Minería de datos sobre grafos: un enfoque multiobjetivo aplicado a bioremediación

J. Zaiat[1] and R. Romero-Zaliz[2]

Resumen— **La explotación de catalizadores biológicos (sobre todo microorganismos) para eliminar agentes contaminantes del ambiente, *bioremediación*, requiere la integración de gran cantidad de datos provenientes de diversas fuentes. Reconocer que componentes de una molécula puede reaccionar con algún microorganismo resultaría muy útil para poder atacar un compuesto contaminante hasta subdividirlo en varias submoléculas inocuas. La minería de datos resulta entonces una herramienta muy útil para resolver este problema, sin embargo existen pocos trabajos sobre minería de datos aplicada a bases de datos basadas en grafos, tal como bases de moléculas químicas. Teniendo en cuenta la problemática que esto conlleva se ha desarrollado un algoritmo genético multiobjetivo cuya función es obtener subestructuras interesantes que ayuden a comprender mejor las bases que hacen que un compuesto sea o no nocivo para el medioambiente.**

Palabras clave— **Agrupamiento de datos, algoritmo genético multiobjetivo, bioremediación**

I. Introducción

Algunos microorganismos han adquirido la capacidad de catabolizar compuestos químicos que no forman parte de su metabolismo central cuando éstos se encuentran en su ambiente [1]. Esta capacidad está siendo utilizada en el desarrollo de estrategias cuyo objetivo es la limpieza de compuestos contaminantes en suelos y aguas, conocido como *bioremediación* [2].

La bioremediación ofrece muchas posibilidades interesantes desde el punto de vista bioinformático ya que se ha explorado muy levemente. Esta disciplina requiere la integración de enormes cantidades de datos provenientes de varias fuentes: estructura química y reactividad de compuestos orgánicos; secuencia, estructura y función de las proteínas (enzimas); genómica comparativa; microbiología ambiental; etcétera [3].

Con el fin de poder eliminar todos los posibles agentes contaminantes en un entorno dado, es necesario saber que microorganismo o conjunto de microorganismos son capaces de descomponer las moléculas contaminantes en submoléculas que no resulten dañinas para el ambiente. Reconocer que componentes de una molécula puede reaccionar con algún microorganismo resultaría muy útil para poder atacar un compuesto contaminante hasta subdividirlo en varias submoléculas inocuas.

[1]Dpto. de Computación. FCEyN. Universidad de Buenos Aires. Ciudad Autónoma de Buenos Aires. Argentina. E-mail: jzaiat@dc.uba.ar

[2]Dpto. Ciencias de la Computación e I.A. (DECSAI). Universidad de Granada. Granada. España. E-mail: rocio@decsai.ugr.es

Con esta finalidad hemos desarrollado un algoritmo genético multiobjetivo capaz de realizar un agrupamiento de datos (clustering) sobre moléculas químicas a nivel atómico. De esta manera es posible reconocer subestructuras atómicas, i.e. submoléculas, presentes en varios compuestos contaminantes para luego identificar el o los microorganismos capaces de descomponerlas. Este trabajo forma parte de un proyecto de colaboración con el grupo de investigación del Dr. Alfonso Valencia en el CNB (Centro Nacional de Biotecnología), Madrid, España.

Este trabajo se desarrollará en varias secciones. En la Sección II se explican algunos conceptos introductorios y la motivación que lleva al desarrollo de el presente trabajo. En la Sección III se introduce la metodología utilizada incluyendo la descripción detallada del algoritmo genético multiobjetivo. En la Sección IV se presenta la base de datos utilizada y se realiza la comparación con otras estrategias. Finalmente en la Sección V se presentan las conclusiones y los trabajos a futuro.

II. Motivación

A continuación se introducirán algunos de los conceptos utilizados en el desarrollo de este trabajo.

