

Universidad Central “Marta Abreu” de Las Villas,
Facultad de Matemática, Física y Computación,
Centro de Estudios de Informática.



Algoritmo VMO continuo con topologías estructuradas.

Autor: Román Eduardo Lugones Acosta.

Tutores: Dr. Rafael Bello.

Lic. Gonzalo Nápoles.

Santa Clara, 2014

Dictamen con derechos de autor para MFC



Hago constar que el presente trabajo fue realizado en la Universidad Central Marta Abreu de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencias de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor _____

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor _____

Firma del jefe del Seminario _____

No sientas que te falte nada salvo el corazón,
recuerda que lo tienes todo si te alumbra el sol.

A los que están de pie sobre la tierra y hacen de mi algo mejor, a mis muertos a los que tantas veces me encomendé en días de agonía.

Agradecimientos

A tantas personas va dedicada esta página que no me aventuro a nombrarlas, por miedo a que al hurgar en la memoria se queden algunos, y para no pecar, me refugio en Machado cuando reza "conmigo vais mi corazón os lleva".

RESUMEN

Las meta-heurísticas poblacionales constituyen, en la actualidad, una poderosa herramienta para la solución de problemas complejos de optimización. Investigaciones han revelado que no existe la mejor de las meta-heurísticas para solucionar cualquier problema complejo, por lo que la búsqueda de nuevas estrategias en la exploración del espacio de búsqueda para una mejor calidad de las soluciones se ha convertido en un objetivo deseado. De manera que la presente investigación estará encaminada a implementar y validar un modelo basado en una meta heurística poblacional, VMO, la cual será implementada siguiendo una topología estructurada, con el objetivo de evitar estados de convergencia prematura o estancamiento. Las topologías a usar son: en un primer algoritmo, topologías con k-subpoblaciones sin solapamiento; mientras que en la segunda etapa topologías con k-subpoblaciones con solapamiento. Ambas con el objetivo de presentar un modelo más robusto. Los resultados de esta investigación son la obtención de distintas variantes VMO_TOP1 y VMO_TOP2 con mejor comportamiento que el modelo basado en VMO clásico.

ABSTRACT

Population meta-heuristics are currently a powerful tool in the solution of complex optimization problems. Some investigations have shown that there is not a best of meta-heuristics for solving any complex problem. Therefore seeking for new strategies in the exploration of the search space in order to achieve a better quality in the solutions has become a desired goal. So this investigation will be focused in the implementation and validation of a model based in a population meta-heuristic, VMO, which will be implemented following a structured topology aiming to avoid premature convergence or stemming states. The used topologies are: in the first algorithm topologies with k-subpopulations without overlapping. While in a second stage we use topologies with k-subpopulations with overlapping. Both are used with the goal of presenting a more robust model. The results of this investigation allowed to obtain distinct variants VMO_TOP1 and VMO_TOP2 with a better performance than the classic VMO based model.

TABLA DE CONTENIDOS

INTRODUCCIÓN.....	1
CAPÍTULO 1 MÉTODOS HEURÍSTICOS DE BÚSQUEDA.....	2
IMPORTANCIA DE LOS PROBLEMAS DE OPTIMIZACIÓN.....	3
MÉTODOS DE SOLUCIÓN DE PROBLEMAS.....	4
META-HEURÍSTICAS.....	8
<i>Meta-heurísticas de trayectoria simple.....</i>	<i>12</i>
<i>Meta-heurísticas poblacionales.....</i>	<i>14</i>
EJEMPLOS DE USO DE LAS META-HEURÍSTICAS EN PROBLEMAS REALES.....	17
<i>Meta-heurísticas en la Logística Portuaria.....</i>	<i>17</i>
<i>Las redes de telecomunicaciones y las meta-heurísticas.....</i>	<i>18</i>
<i>Manejo de empresas de transporte público de autobuses.....</i>	<i>19</i>
CONCLUSIONES PARCIALES.....	19
CAPÍTULO 2 ALGORITMO VMO CON TOPOLOGÍA LOCAL.....	2
OPTIMIZACIÓN BASADA EN MALLAS VARIABLES.....	2
<i>Descripción general de la meta heurística.....</i>	<i>3</i>
<i>Proceso de expansión.....</i>	<i>4</i>
<i>Proceso de contracción.....</i>	<i>7</i>
TOPOLOGÍA DE VECINDAD EN PSO.....	9
<i>Modelo gbest y lbest.....</i>	<i>11</i>
IMPLEMENTACIÓN DEL VMO CON TOPOLOGÍAS.....	13
<i>VMO local sin solapamiento.....</i>	<i>17</i>
<i>VMO local con solapamiento.....</i>	<i>18</i>
CONCLUSIONES PARCIALES.....	19
CAPÍTULO 3 ANÁLISIS EXPERIMENTAL DE LOS RESULTADOS.....	2
ESPECIFICACIONES DEL EXPERIMENTO.....	2
TÉCNICAS ESTADÍSTICAS.....	16
<i>Descripción de los tests estadísticos utilizados.....</i>	<i>17</i>
MARCO EXPERIMENTAL.....	18
ANÁLISIS DE LOS RESULTADOS DE LOS TEST ESTADÍSTICOS PARA LAS DISTINTAS VARIANTES DE DISEÑO.....	19
<i>Dimensión 10.....</i>	<i>19</i>
<i>Dimensión 30.....</i>	<i>21</i>

<i>Dimensión 50</i>	23
CONCLUSIONES PARCIALES.....	25
CONCLUSIONES	1
RECOMENDACIONES	1
REFERENCIAS BIBLIOGRÁFICAS	1
ANEXO A. TABLAS DE LOS VALORES OBTENIDOS PARA CADA UNA DE LAS FUNCIONES DE PRUEBA DEL CEC´2005 POR LOS MODELOS ESTUDIADOS	1

LISTA DE FIGURAS

Figura 1.1 <i>Clasificación de los problemas de optimización</i>	3
Figura 1.2 <i>Encasillamiento de los algoritmos heurísticos en óptimos locales</i>	7
Figura 1.3 <i>Meta-heurísticas de acuerdo a su inspiración</i>	11
Figura 1.4 <i>Espacio de búsqueda</i>	12
Figura 2.1 <i>Esquema general de modelo VMO</i>	9
Figura 2.2 (a) <i>Topología de estrella (gbest)</i> (b) <i>Topología de anillo (lbest)</i>	11
Figura 2.3 <i>Esquema de generación de subpoblaciones</i>	15
Figura 2.4 <i>Esquema general de la topología estructurada</i>	16
Figura 2.5 <i>Esquema del modelo VMO-Top1</i>	17
Figura 2.6 <i>Proceso general de modelo VMO-Top1</i>	18
Figura 2.7 <i>Esquema del modelo VMO-Top2</i>	18
Figura 2.8 <i>Proceso general de modelo VMO-Top2</i>	19

LISTA DE TABLAS

Tabla 1.1 <i>Número de artículos por técnicas bio-inspiradas (Delgado, 2010)</i>	10
Tabla 3.1 <i>Resultados para dimensión 10</i>	19
Tabla 3.2 <i>Mediana</i>	20
Tabla 3.3 <i>Resultados del test de Wilcoxon para dimensión 10</i>	21
Tabla 3.4 <i>Resultados para dimensión 30</i>	21
Tabla 3.5 <i>Mediana</i>	23
Tabla 3.6 <i>Resultados del test de Wilcoxon para dimensión 30</i>	23
Tabla 3.7 <i>Resultados para dimensión 50</i>	24
Tabla 3.8 <i>Mediana</i>	25
Tabla 3.9 <i>Resultados del test de Wilcoxon para dimensión 50</i>	25
Tabla A.1 <i>Resultados del modelo VMO puro</i>	1
Tabla A.2 <i>Resultados del modelo VMO-Top1</i>	2
Tabla A.3 <i>Resultados del modelo VMO-Top2</i>	3

INTRODUCCIÓN

Los problemas de optimización continua desempeñan un papel esencial en la formulación de muchas situaciones de la vida real; partiendo de problemas clásicos hasta problemas más complejos que se abordan necesariamente desde la perspectiva de la Inteligencia Artificial. Algunos de estos problemas de optimización son relativamente fáciles de resolver como los problemas lineales (las restricciones y la función objetivo constituyen expresiones lineales) y otros no tan simples, los cuales por su propia naturaleza no pueden ser abordados mediante un procedimiento de búsqueda exhaustivo que obtenga la mejor solución tras una búsqueda por todo el dominio de posibles valores (Molina Cabrera, 2007).

Debido a la gran complejidad que estos problemas han alcanzado, en la mayoría de las veces, la aplicación de un método exacto para su solución, se ha vuelto impracticable. Esto ha proporcionado el auge de los métodos heurísticos, vocablo este, que surge del griego *heuriskein*, que significa encontrar. Un método o procedimiento heurístico pudiera definirse entonces como (Martínez et al., 2006):

“Un método heurístico es un procedimiento para resolver un problema complejo de optimización mediante una aproximación intuitiva, en la que la estructura del problema se utiliza de forma inteligente para obtener una buena solución de manera eficiente”

Una de las principales desventajas de estos métodos, es que dependen en gran medida del problema para el cual fueron diseñados. Por esta razón surge un grupo de métodos bajo el nombre de meta-heurísticas, con el objetivo de obtener mejores soluciones que los heurísticos tradicionales para un mayor número de problemas con solo realizársele pequeñas modificaciones. En (Glover and Kochenberger, 2003) se definen los procedimientos meta-heurísticos como:

“Los procedimientos meta heurísticos son una clase de métodos aproximados que están diseñados para resolver problemas complejos de optimización, en los que los heurísticos clásicos no son efectivos. Las metas heurísticas proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos”.

Numerosos son los procedimientos meta-heurísticos creados, sobre todo a partir de la formulación del así denominado Teorema de No Free Lunch (Wolpert and Macready, 1997) que dio pie a una suerte de catarsis, a partir de la cual quedó definitivamente claro que un algoritmo de búsqueda u optimización se comporta en estricta concordancia con la cantidad y calidad del conocimiento específico del problema que incorpora, o sea que no existe un único algoritmo capaz de resolver con la mayor calidad todo tipo de problemas.

Dentro de las tantas clasificaciones que a lo largo de los años se le ha dado a las meta-heurísticas, una de las más comunes es subdividir las en: métodos de búsqueda local o meta-heurísticas de trayectoria simple y meta-heurísticas poblacionales.

Los métodos de Búsqueda Local (BL), se enfocan en la explotación de las soluciones ya encontradas, o sea, realizar un proceso de búsqueda en las vecindades de una solución, denominada solución actual, que en un inicio se pudiera generar de manera aleatoria, sufriendo un proceso de transformación en el cual se obtiene una nueva solución que reemplazará a la actual si mejora a esta siguiendo algún criterio de comparación. Este proceso se repetirá hasta que se genera un número máximo de soluciones o se alcanza un óptimo local.

Varios de los métodos de Búsqueda Local que se han desarrollado, podemos citar por ejemplo: los métodos de Cuasi-Newton (Luenberger, 1989) entre los cuales encontramos L-BFGS y L-BFGS-B, la muy utilizada Búsqueda Tabú (Glover and Laguna, 1997), el Método Solis Wets (Solis and Wets, 1981), Método Simplex de Nelder y Mead (Spendley et al., 1962).

Dichos métodos son muy precisos y de rápida convergencia, pudiendo caer estancados en óptimos locales y no alcanzando en ocasiones el óptimo global que es lo que realmente se pretende. Esto es conocido como el fenómeno de la convergencia prematura.

Por otra parte están los algoritmos poblacionales. Estos emplean un conjunto de soluciones (población) en cada iteración del algoritmo en lugar de utilizar una única solución como las meta-heurísticas del grupo anterior, por lo que su búsqueda realiza el énfasis en el fomento de la exploración. Estas proporcionan de forma intrínseca un mecanismo de exploración paralelo al proceso de búsqueda de soluciones, y su eficacia depende en gran medida de cómo se manipule dicha población. Dentro de esta clasificación se destacan los Algoritmos Evolutivos y los algoritmos basados en Inteligencia Colectiva.

Dentro de este subgrupo se puede citar entonces los Algoritmos Genéticos (AGs) (Goldberg, 1998), la Búsqueda Dispersa (Laguna and Martí, 2003), los Algoritmos Basados de Distribuciones (*Estimation of Distribution Algorithms*, EDA) (Lozano et al., 2006), la Optimización de Colonias de Hormigas (*Ant Colony Optimization*; ACO) (Dorigo and Caro, 1999) y los Sistemas de Partículas (*Particle Swarm Optimization*, PSO) (Kennedy and Eberhart, 1995), Agujero Negro (Hatamlou, 2013), Optimización de Mallas Variables de su inglés (*Variable Mesh Optimization*) (Puris et al., 2012) desarrollada por el grupo de Inteligencia Artificial de la Universidad Central “Marta Abreu” de Las Villas, entre otras.

Estos métodos trabajan en cada iteración del algoritmo, con un conjunto (población) de soluciones en vez de una única solución. De esta manera los algoritmos poblacionales proveen una forma natural para explorar el espacio de búsqueda. Por lo tanto la manera en que esta población es manipulada afectará de forma directa el rendimiento.

Como una alternativa para mejorar la eficiencia de las meta-heurísticas, han surgido varias propuestas que utilizan un enfoque de procesamiento estructurado de éstas. Con el uso de esta técnica se tratan de evitar los estados de convergencia prematura o estancamiento; una forma de lograr esto es preservando la diversidad en la población. Un método común para estructurar los algoritmos de inteligencia colectiva es utilizar múltiples subpoblaciones de forma tal que se mantenga el proceso de exploración de manera simultánea en varias zonas del espacio de búsqueda. Varios son los ejemplos de meta-heurística en las cuales se ha hecho uso de las topologías: el EPUS-PSO variante de PSO con topología de estrella (Hsieh et al., 2008), el DMS-PSO variante de PSO con topologías de subpoblaciones dinámicas (Zhao et al., 2008), en la optimización basada en las colonias de hormigas (Christodoulou, 2005), entre otras.

