



Revista de Matemática: Teoría y
Aplicaciones

ISSN: 1409-2433

mta.cimpa@ucr.ac.cr

Universidad de Costa Rica
Costa Rica

BERNÁBE L., MARIA BEATRIZ; RODRÍGUEZ F., MARCO ANTONIO; GONZÁLEZ V.,
ROGELIO; ESTRADA A., MARTÍN

UNA PROPUESTA BIOINSPIRADA BASADA EN VECINDADES PARA
PARTICIONAMIENTO

Revista de Matemática: Teoría y Aplicaciones, vol. 23, núm. 1, enero, 2016, pp. 221-239
Universidad de Costa Rica
San José, Costa Rica

Disponible en: <http://www.redalyc.org/articulo.oa?id=45343487010>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

UNA PROPUESTA BIOINSPIRADA BASADA EN VECINDADES PARA PARTICIONAMIENTO

A BIOINSPIRED PROPOSAL BASED ON NEIGHBORHOODS FOR PARTITIONING

MARIA BEATRIZ BERNÁBE L.*

MARCO ANTONIO RODRÍGUEZ F.[†] ROGELIO GONZÁLEZ V.[‡]

MARTÍN ESTRADA A.[§]

Received: 20 Mar 2014; Revised: 27 Aug 2015;

Accepted: 1 Sep 2015

*Facultad de Ciencias de la Computación. Benemérita Universidad Autónoma de Puebla, Puebla, México. E-Mail: beatriz.bernabe@gmail.com

[†]Misma dirección que/Same address as: M.B. Bernábe. E-Mail: marco89_rf@hotmail.com

[‡]Misma dirección que/Same address as: M.B. Bernábe. E-Mail: rogelio.gzzvzz@gmail.com

[§]Misma dirección que/Same address as: M.B. Bernábe. E-Mail: estradajose660@gmail.com

Resumen

Una de las principales fuentes de inspiración para proponer nuevos paradigmas computacionales ha sido la observación de la naturaleza. Diversas técnicas en inteligencia artificial han surgido de esta manera.

Uno de los esfuerzos que ha causado gran impacto es imitar la manera en que sobreviven otros seres vivos, y en particular, el estudio del funcionamiento cerebral es útil para proponer esquemas análogos y dar solución a algunos problemas. En este punto, los sistemas bioinspirados han surgido como un conjunto de modelos que están basados en el comportamiento y la forma de actuar de ciertos sistemas biológicos, los cuales pueden verse en áreas como la minería de datos e investigación de operaciones donde se distingue el agrupamiento de datos. A partir de la necesidad de resolver problemas de agrupamiento, hemos propuesto un algoritmo de particionamiento bioinspirado de búsqueda por vecindades. Este agrupamiento en una connotación bioinspirada, ha sido planteado después de observar algunas características comunes entre el particionamiento y la conducta del ser humano, donde dichas características pueden ser modeladas. Debido a la alta complejidad del particionamiento hemos incorporado la búsqueda por entorno variable (VNS) en el algoritmo de agrupamiento bioinspirado. La elección de esta metaheurística obedece a la semejanza que hay entre VNS y el modo en que los seres vivos se organizan para resolver situaciones de conflicto.

Palabras clave: conglomerado geográfico; bioinspirado; seres vivos; búsqueda por vecindades.

Abstract

One of the main sources of inspiration to propose new computational paradigms has been the observation of nature. Diverse artificial intelligence techniques have been created in this way.

One of the efforts that has caused great impact is imitating the way some living beings survive, and in particular, the study of the study of brain function is useful to propose analogous schemes and solve some problems. In this point, the bioinspired systems have been originated as a set of models based on the behavior of certain biological systems, which can be seen in areas such as data mining and operations research where data clustering stands out. From the need of solving clustering problems, we have proposed a bioinspired neighborhood search partitioning algorithm. This algorithm, under a bioinspired connotation, has been proposed after observing some of the characteristics in common between clustering and human behavior, where said characteristics can be modeled. Given the high complexity of data clustering, we have incorporated variable neighborhood search (VNS) into the bioinspired clustering algorithm. We chose this metaheuristic because of the similarity that exists between VNS and the way that living beings get organized to solve conflict situations.

Keywords: geographical clustering; bioinspired; living beings; neighborhood search.

Mathematics Subject Classification: 62H30, 91C20, 90C27, 90C59.