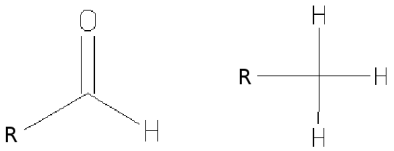
A. Clustering conceptual

El problema de clustering conceptual consiste en, dados un conjunto de descripciones en base a atributos de ciertas entidades, un lenguaje de descripción para caracterizar clases de estas entidades y un criterio de calidad de clasificación; particionar las entidades en clases de tal manera que se maximice el criterio de calidad de clasificación y, simultáneamente, en determinar descripciones generales de estas clases en el lenguaje de descripción dado. Por ello, un método de clustering conceptual busca no sólo una clasificación de las entidades, sino también una descripción simbólica de las clases propuestas. Un aspecto importante que distingue al clustering conceptual es que, a diferencia del análisis de clusters clásico, las propiedades de las descripciones de clases se toman en consideración en el proceso de determinación de estas clases.

En el caso particular del problema de bioremediación, la base de datos consiste en un conjunto de compuestos químicos definidos por sus átomos y los enlaces químicos que los unen. Por lo tanto, el lenguaje utilizado en el clustering conceptual está formado por todos los posibles átomos (e.g., carbono

(C), oxígeno (O), hidrógeno (H)), teniendo en cuenta sus números de oxidación, y sus posibles enlaces covalentes (e.g., simple, doble y triple).

A partir de esta información se pueden extraer subestructuras comunes a varios compuestos tales como las que se pueden observar en la Figura 1.



(a)Carbonilo (b)Metilo

Fig. 1. Ejemplos de posibles subestructuras. Nótese que los vértices sin un átomo explícito corresponden a átomos de carbono. R corresponde a un átomo genérico.

B. Trabajos anteriores

Existen varios trabajos sobre minería de datos sobre grafos, un resumen sobre ellos puede encontrarse en [4]. Uno de los trabajos mas interesantes es el del algoritmo AGM [5], [6]. Este algoritmo es una extensión del algoritmo clásico para extracción de reglas de asociación llamado Apriori [7]. Esta extensión permite su aplicación a base de datos estructuradas en forma de grafos. El algoritmo Apriori, en su versión original, puede extraer la co-ocurrencia de ítems en varios registros de una base de datos de una manera eficiente. Lamentablemente, la estructura de datos que puede manejar se limita a un sistema de representación de datos en forma lineal. Para la extensión a estructuras de datos en forma de grafos se introduce una representación de datos conocida como *matriz de adyacencia*. La regla de la asociación para relacionar un subgrafo con otro se define bajo casi las mismas definiciones de soporte y confianza del algoritmo Apriori. Además, el algoritmo AGM permite derivar todos los subgrafos inducidos más frecuentes, tanto dirigidos como no-dirigidos, conteniendo ciclos y auto-ciclos, con nodos y aristas etiquetados, como sin etiquetar, de una base de datos estructurada.

La metodología utilizada por AGM encuentra todos los subgrafos inducidos de manera exhaustiva, recortando los tiempos de ejecución mediante un valor mínimo de soporte.

Debido a que el algoritmo AGM utiliza una método exhaustivo, necesita de muchas horas y espacio de memoria para proveer al usuario de todos los resultados. Debido a ello es necesario el desarrollo de técnicas heurísticas para acelerar este proceso de búsqueda.