Teniendo en cuenta lo expuesto anteriormente y aprovechando las ventajas que puede traer la implementación de algoritmos poblacionales siguiendo una topología estructurada, la presente investigación estará encaminada a desarrollar un modelo basado en VMO, del cual se obtendrán dos variantes estructuradas; VMO-Top1 donde crearán subpoblaciones sin solapamiento y VMO-Top2 donde dichas subpoblaciones estarán solapadas. Con el objetivo de conocer si las soluciones que se obtienen muestran una verdadera mejoría de las soluciones estadísticamente hablando, con respecto a las que obtiene la meta-heurística

VMO, sin necesidad de incrementar la complejidad temporal y espacial del algoritmo original.

Por todo lo anteriormente expuesto enunciamos el siguiente **problema de investigación**:

- En el VMO clásico los nodos están totalmente comunicados, esta característica confiere una alta tasa de convergencia provocando que quede atrapado en soluciones locales.

En consecuencia con dicho problema pueden plantearse las siguientes **preguntas de investigación**:

- ¿Qué debe caracterizar una versión estructurada del VMO?
- ¿Qué elementos de deben tener en cuenta para comparar adecuadamente el desempeño de los modelos VMO-Top1 y VMO-top2 con la versión clásica?

En consecuencia con lo anterior se plantea el **objetivo general** de este trabajo:

- Diseñar e implementar una variante del algoritmo VMO continuo con topologías estructuradas, que incorpore estas estrategias en el proceso de generación de nodos.

Para ganar en claridad durante el proceso investigativo, se desglosa este objetivo general en los siguientes **objetivos específicos**:

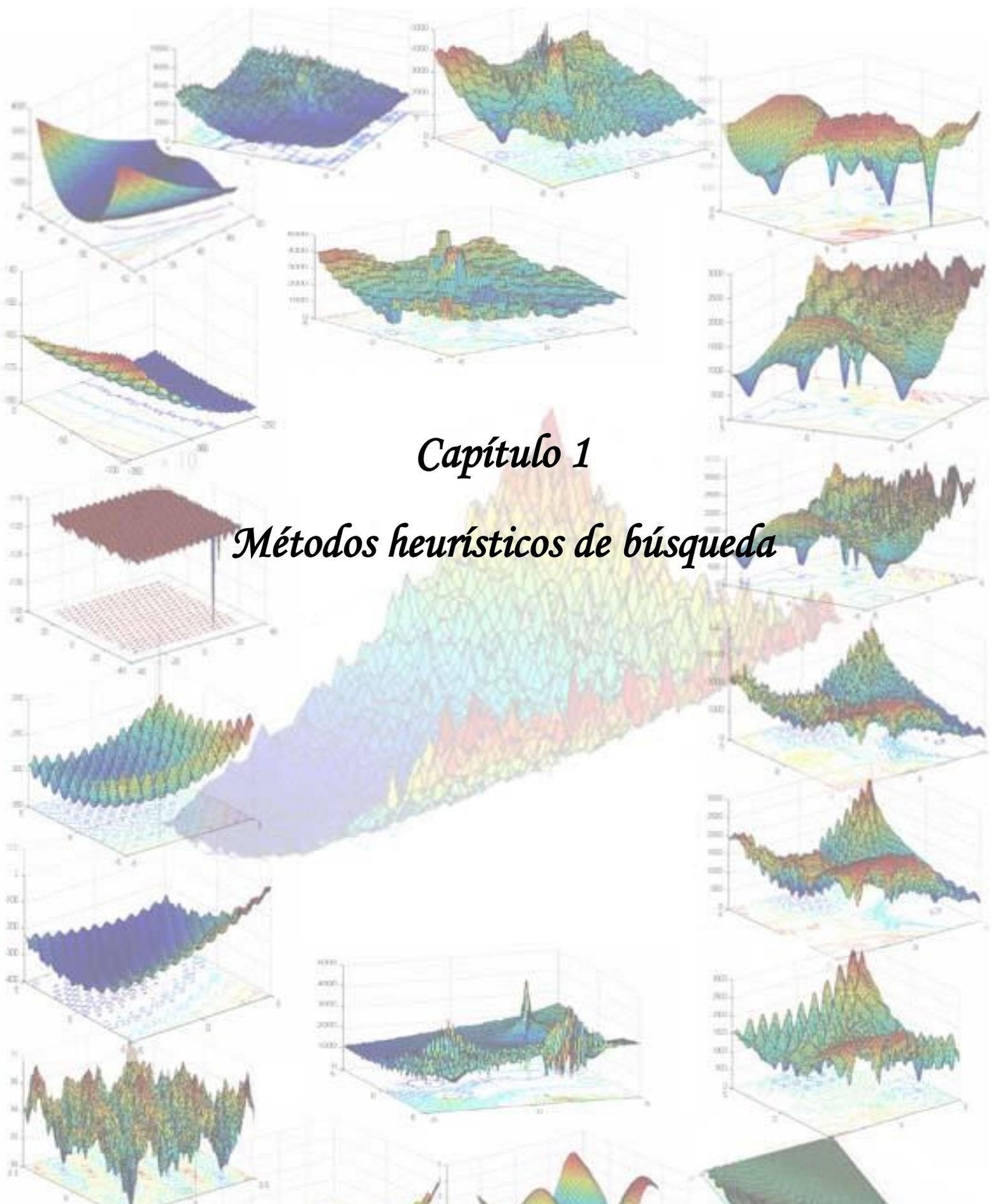
- Diseñar una nueva topología estructurada para VMO continuo, que esté inspirada en otras variantes locales poblacionales (ej. Topología de Anillo en Optimización de Enjambres de Partículas).
- Implementar dos alternativas distintas del modelo teniendo en cuenta la información a compartir.
- Diseñar un experimento estadístico que permita evaluar el desempeño del algoritmo, respecto a la variante clásica, en problemas multimodales.

Luego de haber construido el marco teórico se formuló la siguiente **hipótesis de investigación**:

En la versión estructurada de VMO aumenta el desempeño de esta meta-heurística en problemas de optimización global, en espacio de búsqueda multimodales, sin necesidad d aumentar la complejidad del modelo original.

Dicha hipótesis será validada si se comprueba que las soluciones obtenidas por las versiones estructuradas presentadas, difieren, mostrando mejoría, a las del VMO clásico.

Este trabajo se compone de tres capítulos. Luego de esta Introducción, en el Capítulo 1 se exponen los métodos heurísticos de búsqueda. En el Capítulo 2 se aborda de manera específica el modelo en sí y sus distintas alternativas. En el Capítulo 3 se realizan análisis estadísticos con el fin de validar los modelos propuestos. Finalmente se elaboran las Conclusiones y se listan las Referencias Bibliográficas que se utilizaron en este trabajo.



Capítulo 1

Métodos heurísticos de búsqueda

CAPÍTULO 1 MÉTODOS HEURÍSTICOS DE BÚSQUEDA

Los problemas de optimización continua desempeñan un papel esencial en la formulación de muchos problemas básicos de la vida real. Numerosas investigaciones aseguran este hecho, tanto en la gestión de servicios (planificación de los servicios hospitalarios (Lamiri et al., 2009), cabina de servicio (Schilde et al., 2011), los problemas de planificación de la salud (Nickel et al., 2012, Shariff et al., 2012), etc.), así como aplicaciones industriales (planificación de la producción (Clark et al., 2011), el avance ingeniería de diseño (Liao, 2010), la planificación financiera (Guillén et al., 2007), la gestión de riesgos (Papadakos et al., 2011), entre otros.

Todos ellos van desde problemas clásicos hasta problemas más complejos que se abordan necesariamente desde la perspectiva de la Inteligencia Artificial. algunos de estos problemas de optimización son relativamente fáciles de resolver como los problemas lineales (las restricciones y la función objetivo constituyen expresiones lineales) y otros no tan simples, los cuales por su propia naturaleza no pueden ser abordados mediante un procedimiento de búsqueda exhaustivo que obtenga la mejor solución tras una búsqueda por todo el dominio de posibles valores (Molina Cabrera, 2007).

De una manera sencilla resolver un problema de optimización no es más que, dado un dominio de soluciones y una función objetivo, encontrar un óptimo global de dicha función o sea dada una función $F: D \rightarrow R$ siendo D continua en R_n , encontrar un punto $x_0 \in D$ tal que:

- $F(x_0) \leq F(x)$ para todo $x \in D$, en el caso de minimización de la función F .
- $F(x_0) \geq F(x)$ para todo $x \in D$, en el caso de maximización de la función F .

La idea es asignar valores del dominio, permitidos por las restricciones la tal forma que la función sea optimizada. Los componentes de un problema de optimización son (Bello and Puris, 2009):

- Función objetivo (*Fitness function*).

- Conjunto de parámetros (desconocidos) los cuales afectan el valor de la función objetivo.
- Conjunto de restricciones que limitan los valores que se pueden asignar.

Importancia de los problemas de optimización

La optimización en el sentido de encontrar la mejor solución, o al menos una solución lo suficientemente buena para un problema es un campo de vital importancia en la vida real. Constantemente estamos resolviendo pequeños problemas de optimización, como el camino más corto de ir un lugar a otro, la organización de una agenda, etc.

En general éstos son lo suficientemente pequeños y pueden ser resueltos sin recurrir a elementos externos a nuestro cerebro. Pero conforme se hacen más grandes y complejos, el uso de los ordenadores para su resolución es inevitable. Debido a la gran importancia de los problemas de optimización, a lo largo de la historia de la Informática se han desarrollado múltiples métodos para tratar de resolverlos (Nieto, 2006). Una clasificación muy simple de estos métodos se muestra en la Figura 1.1 *Clasificación de los problemas de optimización*.

Inicialmente, las técnicas las podemos clasificar en exactas (o enumerativas, exhaustivas, etc.) y técnicas aproximadas. Las técnicas exactas garantizan encontrar la solución óptima para cualquier instancia de cualquier problema en un tiempo acotado. El inconveniente de estos métodos es que el tiempo necesario para llevarlos a cabo, aunque acotado, crece exponencialmente con el tamaño del problema, ya que la mayoría de éstos son NP-Completos. Esto provoca en muchos casos que el tiempo necesario para la resolución del problema sea inabordable. Por lo tanto, los algoritmos aproximados para resolver estos problemas están recibiendo una atención cada vez mayor por parte de la comunidad internacional a lo largo de los últimos años. Estos métodos sacrifican la garantía de encontrar el óptimo a cambio de encontrar una buena solución en un tiempo razonable (Nieto, 2006).

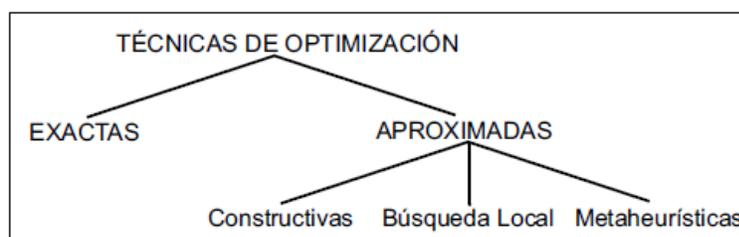


Figura 1.1 Clasificación de los problemas de optimización

Métodos de solución de problemas

El uso de la computadora para dar respuesta a problemas reales tales como: ¿Cuántos caminos hay para.....? ¿Listar todas las posibles soluciones para...? ¿Hay un camino para...? usualmente requiere de una búsqueda exhaustiva dentro del conjunto de todas las soluciones potenciales, por eso los algoritmos que resuelven este tipo de problemas reciben el nombre de algoritmos exactos o de búsqueda exhaustiva. Por ejemplo, si se desean encontrar todos los caminos de un laberinto, se deben examinar todos los caminos iniciando desde la entrada.

Un ejemplo es la búsqueda con retroceso o *backtracking*, que trabaja tratando continuamente de extender una solución parcial. En cada etapa de la búsqueda, si una extensión de la solución parcial actual no es posible, se va hacia atrás para una solución parcial corta y se trata nuevamente. El método retroceso se usa en un amplio rango de problemas de búsqueda, incluyendo el análisis gramatical (*parsing*), juegos, y planificación (*scheduling*).

La segunda técnica es tamiz o criba, y es el complemento lógico de retroceso en que se tratan de eliminar las no-soluciones en lugar de tratar de encontrar la solución. El método tamiz es útil principalmente en cálculos numéricos teóricos. Se debe tener en mente, sin embargo, que retroceso y tamiz son solamente técnicas generales. Se aplicarán en algoritmos cuyos requerimientos en tiempo son prohibitivos.

En general, la velocidad de los ordenadores no es práctica para una búsqueda exhaustiva de más de 100 elementos. Así, para que estas técnicas sean útiles, deben considerar solamente una estructura dentro de la cual se aproxima el problema. La estructura debe ser hecha a medida, a menudo con gran ingenio, para cuadrar con el problema particular, de modo que el algoritmo resultante será de uso práctico. Los métodos exactos de resolución de problemas se han aplicado con éxito en disímiles de problemas.

Algunos ejemplos de estos métodos son los algoritmos voraces, algoritmos de divide y vencerás, algoritmos de ramificación y poda, *backtraking*, etc. Todos estos procedimientos resuelven problemas que pertenecen a la clase P de forma óptima y en tiempo razonable. Como se ha comentado anteriormente, existe una clase de problemas, denominada NP, con

gran interés práctico para los cuales no se conocen algoritmos exactos con tiempos de convergencia en tiempo polinómico. Es decir, aunque existe un algoritmo que encuentra la solución exacta al problema, tardaría tanto tiempo en encontrarla que lo hace completamente inaplicable.