1 Introducción

El proceso de la naturaleza, ha innovado, creado, validado, mejorado y diversificado los sistemas de los seres vivos donde el hombre se ha caracterizado siempre por una búsqueda constante de nuevas y diferentes formas para mejorar sus condiciones de vida. Una motivación interesante es imitar la manera en que sobreviven otros seres vivos, sin embargo también el estudio del funcionamiento cerebral es útil para proponer esquemas de analogías y dar solución a algunos problemas. En este punto, los sistemas bioinspirados han surgido como un conjunto de modelos que están basados en el comportamiento y la forma de actuar de sistemas biológicos en áreas de la inteligencia artificial, la minería de datos y la investigación de operaciones con múltiples aplicaciones en el campo de la bioinformática, que obedece a la gran y diversa expansión de la cantidad de datos producidos por la biología u otros tipos de datos. Esta situación ha generado una necesidad de algoritmos precisos de predicción y clasificación. La precisión de los algoritmos de clasificación puede verse afectada por una diversidad de factores, algunos de ellos considerados genéricos en cualquier algoritmo de aprendizaje automático y, por tanto, aplicables a los distintos campos de investigación. Son estos factores genéricos los que han recibido atención por la comunidad de aprendizaje automático y reconocimiento de patrones durante un gran número de años. Considerar diversos algoritmos bioinspirados para iniciar una investigación de este carácter, es una tarea necesaria. Sin embargo un caso obligado tanto de estudio y análisis como de aplicación dentro de la optimización combinatoria, es la optimización por Colonias de Hormigas. Las hormigas, a la hora de buscar comida se dispersan, en un primer instante, de una forma totalmente aleatoria. Cuando una hormiga o varias hormigas alcanzan una fuente de alimento, el resto de la colonia es alertada a través de la liberación de feromonas, de esta manera el patrón de búsqueda de las hormigas cambia de aleatoria a sistemático formando rutas de flujo desde y hacia las fuentes de alimento. Estas rutas son eficientes ya que dependiendo de la concentración de feromonas el flujo de hormigas es diferente; a mayor feromonas más hormigas, las cuales a su vez siguen el camino más corto, que en diversos problemas de optimización como el particionamiento, se expresa como función objetivo de minimización de distancias. En este sentido, se pueden encontrar en la literatura computacional algoritmos no bioinspirados bien definidos y que son

útiles para solucionar el problema del camino más corto, pero muy costosos en tiempo computacional. Sin embargo, la diferencia entre estos algoritmos y el de la colonia de hormigas es que los primeros alcanzan la solución mediante una búsqueda en serie (mirando un determinado camino hasta que no se puede avanzar) mientras que el algoritmo bioinspirado lo hace de una forma paralela (las hormigas toman varios caminos hasta que obtienen la convergencia, es decir, el camino final por el que camina toda la colonia). Una aplicación muy útil de este algoritmo es en la clasificación de objetos. En este trabajo, a diferencia de la búsqueda “paralela” de las hormigas, nuestra búsqueda es por vecindades que en un contexto bioinspirado, induce a la siguiente conclusión: El particionamiento clásico en presencia de restricciones de búsqueda por vecindades, es posible presentarlo como un algoritmo bioinspirado en analogía con el comportamiento de “reunión” de seres humanos en situaciones críticas o emergentes donde estos seres deben agruparse. Por otra parte, sabemos que los algoritmos de agrupamiento son herramientas computacionales que optimizan uno o más criterios usando ejemplos o experiencias pasadas. En este escenario, el método de aproximación para particionamiento que hemos construido se basa en búsqueda por vecindades como habitualmente lo hacen los humanos en condiciones de emergencia. Nuestras conclusiones conducen a proponer al particionamiento como un algoritmo de trayectoria-bioinspirado que promete soluciones cuasi-óptimas cuando la técnica de optimización se basa en búsquedas por vecindades y resuelve una gran cantidad de problemas de agrupamiento para datos de tipo espacial o biológico siempre que estén descritos espacialmente. El presente trabajo está organizado como sigue: la introducción como sección I. En la sección II se expone la bibliografía revisada. La sección III explica el particionamiento, y la sección IV muestra el método de búsqueda por vecindades. Por último, la sección V presenta la búsqueda por vecindades bioinspirada. Las conclusiones son presentadas en la sección VI.

2 Revisión de la bibliografía: bioinspirado

La teoría de la evolución es considerada actualmente como la combinación de las posturas de Darwin y se le conoce como el paradigma Neo-Darwiniano. Este paradigma sostiene que los procesos de vida del planeta se pueden explicar mediante unos pocos procesos estadísticos que actúan en las poblaciones y especies. Estos procesos son: reproducción, mutación, competencia y selección. Al resultado de aplicar estas operaciones se le conoce como evolución donde el concepto importante en la evolución es la selección natural, la cual constituye un método adaptativo en los componentes de una población ante cambios en las

condiciones del entorno. Aquí es importante subrayar que las operaciones de selección y competencia en la evolución, al entenderse como un proceso adaptativo, se asume un mecanismo implícito de organización y orden (agrupamiento) frente a distintas situaciones emergentes o de conflicto de cualquier nivel. Este aspecto es importante en el presente trabajo, el cual recogemos y extendemos en las próximas secciones.

La evolución artificial intenta emular en un equipo de cómputo el mismo procedimiento que ha tenido lugar a lo largo de millones de años en el mundo natural. Para ejecutar esta emulación, la evolución artificial genera y simula los individuos de una población, los cuales se someten a determinadas operaciones, como lo son: selección, cruce y mutación. Los diversos enfoques de la evolución artificial son utilizados para resolver principalmente problemas de optimización, ya que sin necesidad de conocimiento alguno del problema, logran encontrar una solución satisfactoria. En la figura 1 se muestra el diagrama de flujo del proceso de evolución artificial. Al conjunto de estos enfoques se les conoce como algoritmos bioinspirados, ya que como su nombre lo dice, basan su metodología en la naturaleza, evolución, selección natural, generación de nuevos individuos, intercambio de información entre individuos, etc. [4].