III. Metodología

Para poder resolver el clustering conceptual sobre el problema de bioremediación, se utilizó un algoritmo genético multiobjetivo llamado EMO-CC (Evolutionary MultiObjective Conceptual Clustering) [8]. Este algoritmo ha sido diseñado para poder realizar búsquedas en bases de datos estructuradas y está basado en el algoritmo genético multiobjetivo NSGA-II [9] (ver Figura 2). El objetivo es encontrar subestructuras que optimicen simultáneamente la *especificidad* y *sensibilidad* de la subestructura, siendo estos objetivos contradictorios. Es por ello que se optó por utilizar un algoritmo multiobjetivo de tal manera que se puedan extraer todas aquellas subestructuras no-dominadas, en el sentido que una ninguna otra solución es superior a ellas en todos los objetivos a la vez (i.e., frente óptimo de Pareto [10], [11]). Otro objetivo que es considerado, aunque en forma indirecta, es la diversidad de subestructuras, la cual consiste en mantener un conjunto bien distribuido de soluciones en el frente de Pareto. Para ello, se utiliza una función de no-dominancia localizada. Es decir, una solución es considerada no-dominada si cumple su definición clásica con respecto solamente al conjunto de soluciones vecinas. En esta implementación particular se utiliza el *coeficiente de Jaccard* [12] para el cálculo de vecindad. Entonces dos subestructuras pertenecen a la misma vecindad si comparten al menos un 50% de instancias de la base de datos basándose en este coeficiente.

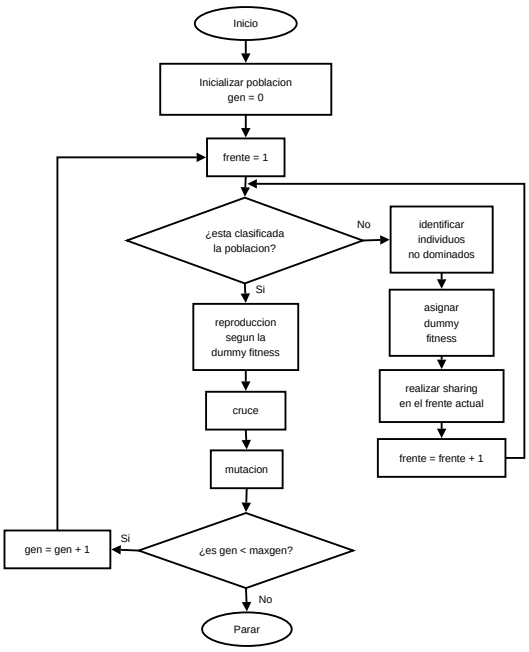


Fig. 2. Esquema general del algoritmo NSGA-II.

Sin embargo, está limitada a trabajar con estructuras que puedan representarse como grafos dirigidos acíclicos. En el caso del problema de bioremediación las estructuras utilizadas contienen ciclos, tales como los anillos aromáticos (ver Figura 3). Para poder utilizar el EMO-CC en este nuevo dominio se realizaron algunos cambios tanto a la representación de los cromosomas como a las funciones de aptitud. A continuación se detallan los principales componentes del algoritmo genético multiobjetivo.

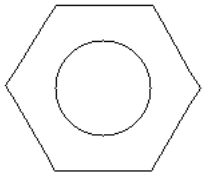


Fig. 3. Anillo aromático. Nótese que los vértices sin un átomo explícito corresponden a átomos de carbono, por lo cual este anillo está formado por 6 átomos de carbono.

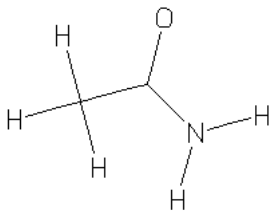
A. Representación de los cromosomas

Las soluciones del AG utilizado son grafos, los cuales representan moléculas químicas. Estas moléculas son moléculas válidas, es decir, se tiene en cuenta los números de oxidación de cada átomo para así no generar más enlaces de los posibles para un átomo dado. A la vez, estas moléculas pueden no ser estables, en el sentido que pueden llegar a tener cargas positivas o negativas. A lo largo de toda la ejecución del AG se mantienen siempre en la población soluciones válidas y se descartan aquellas que no corresponde a una posible molécula química. Una cromosoma válido debe además ser un grafo conexo.