Además, un algoritmo exacto es completamente dependiente del problema (o familia de problemas) que resuelve, de forma que cuando se cambia el problema se tiene que diseñar un nuevo algoritmo exacto y demostrar su optimalidad. Para la mayoría de problemas de interés no existe un algoritmo exacto con complejidad polinómica que encuentre la solución óptima a dicho problema. Además, la cardinalidad del espacio de búsqueda de estos problemas suele ser muy grande, lo cual hace inviable el uso de algoritmos exactos ya que la cantidad de tiempo que necesitaría para encontrar una solución es inaceptable. Debido a estos dos motivos, se necesita utilizar algoritmos aproximados o heurísticos que permitan obtener una solución de calidad en un tiempo razonable(Esmorís, 2013).

El término heurística proviene del vocablo griego *heuriskein*, que puede traducirse como encontrar, descubrir o hallar. Desde un punto de vista científico, el término heurística se debe al matemático George Polya quien lo empleó por primera vez en su libro *How to solve it* (Polya, 2014). Con este término, Polya englobaba las reglas con las que los humanos gestionan el conocimiento común y que, a grandes rasgos, se podían simplificar en(Esmorís, 2013):

- Buscar un problema parecido que ya haya sido resuelto.
- Determinar la técnica empleada para su resolución así como la solución obtenida.
- En el caso que sea posible, utilizar la técnica y solución descrita en el punto anterior para resolver el problema planteado.

Existen dos interpretaciones posibles para el término heurística. La primera de ellas concibe las heurísticas como un procedimiento para resolver problemas. La segunda interpretación de heurística entiende que estas son una función que permite evaluar la bondad de un movimiento, estado, elemento o solución.

Existen métodos heurísticos (también llamados algoritmos aproximados, procedimientos inexactos, algoritmos basados en el conocimiento o simplemente heurísticas) de diversa

naturaleza, por lo que su clasificación es bastante complicada. Se sugiere la siguiente clasificación (Esmorís, 2013):

1. **Métodos constructivos:** Procedimientos que son capaces de construir una solución a un problema dado. La forma de construir la solución depende fuertemente de la estrategia seguida. Las estrategias más comunes son:

➤ **Estrategia voraz:** Partiendo de una semilla, se va construyendo paso a paso una solución factible. En cada paso se añade un elemento constituyente de dicha solución, que se caracteriza por ser el que produce una mejora más elevada en la solución parcial para ese paso concreto. Este tipo de algoritmos se dice que tienen una visión “miope” ya que eligen la mejor opción actual sin que les importe qué ocurrirá en el futuro.

➤ **Estrategia de descomposición:** Se divide sistemáticamente el problema en subproblemas más pequeños. Este proceso se repite (generalmente de forma recursiva) hasta que se tenga un tamaño de problema en el que la solución a dicho subproblema es trivial. Después, el algoritmo combina las soluciones obtenidas hasta que se tenga la solución al problema original. Los algoritmos más representativos de los métodos de descomposición son los algoritmos de divide y vencerás tanto en su versión exacta como aproximada.

➤ **Métodos de reducción:** Identifican características que contienen las soluciones buenas conocidas y se asume que la solución óptima también las tendrá. De esta forma, se puede reducir drásticamente el espacio de búsqueda.

➤ **Métodos de manipulación del modelo:** Consisten en simplificar el modelo del problema original para obtener una solución al problema simplificado. A partir de esta solución aproximada, se extrapola la solución al problema original. Entre estos métodos se pueden destacar: la linealización, la agrupación de variables, introducción de nuevas restricciones, etc.

2. **Métodos de búsqueda:** Parten de una solución factible dada y a partir de ella intentan mejorarla. Algunos son:

➤ **Estrategia de búsqueda local 1:** Parte de una solución factible y la mejora progresivamente. Para ello examina su vecindad y selecciona el primer movimiento que produce una mejora en la solución actual (*first improvement*).

- **Estrategia de búsqueda local 2:** Parte de una solución factible y la mejora progresivamente. Para ello examina su vecindad y todos los posibles movimientos seleccionando el mejor movimiento de ellos, es decir aquél que produzca un incremento (en el caso de maximización) más elevado en la función objetivo (*best improvement*).
- **Estrategia aleatorizada:** Para una solución factible dada y una vecindad asociada a esa solución, se seleccionan aleatoriamente soluciones vecinas de esa vecindad.

El principal problema que presentan los algoritmos heurísticos es su incapacidad para escapar de los óptimos locales. En la Figura 1.2 se muestra cómo para una vecindad dada el algoritmo heurístico basado en un método de búsqueda local se quedaría atrapado en un máximo local. En general, ninguno de los métodos constructivos descritos anteriormente tendrían por qué construir la solución óptima global.

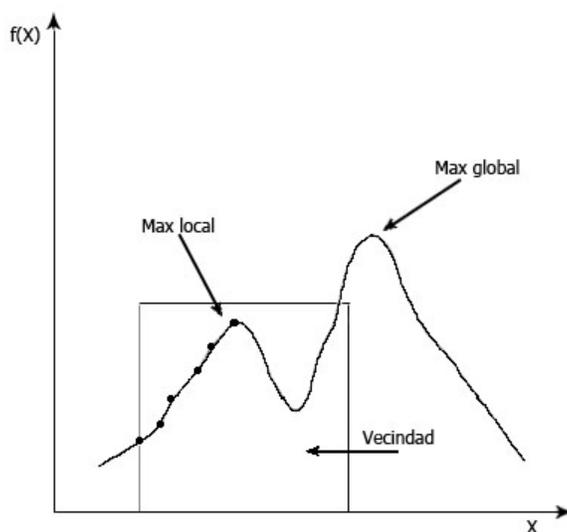


Figura 1.2 Encasillamiento de los algoritmos heurísticos en óptimos locales

Los algoritmos heurísticos no poseen ningún mecanismo que les permita escapar de los óptimos locales (Muñoz, 2007). Para solventar este problema se introducen otros algoritmos de búsqueda más elaborados que eviten en la medida de lo posible quedar atrapados. Estos algoritmos de búsqueda más inteligentes, son los denominados meta-heurísticos.

Meta-heurísticas

El término meta-heurística fue acuñado por F. Glover en el año 1986. Con este término, pretendía definir un “*procedimiento maestro de alto nivel que guía y modifica otras heurísticas para explorar soluciones más allá de la simple optimalidad local*” (Glover, 1986). Actualmente, existe varios trabajos científicos publicados que abordan problemas de optimización a través de meta-heurísticas, investigaciones sobre nuevas meta-heurísticas, y extensiones o mejoras de las meta-heurísticas ya conocidas.

La evolución de las meta-heurísticas durante los últimos 25 años ha tenido un comportamiento prácticamente exponencial. En el tiempo que transcurre desde las primeras reticencias (por su supuesta falta de rigor científico) hasta la actualidad, se han encontrado soluciones de muy alta calidad a problemas que hace tiempo parecían inabordable. De modo general, atendiendo los criterios de (Rodríguez Ortiz, 2010), se puede decir que las meta-heurísticas combinan ideas que provienen de cuatro campos de investigación bien distintos:

- Las técnicas de diseño de algoritmos (resuelven una colección de problemas)
- Algoritmos específicos (dependientes del problema que se quiere resolver).
- Fuente de inspiración (del mundo real).
- Métodos estadísticos.

Las meta-heurísticas son técnicas de inspiración biológica y física, que por medio de heurísticas emulan los procesos de optimización que se pueden observar en la naturaleza, como por ejemplo: la evolución. Los algoritmos meta-heurísticos no tienen una base matemática estricta y las funciones buscan directamente en el universo de soluciones con diversas estrategias. Por sus características, generalmente ellos proveen una respuesta en menor tiempo y de alta calidad a problemas de gran complejidad, mostrando su innegable potencial en aplicaciones del mundo real (Michalewicz, 2002, Dorigo, 2003).

Los algoritmos meta-heurísticos abordan problemas que por su condición no pueden ser resueltos satisfactoriamente por los métodos derivativos y por su estructura no necesitan la información diferencial. Sin embargo, aunque resuelven problemas más variados, su

naturaleza iterativa y la ausencia de información sobre las derivadas hacen que sean susceptibles al punto (o puntos) inicial de optimización y hacen que los parámetros del algoritmo ganen gran importancia en el funcionamiento y resultado del mismo (Rao, 1996, Feoktistov, 2006).

Las meta-heurísticas toman su inspiración de procesos que mejoran sus condiciones o cuyo objeto o comportamiento emergente muestre una notable mejoría en alguna característica, ya sea en la calidad final o en el refinamiento de ciertos parámetros del proceso. Por ejemplo, el temple de un material, es un proceso que mejora la calidad del material por el control de la temperatura (Jang, 1996, de Castro, 2007). Teniendo en mente los procesos, se intenta simularlos por medio de heurísticas emulando el comportamiento de mejoría (optimización) que presenta el proceso de donde nace la idea; así, estos métodos han logrado soluciones aceptables a problemas de gran dificultad en un tiempo y a unos costos razonables (Bäck, 1997).

Una primera conclusión que se puede extraer de las definiciones dadas es que, en muchos casos, son reglas de sentido común que permiten hacer una búsqueda inteligente. Debido a esta característica, para bastantes meta-heurísticas no existe un marco teórico que las sustente, sino que es a través de los buenos resultados experimentales donde encuentran su justificación. A pesar de que en la actualidad existen diversas opiniones en cuanto a la clasificación de las meta-heurísticas, a continuación se presentan los siguientes criterios de clasificación (Esmorís, 2013):

- **Atendiendo a la inspiración:** Natural: algoritmos que se basan en un símil real, ya sea biológico, social, cultural, etc. Sin inspiración: algoritmos que se obtienen directamente de sus propiedades matemáticas.
- **Atendiendo al número de soluciones:** Poblacionales: buscan el óptimo de un problema a través de un conjunto de soluciones. Trayectoriales: trabajan exclusivamente con una solución que mejoran iterativamente.
- **Atendiendo a la función objetivo:** Estáticas: no hacen ninguna modificación sobre la función objetivo del problema. Dinámicas: modifican la función objetivo durante la búsqueda.

- **Atendiendo a la vecindad:** Una vecindad: durante la búsqueda utilizan exclusivamente una estructura de vecindad. Varias vecindades: durante la búsqueda modifican la estructura de la vecindad.
- **Atendiendo al uso de memoria:** Sin memoria: se basan exclusivamente en el estado anterior. Con memoria: utilizan una estructura de memoria para recordar la historia pasada.

Generalmente, estas características (pueden incluirse más) se suelen combinar para permitir una clasificación más elaborada.

En la Tabla 1.1 se presenta un resumen de las publicaciones sobre las técnicas más populares en aplicaciones, híbridos, mejoras metodológicas, comparaciones y pruebas. Se tienen en cuenta artículos, conferencias, libros, capítulos de libros de las principales publicaciones del área. Se aprecia el aumento del uso de los métodos meta-heurísticos y la mayor aceptación de algunos métodos, como Algoritmos Genéticos (AG), Recocido Simulado (SA) u Optimización de Enjambre de Partículas (PSO), mientras que otros algoritmos como Programación Evolutiva y Estrategias Evolutivas, a pesar de ser métodos más antiguos, no son tan frecuentemente aplicados. De igual manera, las meta-heurísticas consideradas en la Tabla 1.1 se limitan a las que tienen más acogida según la literatura.

Tabla 1.1 *Número de artículos por técnicas bio-inspiradas (Delgado, 2010)*

Meta-heurística	Números de artículos (metodologías y aplicaciones)			
	antes del 2001	2001 - 2005	2006 - 2010	Total
Algoritmos Genéticos	7285	14723	13850	35858
Recosido Simulado	3490	3287	3388	10165
PSO	7	742	5262	6011
Colonias de Hormigas	47	699	2223	2969
Búsqueda Tabú	465	692	793	2969
Evolución diferencial	93	382	1362	1837
Algoritmos Inmunológicos	65	420	1180	1665
Programación evolutiva	252	382	321	955
Estrategias evolutivas	168	342	620	930
GRASP	235	298	392	925

Algoritmos Meméticos	5	141	396	542
Búsqueda dispersa	21	93	170	284
Hooke and Jeeves Pattern search	113	34	55	202
Quimiotaxis bacteriana	N.A.	1	22	23

Las fuentes de inspiración de las meta-heurísticas se pueden dividir en tres grandes grupos: evolución, procesos físicos y aprendizaje (Esmorís, 2013). Los diferentes métodos desarrollados hasta la actualidad pueden ser clasificados atendiendo las características de estos grupos, ya sea dentro de una de las líneas estrictamente o en una combinación de ellas. En la Figura 1.3 *Meta-heurísticas de acuerdo a su inspiración* se observan las técnicas más representativas de las diferentes categorías.