Los primeros intentos para implementar evolución artificial para resolver problemas de optimización se sitúan en los años 30s. Desde entonces, diversas versiones de algoritmos han sido propuestas por diversos investigadores. En 1954 Nils Barricelli desarrolla las primeras simulaciones de procesos de evolución y las utiliza sobre problemas de optimización [1]. En 1956 George J. Friedman propone una aplicación de técnicas evolutivas a la robótica. Es un proceso similar a la selección natural, propuso evolucionar circuitos de control similares a lo que hoy se conoce como redes neuronales [8]. En 1958, Hans Joachim Bremermann [3], consideró la evolución como un proceso de optimización y usó cadenas binarias que se combinaban por medio de operadores de reproducción, selección y mutación. En los años 60s tres grupos marcaron el nacimiento de los AEs. Primero en la Technische Universität de Berlin un grupo formado por Ingo Rechenberg y Schwefel formularon en 1965 ideas sobre cómo un proceso evolutivo puede usarse para resolver problemas numéricos de optimización complejos [16]. Como segundo grupo, en 1966 Lawrence J. Fogel visualizó la evolución como un medio para alcanzar las metas de la Inteligencia Artificial, por lo que evolucionó agentes inteligentes representados como máquinas de estado finito. A este paradigma se le conoce como Programación Evolutiva [6]. Por último, en 1975 John Henry Holland vio a un proceso evolutivo como la clave en el diseño e implementación de sistemas adaptativos robustos, los cuales pueden desempeñarse en un ambiente cambiante e incierto, con capacidad de autoadaptación.

Estas ideas originaron los llamados planes reproductivos, mejor conocidos como el paradigma de los Algoritmos Genéticos [9]. En los 70s se avanzó en los estudios empíricos y en la teoría con respecto a los paradigmas mencionados, mejorando el desempeño y su aplicabilidad. Durante los años 80s, se amplió la diversidad de aplicaciones y se generaron variantes de estos paradigmas. En los 90s, surgieron los primeros congresos, lo que permitió el trabajo de colaboración y diferenciación. En 1992, en su tesis doctoral, Marco Dorigo propuso una técnica probabilista para resolver problemas computacionales relacionados con la búsqueda de caminos en grafos llamada Ant Colony Optimization (ACO), basada en el comportamiento de una colonia de hormigas buscando caminos entre el nido y la fuente de comida más cercana [5]. En 1995 Kennedy y Eberhart propusieron la Optimización Mediante Cúmulos de Partículas o Particle Swarm Optimization (PSO), el cual es un algoritmo de búsqueda basado en la simulación del comportamiento social del vuelo de las aves dentro de una parvada [11]. Este tipo de algoritmos (PSO y ACO) se les llama de inteligencia colectiva ya que entre los individuos intercambian información y colaboran para un fin en común. En el año 1997, Rainer Storn y Kenneth Price presentan el algoritmo de Evolución Diferencial (ED), el cual se basa en generar nuevos individuos a partir de la población actual utilizando una fórmula simple basada en la diferencia entre vectores [15]. Cambiando de década, en el 2005 D. Karaboga propuso el algoritmo Artificial Bee Colony (ABC), el cual se basa en el proceso de búsqueda de néctar que realizan las abejas para el proceso de optimización [10]. Una de las diferencias fundamentales entre la evolución natural y la evolución artificial, es que en la natural no persigue ningún objetivo en específico, simplemente las especies se van modificando como consecuencia de las fuerzas y fenómenos de la naturaleza que actúan sobre tales especies. La evolución artificial tiene como objetivo producir, en el proceso de evolución, un individuo capaz de resolver una tarea o proceso, medido por su calidad.

2.1 Paradigmas de los sistemas bioinspirados

Los algoritmos bioinspirados emulan, de manera aproximada, el comportamiento de un sistema natural para el diseño de métodos heurísticos para la resolución de problemas. Las características que presentan son las siguientes [7]: Son estocásticos o no determinísticos, esto es, tienen un componente aleatorio en la toma de decisiones. A menudo presentan, de forma implícita, una estructura paralela (múltiples agentes). Son adaptativos, utilizando retroalimentación con el entorno para modificar el modelo y los parámetros [13].

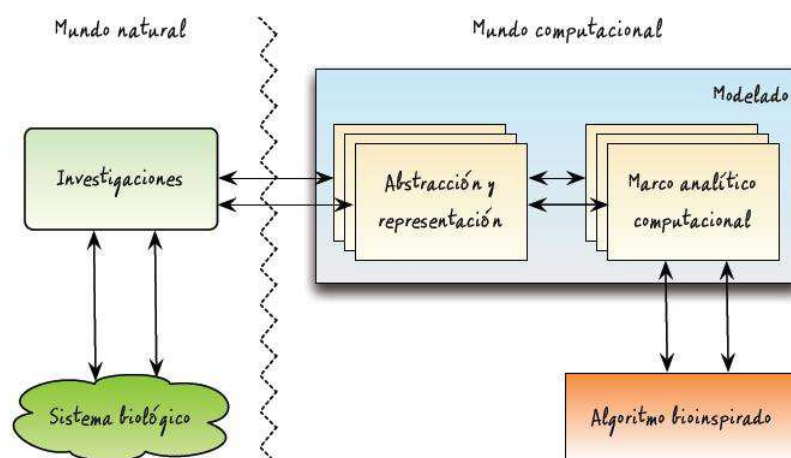


Figura 1: Conexión entre los algoritmos bioinspirados y la biología (adaptado de [17]).

El paso de un modelo biológico a un algoritmo bioinspirado implica, al menos, que concurren las áreas de biología, matemáticas y ciencias de la computación. En [17] los autores proponen un marco conceptual para el desarrollo de modelos computacionales bioinspirados sofisticados, exponiendo la relación existente entre la biología y los algoritmos bioinspirados. Una aproximación a dicho proceso se muestra en la figura 1, donde la parte izquierda se refiere a la investigación en el campo de la biología, mientras que la derecha muestra el desarrollo de algoritmos bioinspirados.