El cromosoma utilizado en la implementación del AG consiste en una lista de átomos y sus enlaces asociados (átomos-enlaces), como se puede ver en la Figura 4. Si bien la representación contiene información redundante (i.e., todos los ejes aparecen dos veces en el mismo cromosoma), esto nos permite realizar cruces más sencillos e intuitivos.

```
C(1)[H(2)-single][H(3)-single][H(4)-single][C(5)-single]
H(2)[C(1)-single]
H(3)[C(1)-single]
H(4)[C(1)-single]
C(5)[C(1)-single][N(6)-single][O(9)-double]
N(6)[H(7)-single][H(8)-single][C(5)-single]
H(7)[N(6)-single]
H(8)[N(6)-single]
O(9)[C(5)-double]
```

(a)Genotipo



(b)Fenotipo

Fig. 4. Ejemplo de un cromosoma. Este cromosoma contiene 9 átomos: 2 carbonos (C) etiquetados como 1 y 5, un nitrógeno (N) etiquetado como 6, un oxígeno (O) etiquetado como 9 y 5 hidrógenos (H) etiquetados como 2, 3, 4, 7 y 8. Para cada uno de estos átomos se registra una lista que contiene los enlaces asociados. Por ejemplo, el primer carbono presenta 4 enlaces covalentes simples con 3 hidrógenos y un carbono.

Otra posibilidad contemplada para la representación del cromosoma consiste en una matriz de adyacencia. Esta representación permite realizar cálculos mucho más rápidamente. Lamentablemente, generar un sistema de cruce en esta representación es más complicado de implementar y poco intuitivo.

No hemos podido desarrollar aún un sistema de representación que presente las ventajas de los dos sistemas mencionados a la vez. Es por ello que tomamos la decisión de utilizar como base la representación átomos-enlaces y, llegado el momento de calcular las funciones de aptitud, se realiza una conversión a una matriz de adyacencia, así aprovechando las ventajas de ambos.

B. Funciones de aptitud

El AG implementado requiere de dos funciones de aptitud: *especificidad* y *sensibilidad*. La primera se calcula como el tamaño del grafo que representa el cromosoma. Este tamaño se calcula simplemente como la cantidad de nodos sumado a la cantidad de ejes, sin importar de que átomos y enlaces se trate. El cromosoma de la Figura 4 tiene una especificidad de 17 (9 átomos + 8 ejes). La sensibilidad se calcula como la cantidad de compuestos de la base de datos que contienen el grafo representado en el cromosoma, independientemente de la cantidad de veces que aparece el grafo en el compuesto.

Estas funciones de aptitud son utilizadas por el algoritmo NSGA-II para seleccionar los mejores padres para la próxima generación. Primero se recuperan aquellos individuos no-dominados hasta llenar el tamaño de la nueva población. En el caso de no existir suficientes, entonces se llena con individuos dominados comenzando con aquellos con menor cantidad de individuos que lo dominan.

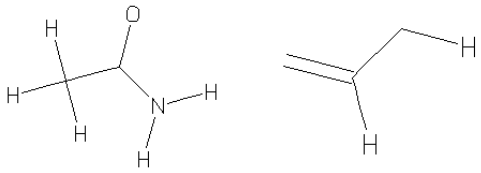
C. Operadores genéticos

Se utiliza un cruce y varias mutaciones como operadores genéticos. Es importante destacar que los operadores utilizados tienen en cuenta la carga de los átomos que forman parte de los grafos utilizados, de tal manera que siempre generan descendientes válidos.

C.1 Cruce

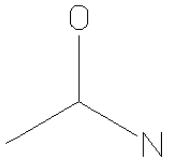
El cruce implementado aprovecha la representación del cromosoma para así generar descendientes que contengan trozos de ambos padres con el fin de obtener mejores soluciones. El proceso de cruce consiste en tomar dos cromosomas, $cromosomaA$ y $cromosomaB$, y extraer de cada uno de ellos un subgrafo, $subgrafoA$ y $subgrafoB$. A partir de estos subgrafos detectar posibles átomos para realizar el cruce, verificando la viabilidad del grafo resultante. Para ello se eligen dos átomos, uno de cada subgrafo, y se eliminan enlaces hasta hacer posible la unión. Se puede ver en la Figura 5 un

ejemplo del operador de cruce comentado.

