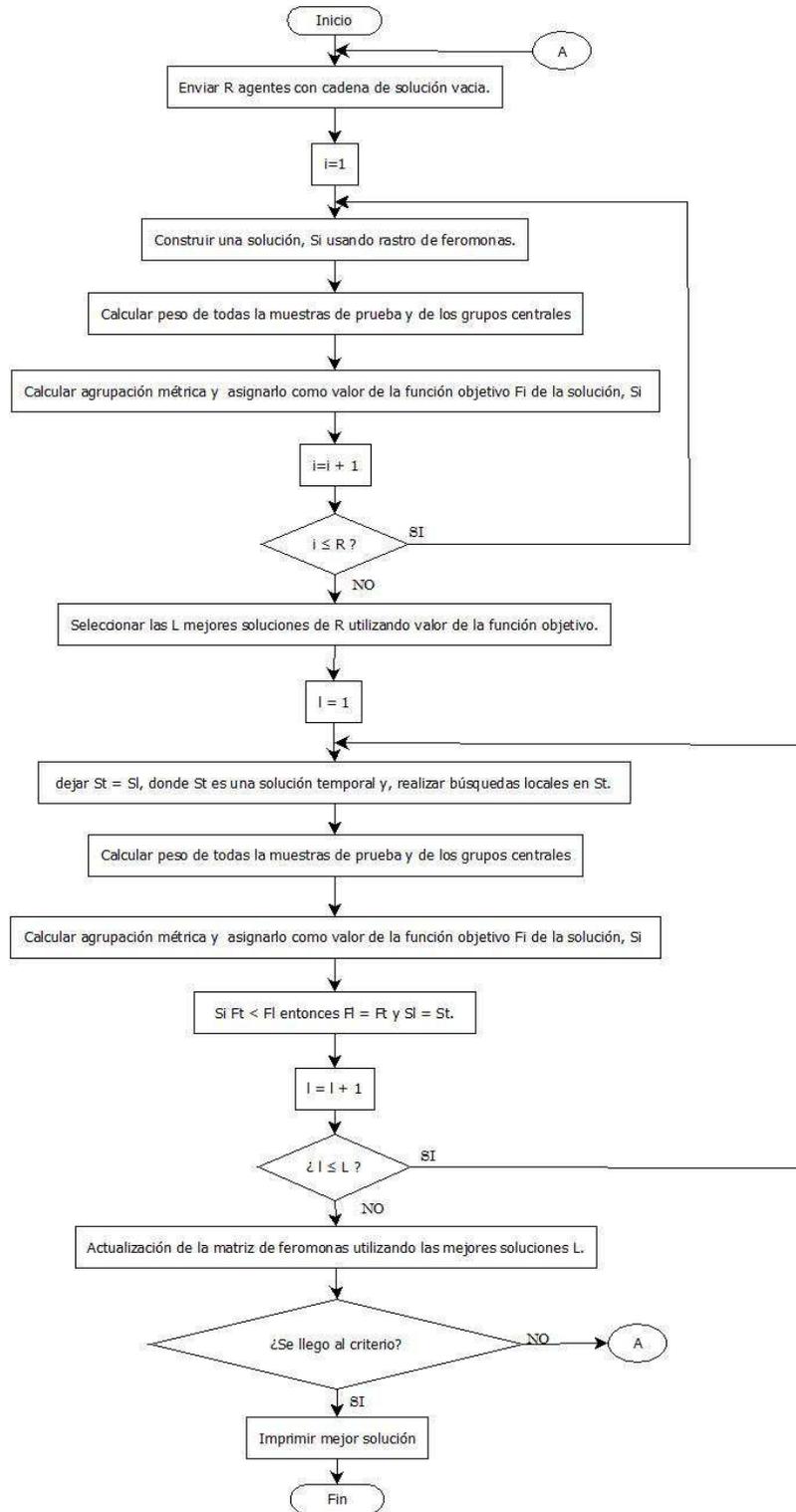


Figura 5. Algoritmo ACO adaptado.

## 6. Experimentación

Para realizar la experimentación con el algoritmo ACO adaptado se utilizó la colección de servicios Web Owls-TC<sup>7</sup>. Los servicios de esta colección se emplean como referente para evaluar algoritmos de clasificación, agrupamiento y conciliación. Esta colección contiene más de 570 descripciones de servicios Web, los cuales están clasificados por humanos en seis dominios de aplicación, de esta forma cualquier algoritmo de agrupamiento o clasificación se puede evaluar comparando los resultados obtenidos con las clasificaciones manuales de la colección. Todos los servicios de esta colección fueron originalmente recuperados de repositorios de servicios Web públicos y de manera semiautomática fueron transformados de WSDL a OWL-S.

El proceso de experimentación comienza ejecutando el programa que lee y procesa cada una de las descripciones de servicios Web de la colección. Por cada archivo de descripción de servicios se extrae de manera automatizada la información relevante de la descripción, específicamente se recuperan los nombres de las operaciones y los parámetros de entrada y salida de cada operación. Con los datos extraídos de todos los servicios se ejecuta el cálculo de las similitudes entre las operaciones, como resultado de estos cálculos se obtienen cuatro matrices de similitud semántica, una por cada una de las medidas de Wu y Palmer [2], Lin [3], Path [4] y Lesk [5], las cuales fueron descritas en la Sección 3. El algoritmo ACO se ejecutó 20 veces con cada una de estas matrices.

La Tabla 1 muestra el número de ejecuciones de cada medición que devolvieron más de seis grupos, es decir, las ejecuciones que se acercaron al resultado deseable de seis clases.