3 Particionamiento

Conocido el problema de optimización a resolver, la tarea inicial consiste en determinar el conjunto de componentes que se pueden emplear para formar una solución y por otro lado debe ser definido el modelo a utilizar, el cual guiará la generación de soluciones a partir del conjunto de componentes disponibles. Una vez inicializados los valores de control, que normalmente se asocian a los componentes de las soluciones o bien a las conexiones entre ellos, se generan las perturbaciones y soluciones vecinas se modifican para guiar a las siguientes soluciones hacia soluciones de alta calidad. Este proceso se distingue en la construcción de nuestro algoritmo. Estamos proponiendo una analogía de comportamiento de los seres vivos en momentos de desorden en emergencias contra las características del particionamiento con búsqueda por vecindades. A modo coloquial el particionamiento inicia con una solución inicial aleatoria del problema a resolver, k centroides representan los grupos, y en combinación con búsqueda

por vecindades, se genera una solución vecina en una estructura de vecindad. Si la solución es mejor, seguirá buscando, con una pequeña perturbación con búsqueda local (cambio de algún centroide), otra solución en la misma estructura de vecindad hasta que la solución empeore. La pequeña perturbación en la búsqueda local emula el comportamiento social humano: Cuando las personas (objetos) atraviesan por situaciones de emergencia como los desastres naturales, usualmente, forman grupos y un líder (centroide) es propuesto para dirigir el grupo. Si una persona no se siente identificada con el resto del grupo o con el líder, esta abandonará el grupo para seguir a otro líder (movimiento de un objeto a otro grupo), por lo regular alguien conocido o cercano (similitud). En este mismo modo la búsqueda local selecciona el grupo con el peor costo de similitud (el grupo menos cohesivo) y el objeto más alejado (disímil) será elegido para buscar un nuevo grupo basado en el objeto más cercano al mismo pero que pertenece a un grupo diferente. El valor del número de iteraciones para búsqueda local determinará cuánto durará esta revuelta (personas abandonando a sus grupos), y así, cuando la similitud de los grupos ya no pueda ser mejorada entonces una revuelta más grande ocurrirá, la estructura de vecindad será alterada en gran manera. Durante el caos, los humanos buscarán mejores sociedades o grupos, al principio podrán cambiar su membresía de un grupo a otro pero cuando la situación no mejora en lo general la única opción que les queda es desbaratar los grupos actuales y seleccionar líderes completamente nuevos para formar grupos con mejor cohesión y con un objetivo en común a lograr. De esta manera, el cambio en las estructuras de vecindad emula el comportamiento humano al reemplazar todos los centroides actuales por nuevos y así formando nuevos grupos con nuevos miembros con la esperanza de encontrar una configuración mejor. Después de esta gran perturbación, una nueva búsqueda local es iniciada y el proceso continúa hasta que el máximo número global de iteraciones es alcanzado. Los aspectos generales de particionamiento y el método elegido para nuestra propuesta VNS-BIO, se presentan en la siguiente sección.

3.1 Preliminares de particionamiento

El problema con el que lidiamos en el artículo está localizado en el agrupamiento bioinspirado y hemos reunido algunos trabajos interesantes que pueden ser encontrados en la literatura pero no encontramos trabajos relacionados acerca de agrupamiento bioinspirado con VNS y en este punto, nuestro algoritmo es una contribución al área.

3.2 Descripción de un problema de agrupamiento como uno de optimización

Dado un conjunto de n objetos denotado por $X = \{x_1, x_2, \dots, x_n\}$, en el cual $x_i \in \mathbb{R}^D$, sea K un número entero previamente conocido, el problema de agrupamiento consiste en encontrar una partición: $P = \{C_1, C_2, \dots, C_k\}$ de X , C_j siendo un grupo formado por objetos similares, satisfaciendo una función objetivo $f(x) : \mathbb{R}^D \rightarrow \mathbb{R}$, y las condiciones:

$$C_i \cap C_j = \emptyset \text{ para } i \neq j, \text{ y } \bigcup_{i=1}^k C_i = X.$$

Para medir la similitud entre dos objetos x_a y x_b se utiliza una función de distancia, siendo la distancia Euclidiana la más utilizada para agrupar objetos geográficos de manera compacta [14]. Así, la distancia entre dos diferentes elementos $x_i = (x_{i1}, \dots, x_{iD})$ y $x_j = (x_{j1}, \dots, x_{jD})$ es

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^D (x_{il} - x_{jl})^2}.$$

Los objetos de un grupo son similares cuando la distancia entre ellos es mínima, esto permite formular la función objetivo, como:

$$\sum_{j=1}^k \sum_{x_i \in C_j} d(x_i, \bar{x}_j)^2. \quad (1)$$

Es decir, minimizar (1) es deseado, donde \bar{x}_j , conocido como el elemento representante del grupo, es la medida de los elementos del grupo C_j :

$$\bar{x}_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i. \quad (2)$$

y es el centro del grupo. Con estas características, el agrupamiento es un problema de optimización combinatoria, y ha sido probado que su complejidad es NP-dura [18].