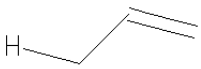


(a)Padre 1

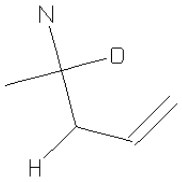
(b)Padre 2



(c)Trozo padre 2



(d)Trozo padre 2

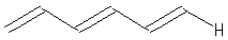


(e)Hijo

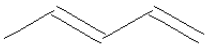
Fig. 5. Ejemplo de operador de cruza. El padre 1 (a) se cruza con el padre 2 (b) generando un hijo (e) con trozos provenientes de cada padre (c-d).

(a)Cromosoma original

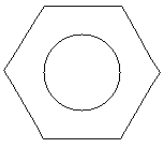
(b)Mutar un átomo



(c)Agregar un átomo



(d)Borrar un átomo



(e)Agregar un enlace

Fig. 6. Ejemplo de los operadores de mutación

C.2 Mutación

En el caso de la mutación, se implementaron cuatro operadores: *mutar un átomo*, *agregar un átomo*, *borrar un átomo* y *agregar un enlace*. Mutar un átomo consiste en modificar un átomo existente por otro compatible, es decir que permita la misma cantidad de enlaces presentes en el cromosoma original (ver Figura 6(b)). Agregar un átomo implica generar un átomo al azar y un enlace asociado e incorporarlo al cromosoma original en algún átomo existente que permita un enlace adicional (ver Figura 6(c)). Borrar un átomo incluye seleccionar un átomo del cromosoma original y eliminarlo conjuntamente con sus enlaces asociados (ver Figura 6(d)). Por último, agregar un enlace consiste en seleccionar dos átomos del cromosoma original que no tengan un enlace y agregárselo. El enlace puede ser simple, doble o triple. Este operador de mutación favorece la creación de ciclos, imprescindible para poder obtener cromosomas que representen anillos aromáticos (ver Figura 6(e)).

Los operadores de mutación se eligen en base a la probabilidad asignada a cada uno. El mejor esquema resulta ser de 0.2 para todos los casos con excepción de *agregar un enlace* para el cual se necesita una probabilidad de 0.4, de esta manera se potencia la aparición de ciclos en las moléculas.

IV. Experimentos y análisis de resultados

Se aplicó EMO-CC a una base de datos de compuestos orgánicos nocivos para el medio-ambiente. Esta base de datos contiene 885 compuestos, de entre los cuales se seleccionaron 50 al azar entre aquellos que solo contienen átomos de carbono, hidrógeno y oxígeno (ver Tabla I). Esta reducción nos permite estudiar mejor el problema haciéndolo más manejable no solo en tiempo de ejecución sino también para estudiar mejor las características particulares del problema. El conjunto de 50 compuestos seleccionados contiene moléculas complejas con 20.38 átomos en promedio. De éstas, 41 contienen al menos un ciclo, 16 al menos dos ciclos anidados, etc.

En la Tabla III se muestran tres soluciones obtenidas por el EMO-CC en su aplicación preliminar al problema. La primera es específica y compleja de obtener ya que contiene ciclos. La segunda muestra una solución muy sensible, dado que todos los compuestos de la base de datos la contienen, sin embargo es bastante poco específica. Por último se puede observar un caso intermedio que resulta una solución de compromiso entre especificidad y sensibilidad. En la Figura 7 se muestra el frente de Pareto de soluciones obtenido. A pesar de que a simple vista parece existen soluciones que se dominan, estas soluciones pertenecen a espacios diferentes teniendo en cuenta las variables y no los objetivos. Los parámetros utilizados en la ejecución del EMO-CC