---

<sup>7</sup> <http://projects.semwebcentral.org/projects/owls-tc/>

| <b>Matriz de similitud</b>       | <b>Número de ejecuciones</b> | <b>Ejecuciones que devolvieron más de seis grupos</b> |
|----------------------------------|------------------------------|---|
| Promedio de todas las mediciones | 20                           | 5   |
| Lin                              | 20                           | 0   |
| Lesk                             | 20                           | 4   |
| Wu-Palmer                        | 20                           | 2   |
| Path                             | 20                           | 3   |

**Tabla 1. Resultados de la experimentación con las mediciones de similitud.**

## **7. Evaluación**

El análisis Wilcoxon [9] es una prueba de hipótesis estadística no paramétrica utilizada para comparar mediciones repetidas en una muestra simple para determinar si el promedio de su población difiere. Los datos comparados corresponden al número de grupos producidos por cada medición de similitud semántica. La Tabla 2 muestra los resultados del análisis de Wilcoxon. Esta prueba se utilizó para determinar estadísticamente que todas las mediciones son diferentes.

| Datos              |                    | n  | P                | Cal. W | Crit. W |
|--------------------|--------------------|----|------------------|--------|---------|
| PromSimilarity     | LeskSimilarity     | 17 | P<0.001          | 2.5    | 34      |
| PromSimilarity     | LinSimilarity      | 20 | P < 0.001        | 1.5    | 166     |
| PromSimilarity     | WuPalmerSimilarity | 18 | P > 0.2          | 70.5   | 123     |
| PromSimilarity     | PathSimilarity     | 18 | 0.005 < P < 0.01 | 27     | 130     |
| LeskSimilarity     | LinSimilarity      | 10 | 0.10 < P < 0.20  | 13     | 69      |
| LeskSimilarity     | WuPalmerSimilarity | 18 | P<0.001          | 8.5    | 123     |
| LeskSimilarity     | PathSimilarity     | 19 | P < 0.001        | 0      | 152     |
| LinSimilarity      | WuPalmerSimilarity | 20 | P < 0.001        | 0      | 166     |
| LinSimilarity      | PathSimilarity     | 20 | P < 0.001        | 0      | 166     |
| WuPalmerSimilarity | PathSimilarity     | 17 | 0.02 < P < 0.05  | 29     | 104     |

**Tabla 2. Resultados del análisis de Wilcoxon realizado a todos los pares de mediciones incluyendo a la medición que representa el promedio de todas.**

El resultado del análisis de Wilcoxon muestra que las comparaciones de mediciones son diferentes, ninguna de las mediciones es igual a la otra o devuelve el mismo resultado. Por lo tanto, dado que todas las mediciones son diferentes y considerando que el promedio de todas es diferente a la las demás y que es la que genera en el mayor número de ejecuciones seis grupos (ver Tabla 1), se seleccionó el promedio de todas las mediciones para la agrupación de los 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 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 [10] 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 empleo difiere significativamente.

En 2007 Arbramowicz et. al [11] 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

El análisis de los resultados de Wilcoxon muestra que la matriz de similitud que obtiene las mejores clasificaciones es la que se calcula como el promedio de las cuatro mediciones, esto es debido a que la mayoría de las ejecuciones de esta medición obtuvieron el número correcto de grupos.

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 ACO para obtener un mejor enfoque de agrupamiento de servicios. El algoritmo ACO 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. Referencias

- [1] M. Dorigo, M. Birattari, T. Stutzle, "Ant colony optimization". Computational Intelligence Magazine. Vol. 1. No. 4. 2006. 28-39 pp.
- [2] Z. Wu, M. Palmer, "Verbs semantics and lexical selection". Proceedings of the 32nd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics. 1994. 133-138 pp.
- [3] D. Lin, "An information-theoretic definition of similarity." ICML. Vol. 98. 1998. 296-304 pp.
- [4] C. Leacock, M. Chodorow, "Combining local context and WordNet similarity for word sense identification". WordNet: An electronic lexical database. Vol 49. No. 2. 1998. 265-283 pp.

- [5] S. Banerjee, T. Pedersen, "An adapted Lesk algorithm for word sense disambiguation using WordNet". Computational linguistics and intelligent text processing. Springer Berlin Heidelberg. 2002. 136-145 pp.
- [6] C. Platzner, F. Rosenberg, S. Dustdar, "Web service clustering using multidimensional angles as proximity measures". ACM Transactions on Internet Technology (TOIT). Vol. 9. No. 3. 2009. 11 pp.
- [7] T. Ghafarian, M. Kahani, Semantic web service composition based on ant colony optimization method. In Networked Digital Technologies. 2009. 171-176 pp.
- [8] M. Dorigo, M. Birattari, Ant colony optimization. In Encyclopedia of Machine Learning. 2010. 36-39 pp.
- [9] Concepts and applications of inferential statistics. R. Lowry. 2014.
- [10] X. Dong, A. Halevy, J. Madhavan, E. Nemes and J. Zhang, Similarity Search for Web services. In Proceedings of the 30th VLDB Conference. Toronto, Canada, 2004.
- [11] W. Abramowicz, K. Haniewicz, M. Kaczmarek, D. Zyskowski, Architecture for Web services filtering and clustering. In Internet and Web Applications and Services. 2007. 18-18 pp.

## **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 Jefa 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. Sus principales áreas de interés son: optimización, heurísticas y meta-heurísticas, y sistemas de múltiples agentes.

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.