4 Estructura de vecindad

Los procesos de búsqueda por vecindades recorren el espacio de soluciones U por medio de un conjunto de transformaciones o movimientos. Las soluciones

que se obtienen a partir de otra a través de uno de los posibles movimientos son conocidas como los vecinos de esta solución y constituyen su vecindario. El conjunto de movimientos posibles da lugar a una relación de vecindad y a una estructura de vecindad en el espacio de soluciones. El esquema general de un procedimiento de búsqueda por vecindades consiste en generar una solución inicial y, hasta que un criterio de parada es cumplido, un movimiento es seleccionado iterativamente para modificar la solución [12]. El vecindario de una solución está formado por las soluciones que pueden ser accedidas a partir de esta empleando uno de los movimientos posibles. Formalmente, una estructura de vecindad sobre un espacio o un universo de búsqueda U es una función $E : U \rightarrow 2^S$ que asocia un vecindario $E(x) \subseteq U$ a cada solución $x \in U$. Una gran cantidad de métodos heurísticos propuestos en la literatura pertenecen a la clase de procedimientos de búsqueda por vecindades. La descripción en pseudocódigo del vecindario de búsqueda se muestra a continuación:

Algoritmo 1 Procedimiento búsqueda por vecindad.

```

 $x \leftarrow$  generar solución (U)
 $x^* \leftarrow x$ 
repetir
   $x \leftarrow$  seleccionar solución (E(x));
  si objeto  $x$  es mejor que el objeto  $x^*$  entonces
     $x^* \leftarrow x$ ;
  fin si
hasta que criterio de parada se cumple

```

La selección de la estructura de vecindad es fundamental en el éxito de los procedimientos de búsqueda ya que determina la calidad del conjunto de movimientos aplicado. Los movimientos combinados aparecen cuando varios movimientos son ejecutados de manera consecutiva sobre una solución. Una combinación adecuada de movimientos enriquece a los vecindarios, lo cual permite tomar pasos más amplios en la aproximación al óptimo. Una característica importante de los movimientos es la factibilidad de las soluciones contribuidas. Formalmente, los procedimientos que solo toman en cuenta movimientos factibles son asociados al concepto, un poco más restrictivo, de estructura de vecindad $E : S \rightarrow 2^S$ que asocia un vecindario $E(x) \subset S$ de soluciones factibles a cada función factible $x \in S$. Las principales metaheurísticas de búsqueda por vecindades se enfocan en el proceso de selección de movimiento. Sin embargo, además la selección de una estructura de vecindad sobre la cual articular la búsqueda; hay otras preguntas relevantes con respecto al éxito del

procedimiento de búsqueda por vecindades, tales como: la función objetivo, evaluación, el procedimiento para generar la solución inicial y el criterio de parada.

4.1 Búsqueda por entorno variable (VNS)

Es una metaheurística reciente que consiste en cambiar la estructura de vecindad en un modo sistemático. La idea original era considerar diferentes estructuras de vecindad y cambiarlas para escapar sistemáticamente de los óptimos locales. El VNS básico obtiene una solución del vecindario de la solución actual, ejecuta una búsqueda local monótona alrededor de esta hasta que se alcanza un óptimo local, el cual reemplaza la solución actual si ocurre una mejora sino se modifica la estructura de vecindad. La representación de las soluciones para problemas de gran tamaño puede ser difícil de leer, y muchos problemas, los problemas Euclidianos en particular (tal como el particionamiento), permiten una descomposición natural, una rutina enfocada permite la representación de la información previamente mencionada para sub-problemas selectos; concretamente, en alguna región del espacio donde la ruta es trazada [12]. VNS está basado en tres simples hechos: 1) Un mínimo local con una estructura de vecindad no lo es necesariamente con una estructura de vecindad diferente. 2) Un mínimo global es un mínimo local con cada estructura de vecindad posible. 3) Para varios problemas, los mínimos locales con la misma o diferente estructura de vecindad están relativamente cerca. Esta última observación, la cual es empírica, implica que los óptimos locales proporcionan información acerca del mínimo global. Los hechos del 1 al 3 sugieren el empleo de diversas estructuras de vecindad en las búsquedas locales para afrontar un problema de optimización. El cambio de estructura de vecindad puede ser realizado de manera determinista, estocástica o determinista y estocástica al mismo tiempo. En nuestro algoritmo el cambio de estructura de vecindad es estocástica y determinista.

4.2 Intensificación y diversificación

Intensificación y diversificación son estrategias de búsqueda que usualmente emplean las metaheurísticas con el objetivo de balancear la exploración del espacio factible, esto significa que intensificamos la exploración alrededor de zonas prometedoras y que diversificamos la búsqueda en zonas poco exploradas. Parece ser que estos conceptos se contradicen mutuamente pero son un detalle esencial para el diseño de una metaheurística. Intensificación es explorar cuidadosamente e intensivamente alrededor de buenas soluciones mientras que la diversificación se emplea para guiar la búsqueda hacia regiones no visitadas. Podemos

caracterizar la intensificación en este trabajo por medio de una búsqueda local que merodea en una sola estructura de vecindad, permitiendo incluso visitas que empeoran la solución actual, en un intento de intensificar aún más la búsqueda sobre la base de soluciones ya visitadas, pero intensificación pura no es suficiente si nuestro objetivo es obtener soluciones de alta calidad. Cuando la búsqueda llega a un punto sin salida, es poco probable que el tal represente al óptimo real en la mayoría de los casos. Por lo tanto, una estrategia capaz de expandir la búsqueda es necesaria. Esto puede considerarse como visitar y explorar diferentes regiones del espacio de búsqueda en lugar de concentrarse en el perímetro cerrado de una buena solución. En la literatura de optimización heurística este concepto es conocido como diversificación y en este trabajo se lleva a cabo por medio del cambio de estructura de vecindad sistemático N_1, N_2, \dots, N_K donde N_1 es una solución donde el primer centroide es el objeto 1 y los demás centroides son elegidos al azar, hasta N_K donde K es el último objeto.