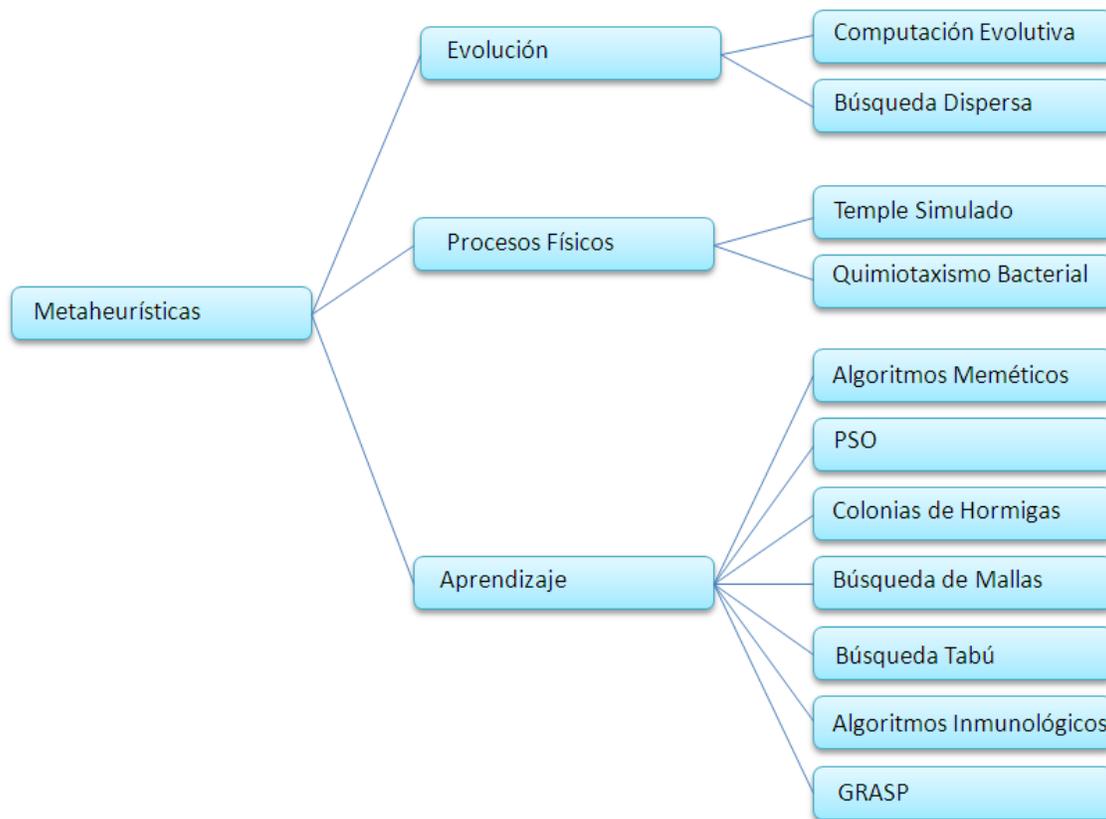


Figura 1.3 *Meta-heurísticas de acuerdo a su inspiración*¹

¹ Tomado de PÉREZ LEÓN, S. R. 2013. *Implementación de un algoritmo basado en Colonias de Hormigas para la optimización de funciones con datos mezclados*. Universidad Central "Marta Abreu" de las Villas.

Meta-heurísticas de trayectoria simple

Los métodos de Búsqueda Local BL o meta-heurísticas de trayectoria simple, definen un proceso de transformación que parte de una solución inicial generada de forma aleatoria o mediante algún otro algoritmo la cual constituye la solución actual. A esta solución se le aplican dicho método para obtener un conjunto de soluciones en un entorno cercano. O sea mediante un proceso iterativo dada una solución actual buscan una mejor solución en su vecindario Figura 1.4 ***Espacio de búsqueda***, dicha mejor solución pasa a ser la solución actual y se repite el proceso hasta que se generan un número máximo de soluciones o se alcanza un óptimo local. Lo dicho puede expresarse de la siguiente forma:

Encontrar una solución x tal que $f(x) = \min f(y)$ para todo $y \in \text{vecindario}(x)$.

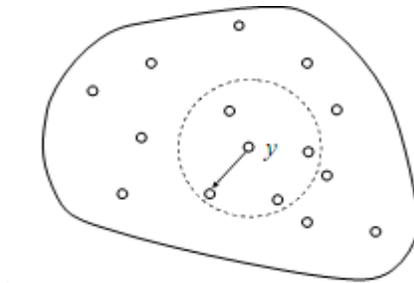


Figura 1.4 *Espacio de búsqueda*

A su vez existen dentro de las propias estrategias de búsqueda local dos subgrupos bien definidos: por un lado los métodos de BL deterministas y por otro los estocásticos. A continuación mostramos algunos representantes de ambos grupos:

Métodos de Búsqueda Local Deterministas:

- **Métodos de Cuasi-Newton** (Luenberguer, 1989): Difieren del método de Newton puro en que no utilizan la matriz Hessiana en sus iteraciones sino que una aproximación de esta. Existen diversas variantes de métodos cuasi newton, por ejemplo los métodos L-BFGS y L-BFGS-B que hacen un uso limitado de la memoria (usa mucha menos memoria que otros algoritmos para el mismo problema); L-BFGS viene de BFGS de memoria limitada. Únicamente necesita la función y su gradiente, pero no la matriz Hessiana. L-BFGS, es capaz de

resolver funciones sin restricciones, mientras que la variante L-BFGS-B puede resolver funciones con restricciones simples (del tipo $l_i < x_i < u_i$, siendo x_i la variable i -ésima y l_i y u_i los límites inferior y superior de esa variable) en sus parámetros.

➤ **Búsqueda Local Guiada:** Acumula sanciones durante una búsqueda. Utiliza algoritmos de sanciones. Cuando el algoritmo de búsqueda local determinado se instala en un óptimo local, GLS modifica la función objetivo con un régimen específico. A continuación, la búsqueda local opera con una función aumentada de objetivo, que está diseñado para llevar a cabo la búsqueda del óptimo local. La clave está en la forma en que la función objetivo se modifica.

Métodos de Búsqueda Local Estocásticos:

➤ **Búsqueda Tabú** (Glover and Laguna, 1997): La búsqueda tabú utiliza un procedimiento de búsqueda local o por vecindades para moverse iterativamente desde una solución x hacia una solución x' en la vecindad de x , hasta satisfacer algún criterio de parada. Para poder explorar regiones del espacio de búsqueda que serían dejadas a un lado por el procedimiento de búsqueda local, la búsqueda tabú modifica la estructura de vecinos para cada solución a medida que la búsqueda progresa. Las soluciones admitidas para $N * (x)$ (vecindad reducida de x) son determinadas mediante el uso de estructuras de memoria. La búsqueda entonces progresa moviéndose iterativamente de una solución x hacia una solución x' en $N * (x)$. Quizás la estructura de memoria más importante usada para determinar las soluciones permitidas a un $N * (x)$, sea la lista tabú. En su forma más simple, una lista tabú es una memoria de corto plazo que contiene las soluciones que fueron visitadas en el pasado reciente (menos de n iteraciones atrás, donde n es el número de soluciones previas que van a ser almacenadas. La búsqueda tabú excluye las soluciones en la lista tabú de $N * (x)$.

- **Método Solis West** (Solis and Wets, 1981): Sigue un esquema de ascensión de colinas con un tamaño de salto adaptativo. Para cada iteración partiendo de una solución actual x . Se genera un valor d mediante una distribución normal con 0 de media y ρ de desviación estándar. La nueva solución se obtiene sumando dicho valor d junto con un valor *bias* que mantiene un cierto grado de “inercia” sobre los movimientos anteriores.
- **Método Simplex de Nelder y Mead** (Spendley et al., 1962): El método utiliza el concepto de un simplex, que es un polítopo de $N+1$ vértices en N dimensiones: un segmento de línea en una línea, un triángulo en un plano, un tetraedro en un espacio tridimensional y así sucesivamente. El método busca de modo aproximado una solución óptima local a un problema con N variables.

Un inconveniente de la BL está indicado por su propio nombre pues se orienta a alcanzar óptimos locales y no lo deseado realmente: encontrar la mejor solución en todo el espacio de búsqueda o sea el óptimo global, pues converge de forma rápida a los óptimos locales los cuales pueden ser soluciones muy alejados del óptimo global. Este comportamiento es denominado convergencia prematura.

Meta-heurísticas poblacionales

Las meta-heurísticas basadas en población, o meta-heurísticas poblacionales, son aquellas que emplean un conjunto de soluciones (población) en cada iteración del algoritmo, en lugar de utilizar una única solución como las meta-heurísticas del grupo anterior. Estas proporcionan de forma intrínseca un mecanismo de exploración paralelo del espacio de soluciones, y su eficacia depende en gran medida de cómo se manipule dicha población. Dentro de esta clasificación se destacan los Algoritmos Evolutivos y los algoritmos basados en Inteligencia Colectiva.

Ambos tienen en común el haber sido inspirados en algún proceso natural, en el primero de los casos en la teoría de la evolución de Darwin (Darwin, 1859), quien planteó que la evolución de las especies se produce por tres conceptos: replicación, variación y selección natural mientras que los algoritmos basados en Inteligencia Colectiva toman su inspiración

en ejemplos biológicos de comportamiento colectivo (enjambres) como es el caso de las colonias de insectos, las bandadas de aves y los cardúmenes de peces.

Los Algoritmos Evolutivos son métodos de optimización y búsqueda de soluciones basados en los postulados de la evolución biológica. En ellos se mantiene un conjunto de entidades que representan posibles soluciones, las cuales se mezclan, y compiten entre sí, de tal manera que las más aptas son capaces de prevalecer a lo largo del tiempo, evolucionando hacia mejores soluciones cada vez.

Numerosos han sido los modelos de AEs propuestos. A continuación mostramos algunos de los más destacados:

- **Algoritmos Genéticos (AGs)** (Goldberg, 1998): Su principal mecanismo empleado para generar nuevos individuos es mediante un operador de recombinación (denominado cruce) que combina información de diferentes individuos para producir los nuevos individuos. Dentro de los AGs, el operador de mutación es empleado, por contra, con una probabilidad baja, simplemente como un operador capaz de aumentar la diversidad en la población.
- **Búsqueda Dispersa** (Laguna and Martí, 2003): En este modelo se mantienen dos subpoblaciones, una élite, de las mejores soluciones encontradas (para explotar las soluciones encontradas), y otra formada por individuos alejados de los individuos de la población anterior (para mantener cierta diversidad en el proceso de combinación). Además, es un modelo que considera el uso de un proceso de mejora local, para explotar más las soluciones encontradas.
- **Evolución Diferencial** (Storn and Price, 1997): Se caracteriza en que, dado que las soluciones se representan mediante vectores reales, las poblaciones de soluciones son modificadas de forma sucesiva mediante operadores de suma y resta vectorial, así como operaciones de intercambio de valores hasta que la población converge.
- **Algoritmos Basados en estimación de Distribuciones** (Estimation of Distribution Algorithms, EDA) (Lozano et al., 2006): Se caracterizan en que no utilizan ni los operadores de cruce ni los de mutación para generar nuevas soluciones. Se genera la nueva población mediante muestreo de una

distribución de probabilidad, la cual es estimada a partir de los mejores individuos de las generaciones anteriores.

Por otra parte, la principal característica de los algoritmos basados en la **Inteligencia Colectiva** viene determinada por la estrecha colaboración social que presentan a través del sistema de comunicación que surge entre los individuos del grupo. Esta comunicación, a su vez, puede aparecer de forma directa o indirecta. La comunicación indirecta ocurre cuando un individuo altera el medio en que se desarrollan y los otros son capaces de captar estos cambios. La comunicación directa es aquella que ocurre a través de la obtención de la ubicación de otros individuos mediante sonido, visibilidad u otra forma directa de interacción. Entre los algoritmos más significativos y estudiados de Inteligencia Colectiva están:

- **Sistemas de Partículas** (*Particle Swarm Optimization*, PSO) (Kennedy and Eberhart, 1995): El PSO se caracteriza en que es un modelo inspirado en modelos naturales de movimiento como las bandadas de pájaros. En estos, la población está compuesta de una serie de partículas que representan las soluciones, y la evolución de la población se realiza debido al desplazamiento en el espacio de búsqueda de dichas partículas. Dicho desplazamiento está principalmente formado por dos direcciones: Una hacia la mejor solución global encontrada; y la otra hacia la mejor solución encontrada por dicha partícula.
- **Optimización basada en colonia de hormigas** (*Ant Colony Optimization*; ACO) (Dorigo and Caro, 1999). Son métodos poblacionales inspirados en el comportamiento de las hormigas naturales. Los mismos realizan un proceso constructivo y estocástico guiado por unos rastros de feromona que van depositando cada hormiga, dando una medida de cuán deseado ha sido un determinado camino, y a través de una función de visibilidad que evalúa la calidad del desplazamiento.

Otro modelo poblacional que fue recientemente desarrollado por el grupo de Inteligencia Artificial del departamento de computación de la Universidad de las Villas y será la base de nuestra investigación es la llamada Optimización basada en Mallas Variables (VMO, de su nombre en inglés *Variable Mesh Optimization*).

Esta meta-heurística propone formas de búsqueda que combinan en un mismo proceso niveles variables de intensificación y exploración. La esencia de este modelo es crear una malla de puntos en el espacio m dimensional, donde se realiza el proceso de optimización de una función $FO(x_1, x_2, \dots, x_m)$; la cual se mueve mediante un operador de expansión hacia otras regiones del espacio de búsqueda. Dicha malla se hace más “fina” en aquellas zonas que parecen ser más promisorias. Es variable en el sentido que la malla cambia su tamaño (cantidad de nodos) y configuración durante el proceso de búsqueda.

Existe otro operador en la búsqueda desarrollada por este modelo que se denomina contracción de la malla y se encarga de seleccionar los puntos más representativos de cada zona explorada del espacio de búsqueda.

Ejemplos de uso de las meta-heurísticas en problemas reales

Diversos son los problema reales en los que las meta-heurísticas han desempeñado un importante papel en la solución de los mismos. Algunas de las áreas en las cuales se ha aplicado con éxito estos algoritmos son: la logística portuaria, la planificación de frecuencias en las telecomunicaciones y la programación de la tripulación de las empresas de transporte público de autobuses, entre otros muchos ejemplos.

Meta-heurísticas en la Logística Portuaria

Las terminales portuarias de contenedores compiten por un mercado que anualmente mueve varios cientos de millones de TEUs (acrónimo del término en inglés *Twenty-foot Equivalent Unit*, que significa Unidad Equivalente a Veinte Pies) alrededor del mundo.

Se trata de un mercado altamente competitivo en el que las grandes operadoras escogen como base de operaciones los puertos de países política y socialmente estables, bien ubicados y con buena relación coste/calidad. Las terminales se enfrentan a amenazas constantes por parte de competidores actuales y futuros.