TABLA I

Base de datos

1-Indanone
cis-Dihydrobenzenediol
cis-1,2-Dihydroxy-1,2-dihydro-8-methylnaphthalene
1-Methylnaphthalene
2,2'-Diphenate
alpha-Pinene
Salicylate
trans,cis-5-Carboxymethyl-2-hydroxymuconate
cis-1,2-Dihydroxy-1,2-dihydro-7-methylnaphthalene
1,6-Dihydroxypyrene
2,3-Dihydroxy-p-cumate
Cyclopropanecarboxylate
trans-9(S),10(S)-Dihydrodiolphenanthrene
Dihydrophloroglucinol
3-(2-Hydroxyphenyl)propionate
cis-2-Oxohept-3-ene-1,7-dioate
4-Hydroxyacetophenone
p-Xylene
4-Methylcyclohexa-3,5-diene-1,2-cis-diol-1-carboxylicacid
Biphenyl
Pyrene-1,2-oxide
trans-9R,10R-Dihydrodiolphenanthrene
4-Oxahomoadamantan-5-one
1,2-Dihydroxynaphthalene
cis-3,4-Dihydroxy-3,4-dihydrophenanthrene
4,4'-Dihydroxy-alpha-methylstilbene
Fumarate
cis-2-Methyl-5-isopropylhexa-2,5-dienal
4-Hydroxybenzoate
4-Phenanthrol
4-Hydroxymethylcatechol
Phenanthrene-1,2-oxide
2-Methylbenzylalcohol
2-Hydroxy-4-carboxymuconatesemialdehyde
3-Isopropylbut-3-enoicacid
EthyleneOxide
cis-4-Carboxymethylenebut-2-en-4-olide
o-Methylbenzoate
2-Hydroxy-5-methyl-cis,cis-muconicsemialdehyde
2-Hydroxypyrene
2-Hydroxy-2-methyl-1,3-dicarbonate
trans-2'-Carboxybenzalpyruvate
3-Methylsalicylaldehyde
cis-1,6-Dihydroxy-2,4-cyclohexadiene-1-carboxylicacid
2-Hydroxybiphenyl
(S)-Mandelate
2-Propanol
1-Formyl-2-indanone
2-Hydroxyacetophenone
2,3-Dihydroxyethylbenzene

se detallan en la Tabla II.

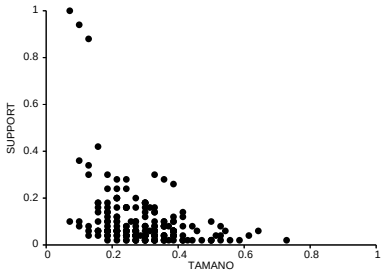


Fig. 7. Pareto obtenido. A pesar de que a simple vista parece existen soluciones que se dominan, estas soluciones pertenecen a espacios diferentes teniendo en cuenta las variables y no los objetivos.

Para realizar una comparación con el EMO-CC se aplicó la base de compuestos químicos tanto al AGM como a otro algoritmo heurístico que realiza minería de datos sobre grafos, conocido como SUBDUE [13], [14]. Este ultimo algoritmo también ha sido diseñado para la búsqueda de patrones en bases de datos estructuradas. Se basa en una estrategia de recorte de caminos (beam-search [15]) para realizar la búsqueda. Los resultados obtenidos resultan interesantes, pero éstos se encuentran concentrados en una región en donde se observan más repeticiones como puede verse en la Figura 9. Es

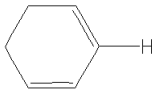
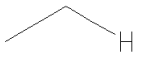
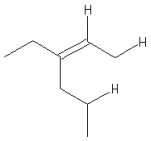
TABLA II

Parámetros de AG

Parámetro	Valor
Tamaño de la población	50
Número de evaluaciones	50000
Probabilidad de cruce	0.8
Probabilidad de mutación	0.2