5 VNS bioinspirado

El algoritmo comienza por cargar una matriz de disimilitud de tamaño n por n , enseguida los parámetros de VNS son leídos; *itBL* e *itVNS*, el primero indica cuántas iteraciones de búsqueda local serán realizadas en cada estructura de vecindad generada, el algoritmo se construye para generar n estructuras de vecindad y el parámetro *itVNS* muestra cuántas veces se explorará el conjunto completo de estructuras de vecindades definidas. Tenemos un arreglo de objetos geográficos y las distancias que hay entre ellos se obtienen de la matriz de disimilitud. El algoritmo repetirá su ciclo principal *itVNS* veces, este ciclo consiste en generar N_K estructuras de vecindad empezando desde una solución inicial generada aleatoriamente. En cada una de las estructuras se realizarán *itBL* iteraciones de una búsqueda local, la cual es el principal elemento bioinspirado de nuestro algoritmo.

El algoritmo utiliza un arreglo de tipo entero para almacenar y actualizar el valor de disimilitud de cada grupo formado para determinar cuál es el grupo más disímil, de esta manera en la línea 13 del algoritmo basta con examinar este arreglo para elegir al centroide adecuado. Del grupo elegido el elemento más alejado del centroide representa a la idea de cómo una persona abandona un grupo social debido a diferencias. El objeto seleccionado será asignado al grupo que contiene al elemento más cercano (parecido) a sí mismo que no pertenece a su grupo (líneas 14 a 16), emulando así a un ser humano que se integra a un grupo donde algún conocido o persona afín se encuentra, por lo tanto el objeto es ahora asignado al centroide (líder) del nuevo grupo.

Algoritmo 2 VNS-BIO.

Entrada: matriz de disimilitud $(n \times n)$.**Entrada:** $itVNS$ (iteraciones globales) y $itBL$ (iteraciones de búsqueda local).**Entrada:** k número de grupos a formar.

```

1: Inicializar arreglo de objetos geográficos  $objArray$ 
2: Inicializar VNS con los parámetros  $itVNS$  e  $itBL$ 
3:  $c_{global} \leftarrow 1$ 
4:  $c_{local} \leftarrow 1$ 
5:  $solEncontrada \leftarrow Falso$ 
6: Construir una solución inicial
7: mientras  $c_{global} < itVNS$  hacer
8:    $N_k \leftarrow 1$ 
9:   mientras  $N_k < n$  hacer
10:    Búsqueda local:
11:    Mover a la estructura de vecindad  $N_k$ 
12:    mientras  $c_{local} < itBL$  and  $solEncontrada = Falso$  hacer
13:      Seleccionar el centroide que representa el grupo con la mayor disimilitud  $c1$ 
14:      Seleccionar un elemento  $obj$  asignado a  $c1$  cuya distancia a  $c1$  es la mayor de entre todos los elementos del grupo.
15:      Seleccionar el centroide  $c2$  que contiene al objeto no-centroide más cercano a  $obj$ .
16:      El objeto  $obj$  es reasignado  $c2$  y el costo de la solución es modificado de acuerdo a la asignación realizada.
17:      Prohibir la selección de  $c1$  y del objeto  $obj$  durante un cierto número de iteraciones. (estrategia tabú).
18:      si el costo de la nueva solución es mejor que el costo de la solución actual entonces
19:         $solEncontrada \leftarrow Verdadero$ 
20:      si no
21:         $c_{local} \leftarrow c_{local} + 1$ 
22:      fin si
23:    fin mientras
24:     $N_k \leftarrow N_k + 1$ 
25:  fin mientras
26:   $c_{global} \leftarrow c_{global} + 1$ 
27: fin mientras

```

En esta parte una estrategia muy conocida de búsqueda tabú es implementada, los estados tabú activos e inactivos: En el algoritmo el grupo que perdió un miembro permanecerá tabú activo durante un cierto número de iteraciones ($k - 1$ por defecto) así como el objeto que se movió (línea 17) para evitar que el proceso de búsqueda se cicle como fue observado en la fase experimental. Estos pequeños cambios mejorarán o empeorarán el costo de la solución actual, aquellos que la mejoren serán almacenados mientras que los otros serán descartados. El ciclo de búsqueda local termina cuando un óptimo local es alcanzado (no hay mejoras) o bien cuando el límite de iteraciones $itBL$ sea alcanzado (líneas 12 y 18 a 21). Por otro lado las estructuras de vecindad representan un alboroto en la sociedad, emulando el descontento que yace entre la sociedad el cual los fuerza a llevar a cabo grandes cambios en el sistema social o político esperando lograr mejores condiciones para la sociedad en general; por lo regular esto significa un reemplazo de los líderes actuales para formar nuevas asociaciones u organismos quizá más equitativos y justos. En el algoritmo esto es emulado al reemplazar los centroides por objetos no centroides (líneas 8 y 24).