Entre las respuestas a estas amenazas destaca la mejora en la gestión logística de la terminal por su incidencia directa en los precios y en la calidad del servicio. En este sentido, los responsables de la terminal necesitan sistemas de gestión y control que les permitan adoptar

decisiones adecuadas con las que hacer frente a las amenazas. La gran mayoría de los problemas que surgen en una terminal portuaria de contenedores pertenecen a la clase de los problemas NP-duros, por lo que deben considerarse procedimientos heurísticos o meta-heurísticos para su resolución en situaciones reales (Batista and Moreno-Vega, 2013).

Las redes de telecomunicaciones y las meta-heurísticas

La industria de las telecomunicaciones ha proporcionado, y sigue proporcionando, una gran cantidad de problemas de optimización que surgen desde el propio diseño del sistema de comunicación hasta algunos aspectos de su funcionamiento. La resolución de estos problemas ha jugado, sin lugar a dudas, un papel muy destacado en el desarrollo y utilización de este tipo de sistemas. No obstante, a medida que se han ido haciendo más populares y su penetración en el mercado es mayor, el tamaño de los sistemas de telecomunicaciones ha ido creciendo y, por tanto, los problemas que plantean tienen una dimensión tan elevada que los hacen inabordables con técnicas exactas. Los algoritmos meta-heurísticos son una de las mejores opciones en este contexto, ya que son capaces de encontrar soluciones de calidad en tiempos aceptables.

Algunos de los problemas abordados en esta área son la planificación de celdas, la asignación automática de frecuencias y la optimización del proceso de difusión en redes *ad hoc* de dispositivos móviles. Debido a las características de estos problemas, se han desarrollado meta-heurísticas avanzadas que abarcan dos ámbitos: optimización multiobjetivo y sistemas de computación grid. El primer caso está motivado por la naturaleza multiobjetivo de dos de los problemas abordados (planificación de celdas y difusión óptima), en los que existen varias funciones contrapuestas que se han de optimizar a la vez. El segundo caso viene dado por la complejidad y dimensión de los problemas planteados, por lo que se han desarrollado modelos que permiten su ejecución en sistemas de computación grid, lo que nos va a permitir, no sólo reducir el tiempo de ejecución de los algoritmos, sino también obtener modelos de búsqueda que permitan resolver estos problemas de forma más efectiva (Valero, 2008).

Manejo de empresas de transporte público de autobuses

Los problemas de programación de la tripulación son bien conocidos, y se han propuesto varias técnicas para resolverlos, en particular, utilizando la formulación puesta a cubrir un solo objetivo. Sin embargo, en la práctica, existe la necesidad de considerar múltiples objetivos, algunos de ellos en conflicto unos con otros; por ejemplo, la calidad y el costo del servicio, lo que implica también que los métodos de solución alternativa han de ser desarrollados.

Lourenço en su trabajo (Lourenço et al., 2001) propone meta-heurísticas multiobjetivo basado en la búsqueda tabú y algoritmos genéticos para dar solución a este tipo de problemas. Estas meta-heurísticas también presentan algunas de las características de innovación relacionados con la estructura del problema de programación de la tripulación que guían la búsqueda de manera eficiente y les permita encontrar buenas soluciones. Algunas de estas nuevas características también se pueden aplicar para el desarrollo de la heurística para otros problemas de optimización combinatoria. En la actualidad estos métodos se han incorporado con éxito en la planificación de sistemas de transporte con GIST y están realmente utilizado por varias compañías(Lourenço et al., 2001).

Conclusiones parciales

De lo expuesto anteriormente se puede concluir que los métodos exactos no son la variante más adecuada para la solución de problemas complejos de optimización. Ello es debido a que estos métodos son dependientes del problema que resuelven. Además la cardinalidad del espacio de búsqueda de problemas complejos suele ser muy grande, lo cual hace inviable el uso de algoritmos exactos, ya que la cantidad de tiempo que necesitaría para encontrar una solución es inaceptable. Por otro lado, las meta-heurísticas de trayectoria simple dependen en gran medida de punto inicial y tienden a quedar estancadas rápidamente en óptimos locales. Por tanto para la presente investigación haremos uso de meta-heurísticas poblacionales ya que estas nos proporcionan un mecanismo natural de exploración en todo el espacio de soluciones y fomentan la diversidad durante todo el proceso de búsqueda.

CAPÍTULO 2 ALGORITMO VMO CON TOPOLOGÍA

LOCAL

En presente capítulo se describe la implementación del algoritmo para resolver problemas de optimización continua sin restricciones. Este tipo de problemas de optimización que clásicamente se formaliza como:

$$\min/\max f(x), x = (x_1, x_2, x_3, \dots, x_n)^T \in R^n \quad (2.1)$$

donde x_i es la variable de decisión definida en el dominio $L_i < x_i < U_i$. Los términos L_i y U_i son los límites inferiores y superiores, respectivamente, que son definidos a priori. En concreto, en este trabajo, nos centramos en las funciones multimodales (véase para una mejor descripción (Suganthan et al., 2005a)), que son más difíciles que los típicos unimodales.

Como ha sido mencionado anteriormente, la meta heurística que utilizada en este trabajo es Optimización basada en Mallas Variables (VMO, de su nombre en inglés *Variable Mesh Optimization*) (Puris et al., 2012).

Optimización Basada en Mallas Variables

La Optimización Basada en Mallas Variables (*Variable Mesh Optimization; VMO*) es una meta heurística poblacional con características evolutivas donde un conjunto de nodos que representan soluciones potenciales a un problema de optimización, forman una malla (población) que dinámicamente crece y se desplaza por el espacio de búsqueda (evoluciona). Para ello, se realiza un proceso de expansión en cada ciclo, donde se generan nuevos nodos en dirección a los extremos locales (nodos de la malla con mejor calidad en distintas vecindades) y el extremo global (nodo obtenido de mejor calidad en todo el

proceso desarrollado); así como a partir de los nodos fronteras de la malla. Luego se realiza un proceso de contracción de la malla, donde los mejores nodos resultantes en cada iteración son seleccionados como malla inicial para la iteración siguiente. La formulación general de la meta-heurística abarca tanto los problemas de optimización continuos como los discretos, en nuestro caso realizaremos énfasis en los problemas continuos. En (Puris, 2009) se da una descripción detallada de la misma.

Descripción general de la meta heurística

La esencia del método VMO es crear una malla de nodos en el espacio m -dimensional, donde se realiza el proceso de optimización de una función $FO(x_1, x_2, \dots, x_m)$; la cual se mueve mediante un proceso de expansión hacia otras regiones del espacio de búsqueda. Dicha malla se hace más “fina” en aquellas zonas que parecen ser más promisorias. Es variable en el sentido que la malla cambia su tamaño (cantidad de nodos) y configuración durante el proceso de búsqueda. Los nodos se representan como vectores de la forma $n(x_1, x_2, \dots, x_m)$.

El proceso de generación de nodos en cada ciclo comprende los pasos siguientes:

- Generación de la malla inicial.
- Generación de nodos en dirección a los extremos locales.
- Generación de nodos en dirección al extremo global.
- Generación de nodos a partir de las fronteras de la malla.

El método incluye los parámetros:

- Cantidad de nodos de la malla inicial (S_{IM}).
- Cantidad máxima de nodos de la malla en cada ciclo (S_{TM} , donde $3 \cdot S_{IM} \leq S_{TM}$).
- Tamaño de la vecindad (k).
- Condición de parada (M).

Proceso de expansión

El algoritmo realiza un proceso de expansión mediante el cual la población se traslada a través del espacio de soluciones. Esta operación se realiza siguiendo los pasos descritos a continuación (Puris et al., 2012).

Paso 1: Generación aleatoria de la malla inicial: Para cada uno de los nodos n_i de la malla inicial se valoriza aleatoriamente cada dimensión con un valor real entre el intervalo definido en cada caso.

Paso 2: Generación de nodos en dirección a los extremos locales: El primer tipo de exploración del modelo se basa en llevar a cabo un análisis de las vecindades para cada uno de los nodos de la malla. La vecindad de n_i esta definida por los k vecinos más cercanos a este (teniendo en cuenta la distancia). El mejor nodo dentro de esta vecindad es seleccionado como extremo local n_i^* . Únicamente es generado un nodo entre n_i y n_i^* si n_i^* posee mejor valor de la función objetivo que n_i . Para calcular los vecinos más cercanos de cada nodo de la malla se utiliza como función de distancia la euclidiana, definida por (2.1):

$$D_{euclidiana}(n_1, n_2) = \sqrt{\sum_{i=1}^M (v_i^1 - v_i^2)^2} \quad (2.1)$$

Los nuevos nodos (n_z) son calculados usando la función definida por (2.2):

$$n_z = F(n_i, n_i^*, Pr(n_i, n_i^*)) \quad (2.2)$$

donde Pr es el factor de cercanía y representa la relación entre la aptitud del nodo actual y su extremo local. Este factor se calcula por la ecuación (2.3). Se toma un valor en el intervalo $[0, 1]$, más grande cuando tiene una mejor calidad n_i :

$$Pr(n_i, n_i^*) = \frac{1}{1 + |fitness(n_i) - fitness(n_i^*)|} \quad (2.3)$$

La función F puede ser descrita de diferentes maneras. En este trabajo la componente v_j del nodo n_z será calculada por (2.4):

$$v_j^z = \begin{cases} \bar{m}_j & \text{if } |\bar{m}_j - v_j^{i*}| > \xi_j \text{ and } U[0,1] \leq Pr(n_i, n_i^*) \\ v_j^{i*} + U[-\xi_j, \xi_j] & \text{if } |\bar{m}_j - v_j^{i*}| \leq \xi_j \\ U[v_j^i, \bar{m}_j] & \text{othercase} \end{cases} \quad (2.4)$$

donde $\bar{m} = average(v_j^i, v_j^{i*})$, $U[x, y]$ denota un valor aleatorio (uniforme) en el intervalo $[x, y]$, y ξ_j define el distancia mínima permitida para cada componente. Su valor disminuye durante el funcionamiento del algoritmo, calculado por (2.5):

$$\xi_j = \begin{cases} \frac{rango(a_j, b_j)}{4} & \text{si } 0.15\% C \\ \frac{rango(a_j, b_j)}{8} & \text{si } 0.15\% C \leq c < 0.3\% C \\ \frac{rango(a_j, b_j)}{16} & \text{si } 0.3\% C \leq c < 0.6\% C \\ \frac{rango(a_j, b_j)}{50} & \text{si } 0.6\% C \leq c < 0.8\% C \\ \frac{rango(a_j, b_j)}{100} & c \geq 0.8\% C \end{cases} \quad (2.5)$$

donde C y c denotan el número máximo de evaluaciones en la función objetivo permitidas y el número actual de evaluaciones respectivamente. Además, el $rango(a_j, b_j)$ denota el dominio de amplitud (a_j, b_j) para cada componente.

La función F se comporta de la siguiente manera: en el primer caso, el valor medio entre el nodo actual y el extremo local es obtenido por la j -ésima componente. En segundo caso, la vecindad del extremo local se desplaza en función de un valor de distancia para la iteración actual. En último caso, es generado un número aleatorio entre el valor medio y el nodo actual.

Paso 3: Generación de nodos en dirección al extremo global: Con el objetivo de acelerar la convergencia se realiza un exploración en dirección al nodo con mejor valor de la función objetivo perteneciente a la población actual denominado extremo global n_g . Los nuevos nodos en dirección a n_g son generados usando al ecuación (2.6):

$$n_g^*(x_j) = G(n_i n_g, Pr(n_i n_g)) \quad (2.6)$$

donde Pr es el factor de cercanía entre el fitness del nodo actual y el extremo. Toma valores entre 0 y 1 y se calcula mediante la ecuación(2.7):

$$\Pr(n_i n_g) = \frac{1}{1 + |fitness(n_i) - fitness(n_g)|} \quad (2.7)$$

La función G es representada de la siguiente manera (2.8):

$$v_j^x = \begin{cases} ave(v_j^i, v_j^g), & \text{si } U[0,1] \leq \Pr(n_i n_g) \\ U[ave(v_j^i, v_j^g), v_j^g] & \text{en otro caso} \end{cases} \quad (2.8)$$

Paso 4: Generación de nodos a partir de los nodos más externos de la malla: Con este paso se completa el proceso de expansión. Solo es aplicado si el total de nodos creados en los pasos anteriores es menor que T.

En el paso 4 son considerados los nodos más internos n_u y externos n_s de la malla que conforman la frontera interior y exterior respectivamente a partir del punto que representa el centro del espacio de soluciones. Para detectar esos nodos es empleada la distancia euclidiana, los nodos de mayor distancia conformarán n_s y los de menor distancia n_u . Partiendo de esos conjuntos, nuevos nodos son creados (uno por cada nodo de la frontera) empleando las ecuaciones 2.9 y 2.10

$$v_j^h = \begin{cases} v_j^s + w_j & \text{si } v_j^s > 0 \\ v_j^s - w_j & \text{si } v_j^s < 0 \end{cases} \quad (2.9)$$

$$v_j^h = \begin{cases} |v_j^u + w_j| & \text{si } v_j^u > 0 \\ |v_j^u - w_j| & \text{si } v_j^u \geq 0 \end{cases} \quad (2.10)$$

donde w_j representa un desplazamiento para cada una de las componentes y se calcula de manera decreciente de acuerdo con la ecuación (2.11):

$$w_j = (w_j^0 - w_j^1) * \frac{C-c}{C} + w_j^1 \quad (2.11)$$

C y c denotan un máximo de evaluaciones permitidas de la función objetivo y el número de la evaluación actual respectivamente, por otra parte w_j^0 representa el desplazamiento inicial y w_j^1 su valor final. Para conseguir el decrecimiento $w_j^0 > w_j^1$ sus valores son calculados como $w_j^0 = rango(a_j, b_j)/10$ y $w_j^1 = rango(a_j, b_j)/100$. Donde $rango(a_j, b_j)$ representa la amplitud del dominio de (a_j, b_j) .