TABLA III

Ejemplos de estructuras obtenidas

Estructura	Especificidad	Sensibilidad
	0.585714	0.06
	0.0714286	1
	0.242857	0.18

decir, se obtienen resultados redundantes que sólo describen a un pequeño subconjunto de la base de datos, a diferencia de los obtenidos por el EMO-CC que cubren todos los compuestos. Algunos de los resultados obtenidos por SUBDUE pueden verse en la Figura 8. Esta diferencia se produce debido a que el algoritmo SUBDUE no es capaz de reveer las soluciones obtenidas en cada iteración, es decir, debido a su estrategia golosa pierde soluciones interesantes al tomar posible decisiones equivocadas. En cambio el EMO-CC es capaz de ir evolucionando las soluciones obtenidas debido a su implementación sobre un algoritmo evolutivo. Por otro lado, debido a la naturaleza exhaustiva del AGM, éste no ha sido capaz de finalizar su ejecución al pedirle un soporte mínimo de al menos una instancia, debido a problemas de memoria. Debido a ello ha sido imposible realizar una comparación.

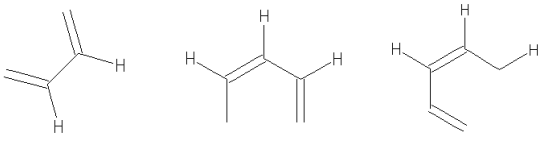
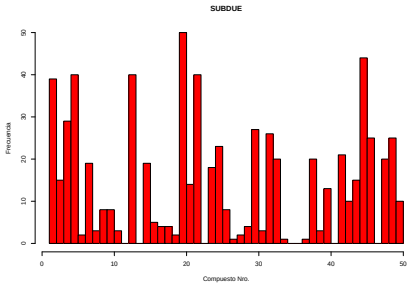
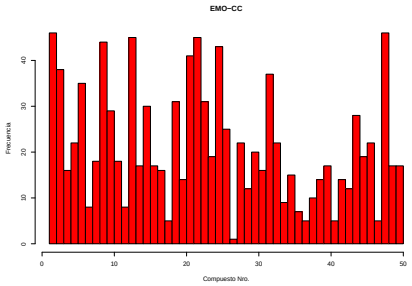


Fig. 8. Algunos resultados obtenidos con SUBDUE



(a)SUBDUE



(b)EMO-CC

Fig. 9. Histogramas. Se grafica la frecuencia de resultados obtenidos por compuesto.

V. Conclusiones y trabajos a futuro

La búsqueda de submoléculas que se encuentren conservadas en varios compuestos químicos nocivos resulta un problema interesante, no sólo desde el punto de vista de la bioremediación, sino también desde el punto de vista de las técnicas metaheurísticas. El espacio de búsqueda resulta ser lo suficientemente extenso para no poder aplicar una estrategia exhaustiva en un tiempo de ejecución razonable. La necesidad de realizar un agrupamiento de las moléculas, representadas en forma de grafos, hace aún más compleja la búsqueda ya que requiere de un algoritmo desarrollado específicamente para ese fin. Además hay que tener en cuenta que no todo grafo que incluya átomos y enlaces válidos resulta en una submolécula válida desde el punto de vista químico. Teniendo en cuenta todos estos problemas se ha desarrollado un algoritmo genético multiobjetivo cuya función es obtener subestructuras interesantes desde el punto de vista de su especificidad y sensibilidad simultáneamente.

Los resultados preliminares obtenidos hasta el momento son prometedores y logran mejorar aquellos obtenidos mediante otras técnicas.

Debido a la complejidad del espacio de búsqueda, aplicar EMO-CC sobre la base de datos completa requiere aún de un tiempo de ejecución considerable. Una posibilidad para reducir estos tiempos y conseguir resultados interesantes consistiría en realizar un preprocesamiento del repositorio original extrayendo pequeños grupos funcionales. A partir de los cuales se podría compactar la base de datos original reduciendo así el espacio de búsqueda sin eliminar subestructuras interesantes. Adicionalemnte, las soluciones obtenidas mediante el EMO-CC permiten generar nuevos atributos, más complejos y relevantes, para describir los compuestos de una base de datos.