5.1 Experiencia computacional

Para probar el algoritmo VNS-BIO, consideramos 4 valores para el número de grupos a formar (G) sobre dos mapas; uno pequeño, Colima y uno de tamaño grande, Puebla. Con 371 y 2580 datos geográficos (datos geo-referenciados) respectivamente. Las corridas son presentadas en la tabla 1 y se comparan los resultados con VNS (estándar) y Recocido Simulado (RS). La notación en la tabla 1 es: G (número de grupos), T (Tiempo en segundos), CF (Costo de la Función) y A (algoritmo).

Las características del equipo utilizado para ejecutar las siguientes pruebas son: Procesador AMD E-350 a 1.60 GHz de doble núcleo, 4GB (2 x 2GB DDR3) en memoria RAM, disco duro 5400 RPM, 8MB cache, sata 3.0Gb/s y sistema operativo Windows 7 Ultimate versión de 64 bits.

Parámetros para RS: Temperatura inicial: 20000, Temperatura final: 0.1, $\alpha = 0.98$ y $L(t) = 5$. Parámetros para VNS Puro: $itVNS = 3$, $itBL = 30$. Parámetros para VNS-BIO: $itVNS = 4$, $itBL = 100$. Basado en un diseño de experimentos para calibrar los parámetros de cada algoritmo, como se puede ver en [2].

Cabe notar que dada la naturaleza de la función bioinspirada que obtiene a los vecinos de la solución actual en VNS-BIO tiene una complejidad menor que VNS Puro, por lo tanto se requiere un mayor valor para el parámetro $itBL$ para intensificar más la búsqueda.

Distrito	Objetos	G	A	CF	T
Colima	371	15	RS	17.3895	00:01
			VNS Puro	17.2272	00:01
			VNS Bio	15.6187	00:01
		30	RS	12.73529	00:01
			VNS Puro	11.8744	00:01
			VNS Bio	11.826	00:01
		45	RS	10.0268	00:02
			VNS Puro	9.8788	00:01
			VNS Bio	9.368798	00:01
		60	RS	8.931499	00:02
			VNS Puro	8.634	00:01
			VNS Bio	7.8389	00:01
Puebla	2580	15	RS	363.7480	00:52
			VNS Puro	360.6902	00:55
			VNS Bio	355.6597	00:47
		30	RS	261.8216	02:07
			VNS Puro	257.9909	01:49
			VNS Bio	256.2463	01:27
		45	RS	220.6704	02:17
			VNS Puro	217.2667	02:13
			VNS Bio	216.2123	02:14
		60	RS	192.1496	03:36
			VNS Puro	189.0808	03:00
			VNS Bio	188.9468	03:00

Tabla 1: VNS-BIO, resultados comparativos.

Hemos desarrollado una herramienta de información geográfica que transforma las soluciones obtenidas en una representación gráfica. La salida de esta herramienta puede apreciarse en las figuras 2 y 3. Cada grupo tiene un color distinto, además se observa la consistencia de los resultados obtenidos en la tabla 1 al generar grupos con características robustas de compacidad.

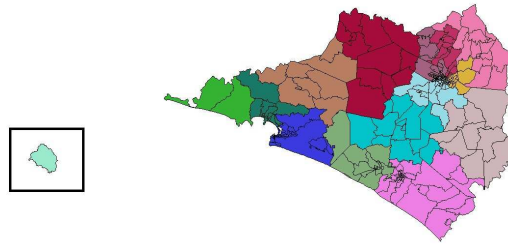


Figura 2: Representación en mapa: Colima, $G = 15$ con VNS-BIO.

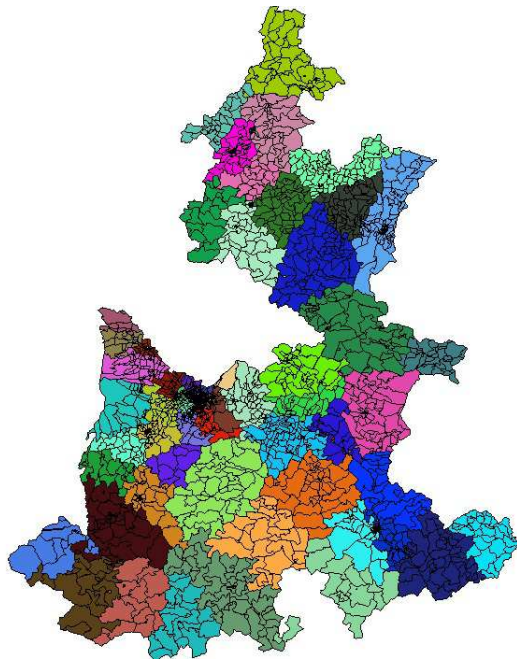


Figura 3: Representación en mapa: Puebla, $G = 60$ con VNS-BIO.