Proceso de contracción

Con la operación de contracción son seleccionados los nodos de la población que serán empleados en la siguiente iteración del modelo. Nodos con mejor valor de la función objetivo son seleccionados entre los pertenecientes a la malla actual y los nuevos nodos creados en el proceso de expansión. Luego de la selección un método para incrementar la diversidad poblacional es aplicado para preservar un mínimo de distancia entre los nodos de la malla. Para conseguir esto un operador de limpieza adaptativo es propuesto. Seguidamente se brinda una descripción más detallada de dicho proceso:

Paso 5: Los nodos de la malla son ordenados de acuerdo con el valor de su *fitness* (ascendente).

Paso 6: La diferencia entre cada nodo y su sucesor es calculada. Sucesores con una diferencia menor el ζ calculado, son eliminados de la malla. ζ es calculado por la ecuación 2.12:

$$\zeta = \begin{cases} \frac{\text{rango}(a_j, b_j)}{4} & \text{si } 0.15\% C \\ \frac{\text{rango}(a_j, b_j)}{8} & \text{si } 0.15\% C \leq c < 0.3\% C \\ \frac{\text{rango}(a_j, b_j)}{16} & \text{si } 0.3\% C \leq c < 0.6\% C \\ \frac{\text{rango}(a_j, b_j)}{50} & \text{si } 0.6\% C \leq c < 0.8\% C \\ \frac{\text{rango}(a_j, b_j)}{100} & c \geq 0.8\% C \end{cases} \quad (2.12)$$

Paso 7: Los nodos con mejor *fitness* son seleccionados para la siguiente iteración del algoritmo. Si el número de nodos necesarios para la realización de una nueva iteración de la malla no es alcanzado, esta se completa con nodos generados de manera aleatoria.

A continuación se presenta un esquema que muestra la estructura general del algoritmo tomando como condición de parada un número máximo de evaluaciones de la función objetivo Figura 2.1 *Esquema general de modelo VMO:*

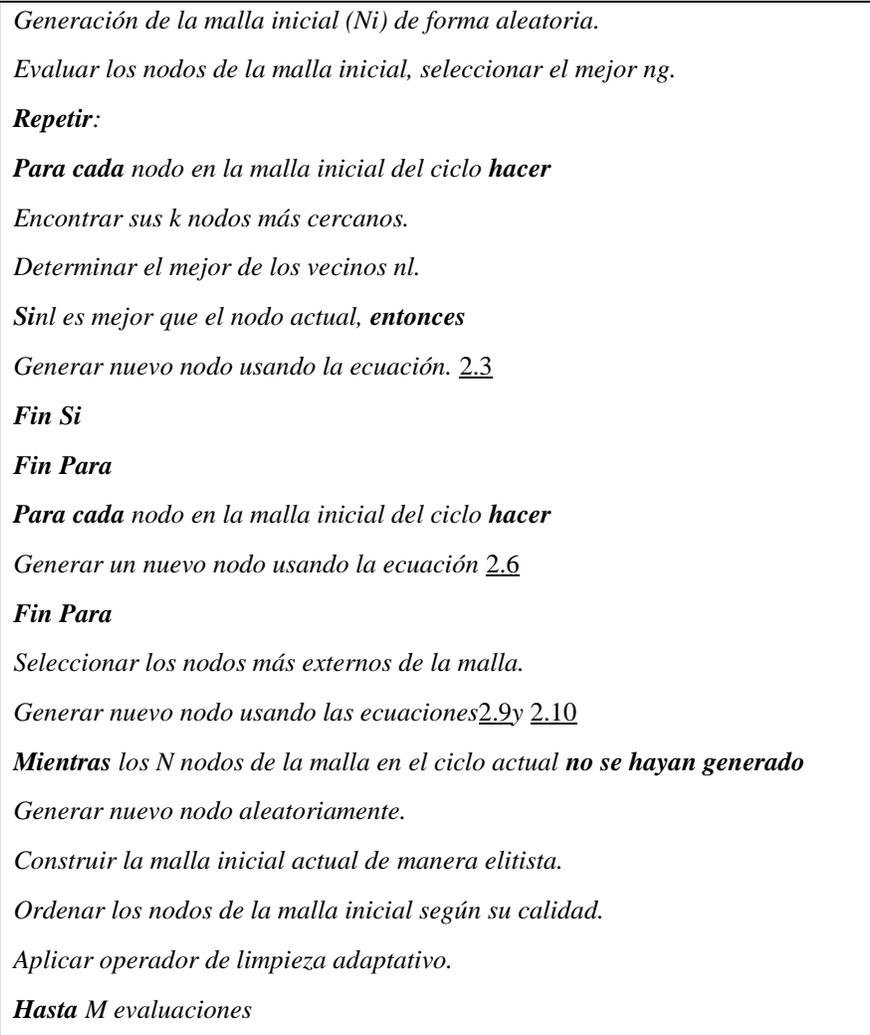


Figura 2.1 Esquema general de modelo VMO

Topología de vecindad en PSO

Un aspecto necesario e importante a la hora de implementar un PSO es la definición de una *topología de vecindad o red social* para todos los miembros del enjambre. Esto es, el subconjunto de partículas con las cuales cada partícula podrá interactuar durante el proceso de búsqueda.

Formalmente, para cada partícula i debe definirse un V_i con $V_i \subset S$, el cual establece que la partícula i podrá comunicarse (intercambiar información) con cualquier partícula $j \in V_i$ mediante la ecuación (2.13). Respecto a la inclusión o exclusión de una partícula en su

vecindad se han realizado pruebas que muestran que no existe gran diferencia en el comportamiento del enjambre si se incluye o no (Poli et al., 2007).

$$gbesti = \min_{pbest_j} \{f(pbest_j)\}, \quad j \in V_i \quad (2.13)$$

De esta ecuación se desprende que, si dos partículas i y j comparten la misma vecindad, esto es $V_i = V_j$, entonces $gbesti = gbest_j$, lo cual significa que estarán siendo atraídas por la misma partícula líder, lo cual podría implicar que terminen explorando la misma región del espacio.

La primera topología de vecindad definida para PSO tenía en cuenta la Distancia Euclidiana, imitando el comportamiento de los modelos biológicos en los que un individuo solo es capaz de comunicarse con otros individuos en una vecindad inmediata (Shi et al., 2007). Esta definición de vecindad exigía el establecimiento de un umbral de cercanía e implicaba un alto costo computacional debido al cálculo de las distancias entre todas las partículas en cada iteración, de ahí que comenzaron a utilizarse topologías de vecindades independientes de las posiciones de las partículas.

La primera topología de este tipo propuesta y una de las más utilizadas en la actualidad fue la topología global. Conocida también como modelo *gbest*, esta define que todas las partículas están conectadas entre sí, o sea, $V_i = S (\forall i \in S)$, lo cual significa que cada partícula puede transmitir su experiencia a todas las demás y beneficiarse de la de cada una de ellas también. En el mismo año en que se publica la versión original de PSO (Shi et al., 2007), sus autores proponen una variante del algoritmo (Bharat et al., 2009) en la que se usa una *topología local*. También conocida como modelo *lbest*, la topología local se refiere a cualquier red de comunicación no global en la que cada partícula intercambia información con un subconjunto propio de todo el enjambre, o sea, $V_i \subset S (\forall i \in S)$.

La forma más común del modelo local es la *topología de anillo*, en la que el enjambre se organiza en formación circular y cada partícula se conecta con las partículas adyacentes. Una manera de definir esta topología para un número variable de vecinos es mediante la fórmula:

$$V_i = \cup_{j=i-m}^{i+m} I(j) \quad m = \left\lfloor \frac{n}{2} \right\rfloor \quad (2.14)$$

con

$$I(j) = \begin{cases} |S| - j & \text{si } j < 1 \\ j & \text{si } 1 \leq j \leq |S| \\ j - |S| & \text{si } j > |S| \end{cases} \quad (2.15)$$

donde n ($n \leq |S|$) es la cantidad de partículas vecinas y $|S|$ el tamaño del enjambre. Nótese que cada partícula pertenece a su vecindad, así como las m partículas adyacentes a ambos lados. Nótese, además, que esta fórmula define vecindades simétricas, o sea, cada partícula posee igual cantidad de partículas vecinas de un lado y del otro (Rada-Vilela et al., 2011).

En la Figura 2.2 (a) *Topología de estrella (gbest)* (b) *Topología de anillo (lbest)* se muestra el modelo *lbest* para $n = 2$ y el modelo *gbest* que puede verse como un caso particular del anterior con $n = |S|$. Las partículas han sido representadas por círculos y las conexiones entre ellas mediante arcos y rectas.

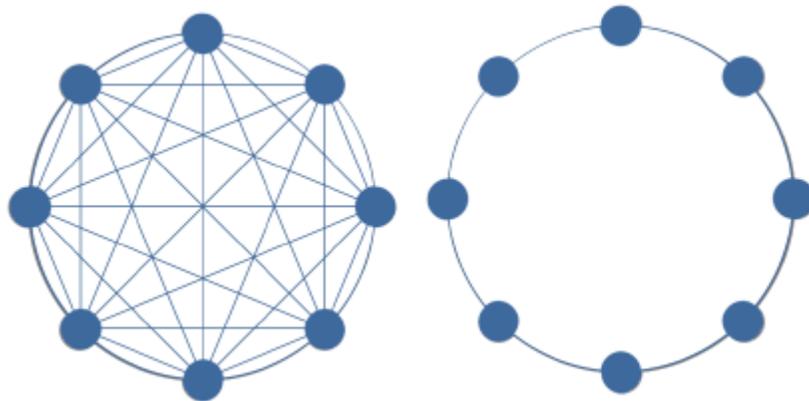


Figura 2.2 (a) *Topología de estrella (gbest)* (b) *Topología de anillo (lbest)*

Aunque se han definido otras topologías de vecindades (Engelbrecht, 2007) (Poli et al., 2007) los modelos más utilizados en la actualidad son el modelo global y el modelo local en su variante de anillo con $n=2$.

Modelo gbest y lbest

Existe una relación directa entre el tamaño de las vecindades de las partículas y la velocidad de convergencia del enjambre. Para grandes vecindades la mejor posición encontrada será usada por mayor cantidad de partículas para actualizar sus posiciones, de modo que la variación del enjambre de una iteración a otra será mucho mayor. En cambio,

con vecindades pequeñas, la mejor posición encontrada se propagará más lentamente y el cambio en las posiciones de las partículas será de menor magnitud otro (Rada-Vilela et al., 2011).

Con respecto a la diversidad dentro del enjambre la relación es inversa. Dado que la mejor posición encontrada se propaga más rápido en vecindades grandes (ej. modelo global), mayor cantidad de partículas serán atraídas hacia ella, disminuyendo la diversidad del enjambre rápidamente de una iteración a otra. Con vecindades pequeñas (ej. modelo local), sin embargo, las partículas serán atraídas por las diferentes mejores posiciones correspondientes a cada una de las vecindades, de modo que el enjambre puede realizar una búsqueda en paralelo. Esto implica una estrategia de búsqueda más exhaustiva (Poli et al., 2007) y una convergencia más lenta pero con mayor diversidad otro (Rada-Vilela et al., 2011).

De estas relaciones se puede establecer que el modelo global converge más rápido, pero es más sensible al estancamiento en óptimos locales. Por su parte, el modelo local tiene más posibilidades de escapar de la región de un óptimo local pero al costo de una menor velocidad de convergencia (Rada-Vilela et al., 2011). En una comparación realizada entre estos dos modelos (Bharat et al., 2009), sobre un conjunto variado de funciones de prueba conocidas, se muestra que en iteraciones tempranas el rendimiento del modelo global supera al del modelo local para mayor cantidad de funciones, pero que luego de pasado cierto número de iteraciones, los resultados del modelo local superan a los del modelo global, especialmente en funciones multimodales.

En general, se plantea que la selección de la topología a utilizar es dependiente del problema a resolver y del número de iteraciones que se puedan emplear para optimizar. En problemas más simples como los unimodales, la utilización del modelo global puede ser conveniente, pues no existe el riesgo de una convergencia prematura en óptimos locales. Sin embargo, en problemas más complejos como los multimodales o multiobjetivos, el beneficio puede ser mayor con el empleo del modelo local y mayor número de iteraciones que permita garantizar una mayor exploración del espacio de búsqueda.

Teniendo en cuenta que no existe una topología idónea para todo tipo de problemas en (Bratton and Kennedy, 2007) se sugiere la utilización de estos dos modelos de vecindades en caso de que se desee desarrollar una investigación rigurosa.