Paralelamente a la aplicación sobre el domino químico, se esta trabajando en otros dominios con características similares, tales como las redes de regulación genética, obteniendo buenos resultados.

Agradecimientos

Agradecemos la colaboración del grupo de investigación del Dr. Alfonso Valencia al facilitarnos la base de datos de compuestos químicos, y en particular a Almudena Trigo por su ayuda para comprender el problema en estudio.

Referencias

[1] R.E. Parales, N.C. Bruce, A. Schmid, and L.P. Wackett, "Biodegradation, biotransformation, and biocatalysis (b3)," *Appl. Environ. Microbiol.*, vol. 68, no. 10, pp. 4699–4709, Oct 2002.

[2] M. Dua, A. Singh, N. Sethunathan, and A.K. Johri, "Biotechnology and bioremediation: successes and limitations," *Appl. Microbiol. Biotechnol.*, vol. 59, pp. 143–152, 2002.

[3] F. Pazos, D. Guijas, A. Valencia, and V. De Lorenzo, "Metarouter: bioinformatics for bioremediation," *Nucleic Acids Research*, vol. 33, pp. D588–D592, 2005, Database Issue.

[4] T. Washio and H. Motoda, "State of the art of graph-based data mining," *ACM, SIGKDD Explorations*, vol. 5, no. 1, pp. 59–68, 2003.

[5] A. Inokuchi, T. Washio, and H. Motoda, "An apriori-based algorithm for mining frequent substructures from graph data," in *Principles of Data Mining and Knowledge Discovery*, 2000, pp. 13–23.

[6] A. Inokuchi, T. Washio, and H. Motoda, "Complete mining of frequent patterns from graphs: Mining graph data," *Machine Learning*, vol. 50, no. 3, pp. 321–354, March 2003.

[7] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, P. Buneman and S. Jajodia, Eds., Washington, D.C., 1993, pp. 207–216.

[8] R. Romero-Zaliz, C. Rubio-Escudero, O. Cordón, O. Harari, C. del Val, and I. Zwir, "Mining structural databases: An evolutionary multi-objetive conceptual clustering methodology," in *Applications of Evolutionary Computing: EvoWorkshops 2006: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoINTERACTION, EvoMUSART, and EvoSTOC, Budapest, Hungary, April 10-12, 2006. Proceedings*, Franz Rothlauf et al, Ed., vol. 3907, chapter EvoBIO Contributions, pp. 159 – 171. Springer Berlin / Heidelberg, 2006.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, 2002.

[10] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc., 2001.

[11] E. Ruspini and I. Zwir, "Automated generation of qualitative representations of complex object by hybrid soft-computing methods," in *Pattern Recognition: From Classical to Modern Approaches*, S. Pal and A. Pal, Eds., Singapore, 2001, pp. 453–474, World Scientific Company.

[12] P. Jaccard, "The distribution of flora in the alpine zone," *The New Phytologist*, vol. 11, no. 2, pp. 37–50, 1912.

[13] I. Jonyer, D. J. Cook, and L. B. Holder, "Discovery and evaluation of graph-based hierarchical conceptual clusters," *Journal of Machine Learning Research*, vol. 2, pp. 19–43, 2001.

[14] D. Cook, L. Holder, S. Su, R. Maglothin, and I. Jonyer, "Structural mining of molecular biology data," *IEEE Engineering in Medicine and Biology, special issue on Advances in Genomics*, vol. 4, no. 20, pp. 67–74, 2001.

[15] D. Poole, A. Mackworth, and R. Goebel, *Computational Intelligence: A Logical Approach*, Oxford University Press, USA, 1998.