6 Conclusiones

En este trabajo se analiza el problema de agrupamiento geográfico con base en dos instancias de prueba asociados a zonas geográficas vinculados con sus respectivos mapas como son de los estados de Colima y Puebla de México, para ello se ha comparado los resultados de nuestra propuesta de algoritmo VNS-BIO en contraste con los resultados de los algoritmos RS y VNS Puro dados en la tabla 1. En el diseño del experimento se consideran las dos instancias de pruebas, Colima con 371 objetos geográficos, y otra prueba de considerable orden con 2580 objetos geográficos para el caso de Puebla. Se ha decidido tomar diferentes valores del número de grupos para cada instancia como son $p = |G|$: 15, 30, 45 y 60. En los resultados de la tabla 1 para las instancias de Colima, se puede apreciar que VNS-BIO obtiene mejores valores que los otros dos algoritmos, además la diferencia en el tiempo de cómputo es despreciable para todos los algoritmos considerados. En el caso de Puebla, VNS-BIO para $G = 30$ y 45 obtiene valores de costo de la función mínimo aceptables, pero logra los mejores tiempos de cómputo que los otros algoritmos en las dos instancias. Para el caso de la prueba de mayor tamaño, se puede apreciar que RS tiene el menor tiempo de ejecución, seguido por VNS Puro y VNS-BIO; aunque la diferencia de tiempo entre estos últimos no es tan grande, lo cual puede permitir deducir que la diferencia crecerá para mapas de mayor tamaño. En general, se puede concluir que VNS-BIO ha superado a los otros dos algoritmos y un pequeño sacrificio en el tiempo de cómputo puede lograr soluciones de mayor calidad con el algoritmo propuesto bioinspirado, además existe la posibilidad de mejorar, por ejemplo obtener una mayor optimización de código en la parte intensiva del algoritmo, la selección e intercambio de objetos geográficos. Por otra parte, los datos geográficos utilizados en este estudio son datos reales de los estados de Colima y Puebla en México, los cuales pueden ser empleados para problemas reales de agrupamiento y optimización, lo cual se traduce en una contribución de este trabajo de investigación. Como trabajo prospectivo, se considera la implementación de un algoritmo constructivo para generar buenas soluciones iniciales así como una combinación más estrecha con búsqueda tabú, utilizando estrategias más complejas. También se tiene la hipótesis que mejores estructuras de memoria podrían mejorar el rendimiento de nuestros procedimientos de intensificación y diversificación. Finalmente, un diseño de experimentos estadístico factorial contribuirá a la selección de buenos parámetros para conseguir soluciones de calidad y además es posible considerar al tiempo como función objetivo para comparar los diferentes métodos en pruebas que corran en un mismo tiempo.

Referencias

- [1] Barricelli, N.A. (1954) “Esempi numerici di processi di evoluzione”, *Methodos* **21-22**: 45–68.
- [2] Bernábe, M.B.; Espinosa, J.E.; Ramírez, J. (2009) “Evaluación de un algoritmo de recocido simulado con superficies de respuestas”, *Revista de Matemática: Teoría y Aplicaciones* **16**(1): 159–177.
- [3] Bremermann, H. (1958) “The evolution of intelligence: The nervous system as a model of its environment”, Technical Report no. 1, Department of Mathematics, University of Washington, Seattle WA, 477(17).
- [4] Cruz, A.R. (2012) *Una Propuesta Evolutiva para el Desarrollo de Comportamientos Robóticos*. Master’s Thesis, Facultad de Física e Inteligencia Artificial, Universidad Veracruzana, Xalapa, México.
- [5] Dorigo, M. (1992) *Optimization, Learning and Natural Algorithms*. Doctoral thesis, Politecnico di Milano, Italy.
- [6] Fogel, L.J.; Owens, A.J.; Walsh, M.J. (1966) *Artificial Intelligence through Simulated Evolution*. John Wiley & Sons, New York.
- [7] Forbes, N. (2004) *Imitation of Life. How Biology is Inspiring Computing*. MIT Press, Cambridge MA.
- [8] Friedman, G.J. (1956) *Selective Feedback Computers for Engineering Synthesis and Nervous System Analogy*. Master’s Thesis, Engineering, University of California, Los Angeles CA.
- [9] Holland, J.H. (1992) *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge MA.
- [10] Karaboga, D. (2005) “An idea based on honey bee swarm for numerical optimization”, TR06/REP, Computer Engineering Department, Engineering Faculty, Erciyes University.
- [11] Kennedy, J.; Eberhart, R. C. (1995) “Particle swarm optimization”, in *Proceedings of IEEE International Conference on Neural Networks*, vol. 4, IEEE Computer Society, Piscataway NJ: 1942–1948.
- [12] Mladenovic, N.; Hansen, P. (1997) “Variable neighborhood search”, *Computers & Operations Research* **24**(11): 1097–1100.

- [13] Olmo, J.L. (2013) *Minería de Datos mediante Programación Automática con Colonias de Hormigas*. Tesis de Doctorado, Departamento de Informática y Análisis Numérico, Escuela Politécnica Superior, Universidad de Córdoba, España.
- [14] Piza, E.; Murilo, A.; Trejos, J. (1999) “Nuevas técnicas de particionamiento en clasificación automática”, *Revista de Matemática: Teoría y Aplicaciones* **16**(1): 51–66.
- [15] Price, K.; Storn, R.; Lampinen J. (2005) *Differential Evolution: A Practical Approach to Global Optimization*. Natural Computing Series, Springer, Berlin.
- [16] Rechenberg, I. (1973) *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog Verlag, Stuttgart.
- [17] Stepney, S.; Smith, R.E.; Timmis, J.; Tyrrell, A. M.; Neal, M. J.; Hone, A. N. W. (2005) “Conceptual frameworks for artificial immune systems”, *International Journal of Unconventional Computing* **1**(3): 315–338.
- [18] Vicente, E.; Rivera, L.; Mauricio, D. (2005) “Grasp en la resolución del problema de clustering”, *Revista de Investigación en Sistemas e Informática*, **2**(2): 16–25.