Implementación del VMO con topologías

El VMO básico suele incurrir en óptimos locales. Esta convergencia prematura puede evitarse ignorando la mejor posición global \mathbf{g} conocida, y atendiendo en su lugar a la mejor posición \mathbf{l} conocida de la subpoblación "circundante" al nodo analizado. Esta subpoblación puede definirse geoméricamente por ej. "los m nodos más cercanas"– o bien de forma social, es decir, como un conjunto de nodos relacionados, con independencia de la distancia que las separa. Si suponemos que existe un vínculo de información entre cada nodo y sus adyacentes, el conjunto de estos vínculos constituye un grafo, una red de comunicación, denominada topología. En nuestra investigación la implementación de las subpoblaciones estará definida por un nodo representativo y los nodos más cercanos a este, dicha cercanía estará dada por un radio que será definido por la ecuación (2.16):

$$r_t = \frac{x_{sup} - x_{inf}}{F * \sqrt[n]{|C_t|}} \quad (2.16)$$

Donde x_{sup} y x_{inf} son, respectivamente, los conocidos límites superior e inferior del dominio de la variable. La dimensión de este problema de optimización se denota por el término n definido por el usuario, y $|C_t|$ representa el número actual de subpoblaciones (inicialmente, $|C_t| = \text{número máximo de subpoblaciones}$). Dependiendo del valor r_t , una solución generada por el modelo VMO podría pertenecer a varios grupos. Esta coincidencia se evita cuando cada solución se inserta en la subpoblación con mínimo valor entre su centro y la solución analizada. Debido al particular comportamiento evolutivo que presenta el modelo VMO en la generación de soluciones, el parámetro F definido por el usuario depende del número de evaluaciones alcanzado (*currentevaluation*). Más específicamente, se ajusta este parámetro en función de (2.17):

$$F = \begin{cases} 6 & \text{si } 0 < CE \leq C/4 \\ 10 & \text{si } C/4 < CE \leq C/2 \\ 20 & \text{si } C/2 < CE \leq 2C/3 \\ 50 & \text{si } 2C/3 < CE \leq 3C/4 \\ 100 & \text{si } 3C/4 < CE \leq C \end{cases} \quad (2.17)$$

De esta forma el proceso de agrupamiento de los nodos en sus respectivas subpoblaciones será realizado de la siguiente manera:

1. Buscar los centros para cada subpoblación.
2. Analizar cada nodo e incluirlo a la subpoblación a la cual pertenezca.

A continuación se presenta un esquema que muestra el proceso de agrupamiento de los nodos en la subpoblaciones Figura 2.3 **Esquema de generación de subpoblaciones**:

```

Creación de subpoblaciones
Inicio{
  {meta heurística (VMO)}
  Para cada nodo de la malla hacer
    Si nodoActual no es centro y no pertenece a ninguna subpob hacer
      Buscar el centro más cercano dentro del radio r
      Si se encuentra un centro i hacer
        nodoActual incluido a subpob con centro i
      sino
        Si # actual de subpob es menor que # máx. de subpob hacer
          nodoActual se convierte en el centro de una nueva subpob
        sino
          Buscar el centro más cercano i
          Incluir nodoActual a subpob con centro i
        Fin Si
      Fin Si
    Fin Para
  }Fin

```

Figura 2.3 *Esquema de generación de subpoblaciones***Actualización de las subpoblaciones**

Es necesario aclarar que en el proceso de contracción de la malla es aplicado, como se ha mencionado antes, un operador de limpieza para garantizar la diversidad. Al ser aplicado dicho operador se corre el riesgo de perderse información referente al centro de una o varias subpoblaciones, para evitar que esto suceda primeramente se comprueba si realmente el nodo que será eliminado es centro o nodo representativo de una subpoblación, de ser así una función llamada *setHuerfanos()* se encarga de buscar los nodos pertenecientes a la subpoblación que tiene como centro al nodo en cuestión y al hacerlo indica a cada nodo que está quedando huérfano, después de realizado lo anterior, otra función llamada *actualizarSubPob()* es la encargada de buscar entre los nodos de una subpoblación que ha perdido su centro un nuevo nodo que la represente. Este proceso se repite cada vez que un centro sea eliminado en el proceso de limpieza.

Después de realizar el agrupamiento de todos los nodos en sus respectivas subpoblaciones en cada una de estas serán aplicados los pasos del 2 al 4 explicados en el epígrafe 2.1.2 de esta memoria, cada subpoblación será tratada como una malla distinta durante el proceso de expansión lo cual posibilitara la exploración, en diferentes zonas, del espacio de búsqueda, de manera simultánea, lográndose, de esta manera, una mayor diversidad en las soluciones generadas. En la siguiente Figura 2.4 *Esquema general de la topología estructurada* se muestra el esquema general de VMO con topología estructurada.

Generación de la malla inicial (N_i) de forma aleatoria.

Repetir:

Creación de subpoblaciones

Para cada subpoblación hacer

Para cada nodo n de la subpob i hacer

Encontrar sus k nodos más cercanos.

Determinar el mejor de los vecinos n_l .

Si n_l es mejor que el nodo actual, entonces

Generar nuevo nodo usando la ecuación. 1.3

Fin Si

Fin Para

Evaluar los nodos de la subpob i , seleccionar el mejor ng

Para cada nodo en la subpob i hacer

Generar un nuevo nodo usando la ecuación 1.6

Fin Para

Seleccionar los nodos más externos de la subpob i .

Generar nuevo nodo en dirección a los externos

Fin Para

Mientras los N nodos de la malla en el ciclo actual no se hayan generado

Generar nuevo nodo aleatoriamente.

Construir la malla inicial actual de manera elitista.

Ordenar los nodos de la malla inicial según su calidad.

Aplicar operador de limpieza adaptativo.

Hasta M evaluaciones

Figura 2.4 *Esquema general de la topología estructurada*

Para lograr una mayor robustez en nuestro modelo se decidió realizar la implementación de dos estrategias, en la primera las subpoblaciones no estarían solapadas y otra en la cual tendrían k nodos comunes, dichas alternativas serán descritas seguidamente con un mayor nivel de detalle.

VMO local sin solapamiento

Esta variante tiene como principal característica que cada nodo pertenece únicamente a una subpoblación (véase Figura 2.5 *Esquema del modelo VMO-Top1*), lo cual significa que no se comparte información entre ellas.

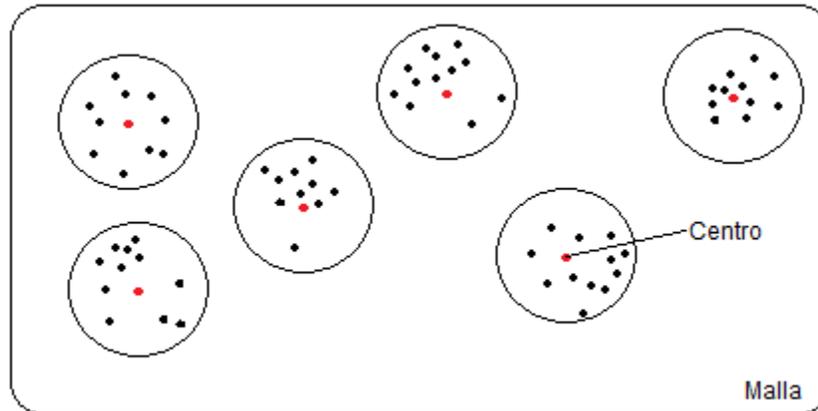


Figura 2.5 *Esquema del modelo VMO-Top1*

Esto trae consigo que cada subpoblación se comporte como una malla independiente, por consiguiente los operadores de generación de nodos en dirección a los óptimos locales, el óptimo global y hacia los nodos fronteras van a trabajar en cada subpoblación de manera autónoma. Todo este proceso se conoce como expansión. A ello se le sumaría el proceso de contracción de toda la malla donde se elegirían los mejores nodos para la próxima iteración. Al final cada subpoblación tendrá el mejor nodo, el óptimo global, por tanto se elegirá el mejor entre todas las subpoblaciones quedando este como resultado final (véase Figura 2.6 *Proceso general de modelo VMO-Top1*).

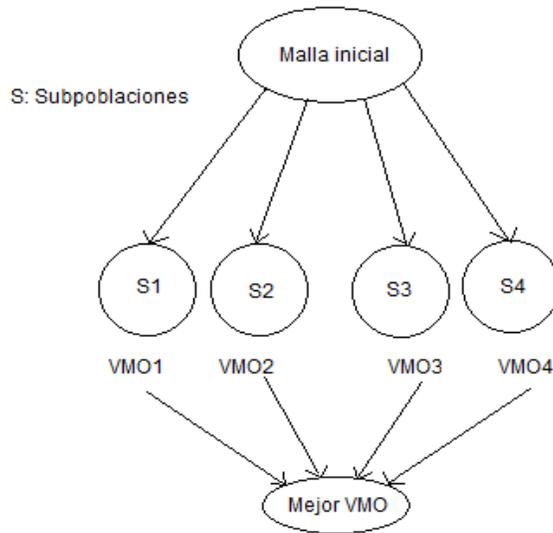


Figura 2.6 Proceso general de modelo VMO-Top1

VMO local con solapamiento

A diferencia de la variante descrita anteriormente, en esta los nodos pueden pertenecer a más de una subpoblación, es decir, solo una cantidad de nodos k pasada como parámetros por el usuario compartirá información (véase Figura 2.7 *Esquema del modelo VMO-Top2*).

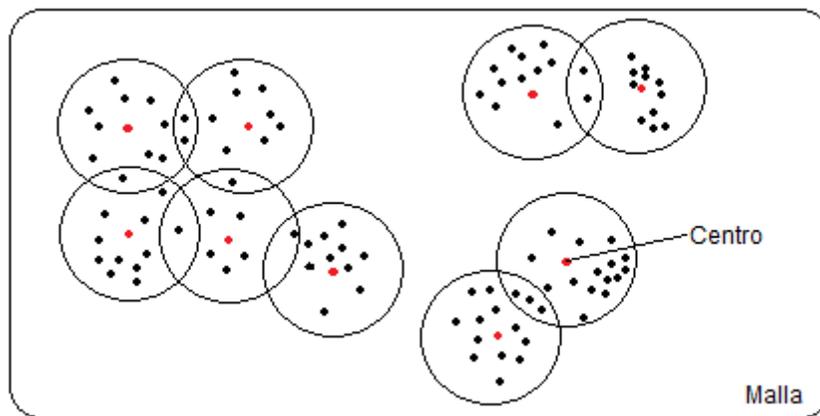


Figura 2.7 Esquema del modelo VMO-Top2

Por tanto, los nodos que pertenezcan a más de una subpoblación serán incluidos en el proceso de expansión (generación de nodos en dirección a los óptimos locales, global y hacia los nodos frontera) en cada una de las subpoblaciones. Para después aplicársele el proceso de contracción a toda la malla. Así al final obtener el mejor nodo de cada

subpoblación (véase Figura 2.8 *Proceso general de modelo VMO-Top2*), dándose la posibilidad que un óptimo global sea común en más de una subpoblación.

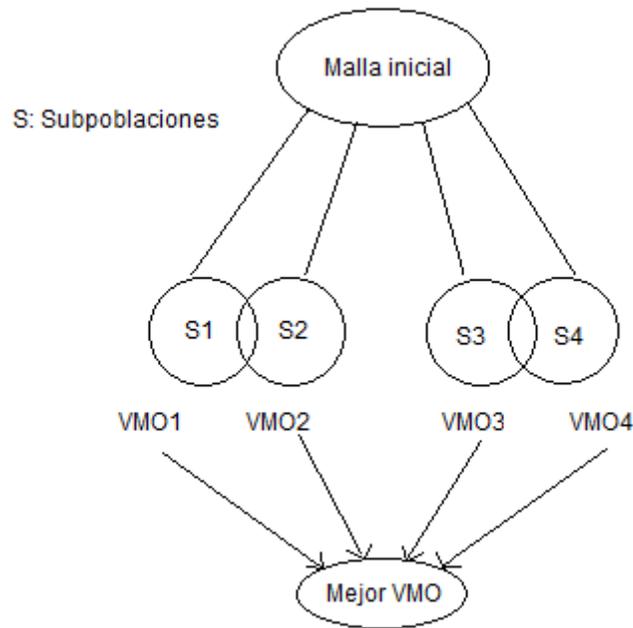
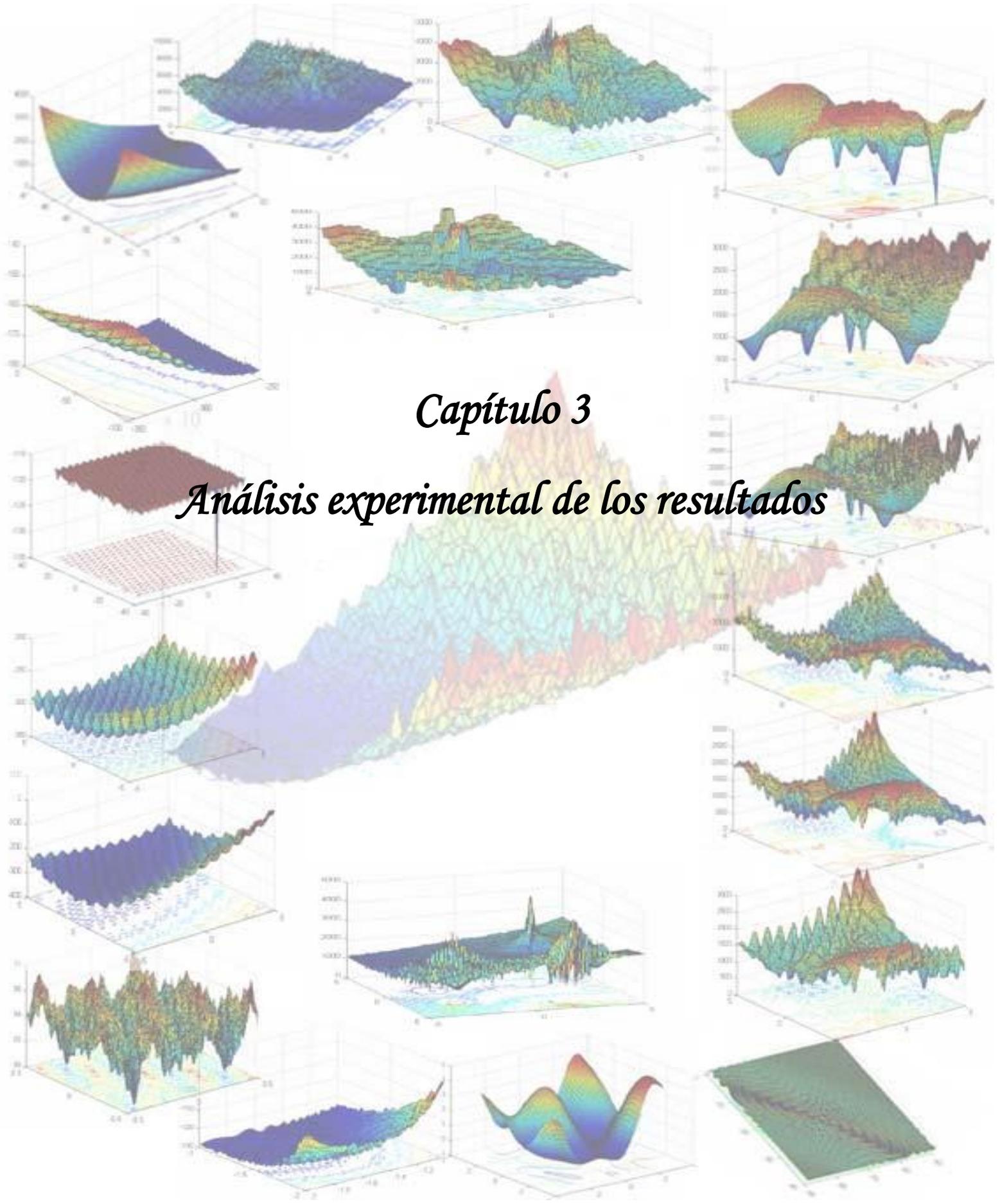


Figura 2.8 *Proceso general de modelo VMO-Top2*

Conclusiones parciales

En la meta-heurística poblacional VMO, debido a sus características, la malla tiende a converger prematuramente, de ahí que se plantee un nuevo modelo que logre escapar del estancamiento. El modelo estructurado se apoya en la división de la malla en subpoblaciones, con la intención de que cada uno explore una región diferente del espacio de búsqueda. Se implementaron dos variantes: subpoblaciones sin solapamiento (donde no se comparte información) y subpoblaciones con solapamiento, en la cual un determinado número de nodos puede pertenecer a más de una subpoblación compartiendo información entre sí.



Capítulo 3

Análisis experimental de los resultados

CAPÍTULO 3 ANÁLISIS EXPERIMENTAL DE LOS RESULTADOS

En presente capítulo se muestra un análisis de los resultados de las dos variantes del diseño VMO-Top1 y VMO-Top2, así como una descripción de los test estadísticos utilizados. Además aborda todo el marco experimental y el proceso de experimentación desarrollado en este trabajo.

Especificaciones del experimento

Antes de conformar cualquiera de las configuraciones que serán analizadas en el capítulo siguiente es necesario hacer un ajuste de varios parámetros del algoritmo estructurado, esenciales para su funcionamiento, estos son:

Parámetros del algoritmo:

- Tamaño de la malla inicial: 50 nodos.
- Número de nodos que definen la vecindad de un nodo en la malla: $k=5$.
- Número máximo de evaluaciones en la función objetivo: $C= 100\ 000$.
- Número máximo de soluciones generadas en cada iteración: $NS= 175$.
- Número de subpoblaciones: $subpob=10$.

De igual manera para hacer el experimento, se tomaron veinte funciones multimodales descritas en (Suganthan et al., 2005b). En cada caso se realizaron 25 simulaciones de las funciones involucradas en la prueba, y se definió el valor del error calculado para cada solución como $(F(x_i) - (F(x_i^*)))$ donde x_i^* es el óptimo global de la función.

Características generales de las funciones utilizadas:

- Ruidosas y rotadas.
- Desplazadas y no separables.

- Multimodales con muchos óptimos locales.
- Existencia de valles entre el óptimo local y el óptimo global.
- Continua pero diferenciable sólo en un conjunto de puntos.

Clasificación de las funciones:

- Siete funciones básicas todas multimodales ($F_6 - F_{12}$)

F_6 -Shifted Rosenbrock's Function

F_7 -Shifted Rotated Griewank's Function without Bounds

F_8 -Shifted Rotated Ackley's Function with Global Optimum on Bounds

F_9 -Shifted Rastrigin's Function

F_{10} -Shifted Rotated Rastrigin's Function

F_{11} -Shifted Rotated Weierstrass Function

F_{12} -Schwefel's Problem 2.13

- Dos funciones multimodales expandidas F_{13} y F_{14}

F_{13} -Expanded Extended Griewank's plus Rosenbrock's Function

F_{14} -Shifted Rotated Expanded Scaffer's

- Once funciones híbridas ($F_{15} - F_{25}$)

F_{15} -Hybrid Composition Function

F_{16} -Rotated Hybrid Composition Function

F_{17} -Rotated Hybrid Composition Function with Noise in Fitness

F_{18} -Rotated Hybrid Composition Function

F_{19} -Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum

F_{20} -Rotated Hybrid Composition Function with the Global Optimum on the Bounds

F_{21} -Rotated Hybrid Composition Function

F_{22} -Rotated Hybrid Composition Function with High Condition Number Matrix

F_{23} -Non-Continuous Rotated Hybrid Composition Function

F_{24} -Rotated Hybrid Composition Function

F_{25} -Rotated Hybrid Composition Function without Bounds

Propiedades de las funciones de prueba:

F_6 -Shifted Rosenbrock's Function

$$F_6(x) = \sum_{i=1}^{D-1} (100(z_i^2 - z_{i+1}^2)^2 + (z_i - 1)^2) + 390 \quad (3.1)$$

Donde $z = x - o + 1$, $x = [x_1, x_2, \dots, x_D]$, D : Dimensión, $o = [o_1, o_2, \dots, o_D]$: Óptimo global.

Propiedades:

- Multimodal
- Desplazada
- No separable
- Escalable
- El rango de valor de las variables es $x \in [-100,100]^D$ y el óptimo es

$$F_6(x^*) = 390.$$

F_7 -Shifted Rotated Griewank's Function without Bounds

$$F_7(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 - 180 \quad (3.2)$$

$$z = (x - o) * M, \quad x = [x_1, x_2, \dots, x_D],$$

D : dimensiones, $o = [o_1, o_2, \dots, o_D]$, M : matriz de transformación lineal con número de condición =3, $M = M' (1+0.3|N(0.1)|)$

Propiedades:

- Multimodal
- Rotada
- Desplazada
- No separable
- Escalable

- Ningún límite para variables x
- el óptimo es $F_7(\mathbf{x}^*) = -180$.

F_8 -Shifted Rotated Ackley's Function with Global Optimum on Bounds

$$F_8(\mathbf{x}) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)\right) + 20 + e - 140 \quad (3.3)$$

Donde $z = (x - o) * M$, $x = [x_1, x_2, \dots, x_D]$, D : Dimensión, o : Óptimo global, $o_{2j-1} = -32o_{2j}$ están aleatoriamente distribuidos en el rango, para $j = 1, 2, \dots, [D/2]$, M : Matriz de transformación lineal.

Propiedades:

- Multimodal
- Rotada
- Desplazada
- No separable
- Escalable
- Óptima global sobre la cota
- El rango de valor de las variables es $x \in [-32, 32]^D$ y el óptimo es

$$F_8(\mathbf{x}^*) = -140.$$

F_9 -Shifted Rastrigin's Function

$$F_9(\mathbf{x}) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) - 330 \quad (3.4)$$

Donde $z = x - o$, $x = [x_1, x_2, \dots, x_D]$, D : Dimensión, o : Óptimo global.

Propiedades:

- Multimodal
- Desplazada
- Separable
- Escalable
- Número enorme de óptimo local

- El rango de valor de las variables es $x \in [-5,5]^D$ y el óptimo es $F_9(x^*) = -330$.

F_{10} -Shifted Rotated Rastrigin's Function

$$F_{10}(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_i) + 10) - 330 \quad (3.5)$$

Donde $z = (x - o) * M$, $x = [x_1, x_2, \dots, x_D]$, D: Dimensión, o: Óptimo global, M: Matriz de transformación lineal.

Propiedades:

- Multimodal
- Desplazada
- Rotada
- No separable
- Escalable
- Número enorme de óptimo local
- El rango de valor de las variables es $x \in [-5,5]^D$ y el óptimo es $F_{10}(x^*) = -330$.

F_{11} -Shifted Rotated Weierstrass Function

$$F_{11}(x) = \sum_{i=1}^D \left(\sum_{k=0}^{kmax} [a^k \cos(2\pi b^k (z_i + 0.5))] \right) - D \sum_{k=0}^{kmax} [a^k \cos(2\pi b^k * 0.5)] + 90 \quad (3.6)$$

Donde $a = 0.5$, $b = 3$, $kmax = 20$, $z = (x - o) * M$, $x = [x_1, x_2, \dots, x_D]$, D: Dimensión, o: Óptimo global, M: Matriz de transformación lineal.

El rango de valor de las variables es $x \in [-0.5,0.5]^D$ y el óptimo es $F_{11}(x^*) = 90$.

Propiedades:

- Multimodal
- Desplazada
- Rotada
- No separable
- Escalable
- Continua pero diferenciable solo en un punto **F_{12} -Schwefel's Problem 2.13**

$$F_{12}(x) = \sum_{i=1}^D (A_i - B_i(x))^2 - 460, \quad (3.7)$$

Donde $A_i = \sum_{j=1}^D (a_{ij} \sin \alpha_j + b_{ij} \cos \alpha_j)$, $B_i(x) = \sum_{j=1}^D (a_{ij} \sin x_j + b_{ij} \cos x_j)$, para $i = 1, \dots, D$, D : Dimensión, A, B : son dos matriz $D * D$, a_{ij}, b_{ij} son números enteros aleatorios en el rango $[-100,100]$, $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_D]$, α_{ij} son números enteros aleatorios en el rango $[-\pi, \pi]$.

Propiedades:

- Multimodal
- Desplazada
- Rotada
- No separable
- Escalable
- Continua pero diferenciable solo en un punto
- El rango de valor de las variables es $x \in [-\pi, \pi]^D$ y el óptimo es $F_{12}(x^*) = -460$

F_{13} -Expanded Extended Griewank's plus Rosenbrock's Function($F_8 F_2$)

$$F_8(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 \quad (3.2)$$

$$F_2(x) = \sum_{i=1}^D (100(x_i^2 + x_{i+1})^2 + (x_i - 1)^2) \quad (3.8)$$

$$F_{13}(x) = F_8(F_2(z_1, z_2)) + F_8(F_2(z_2, z_3)) + \dots + F_8(F_2(z_{D-1}, z_D)) + F_8(F_2(z_D, z_1)) - 130 \quad (3.9)$$

$z = x - o + 1, x = [x_1, x_2, \dots, x_D]$: Óptimo global D : dimensión

Propiedades:

- Multimodal
- Desplazada
- No separable
- Escalable
- El rango de valor de las variables es $x \in [-3,1]^D$ y el óptimo es $F_{13}(x^*) = -130$

F_{14} -Shifted Rotated Expanded Scaffer's

$$F(x, y) = 0.5 + \frac{(\sin(\sqrt{x^2+y^2})-0.5)^2}{(1+0.001(x^2+y^2))^2} \quad (3.10)$$

$$F_{14}(x) = F(z_1, z_2) + F(z_2, z_3) + \dots + F(z_{D-1}, z_D) + F(z_D, z_1) - 300 \quad (3.11)$$

$z = (x - o) * M$, $x = [x_1, x_2, \dots, x_D]$ o : Óptimo global

Propiedades:

- Multimodal
- Desplazada
- No separable
- Escalable
- El rango de valor de las variables es $x \in [-100,100]^D$ y el óptimo es $F_{14}(x^*) = -300$

F_{15} -Hybrid Composition Function

$f_{1-2}(x)$: *Rastring's Funtion*

$$f_i(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (3.5)$$

$f_{3-4}(x)$: *Weierstrass Funtion*

$$f_i(x) = \sum_{i=1}^D \left(\sum_{k=0}^{kmax} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{kmax} [a^k \cos(2\pi b^k * 0.5)] \quad (3.6)$$

$$a = 0.5, b = 3, kmax = 20$$

$f_{5-6}(x)$: *Griewank's Funtion*

$$f_i(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 \quad (3.2)$$

$f_{7-8}(x)$: *Ackley's Funtion*

$$f_i(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)\right) + 20 + e \quad (3.3)$$

$f_{9-10}(x)$: *Sphere Funtion*

$$f_i(x) = \sum_{i=1}^D x_i^2 \quad (3.12)$$

$$\sigma_i = 1 \text{ for } i = 1, 2, \dots, D \quad \lambda = [1, 1, 10, 10, \frac{5}{60}, \frac{5}{60}, \frac{5}{32}, \frac{5}{32}, \frac{5}{100}, \frac{5}{100}]$$

M_i todas las matrices identicas

Propiedades:

- Multimodal
- Separable cerca del óptimo global (Rastrigin)
- Escalable
- Un número grande de óptimo local
- Las propiedades de las funciones están mixtas
- El rango de valor de las variables es $x \in [-5,5]^D$ y el óptimo es $F_{15}(x^*) = 120$

F_{16} -Rotated Hybrid Composition Function

Excepto M_i que son de transformación lineal con número de condición 2, todo las demás configuraciones son iguales a F_{15}

Propiedades:

- Multimodal
- Rotada
- No separable
- Escalable
- Un número grande de óptimo local
- Las propiedades de las funciones están mixtas
- Con ruido Gaussiano en su fitness
- El rango de valor de las variables es $x \in [-5,5]^D$ y el óptimo es $F_{16}(x^*) = 120$

F_{17} -Rotated Hybrid Composition Function with Noise in Fitness

F_{17} : F_{16} con ruido

donde $G(x)$ es $(F_{16} - 120)$

$$F_{17}(x) = G(x) * (1 + 0.2|N(0.1)|) + 120 \quad (3.13)$$

Propiedades:

- Multimodal
- Rotada
- No separable

- Escalable
- Un número grande de óptimo local
- Las propiedades de las funciones están mixtas
- El rango de valor de las variables es $x \in [-5,5]^D$ y el óptimo es $F_{17}(x^*) = 120$

F_{18} -Rotated Hybrid Composition Function

$f_{1-2}(x) = Ackley's Funtion$

$$f_i(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)\right) + 20 + e \quad (3.3)$$

$f_{3-4}(x): Rastring's Funtion$

$$f_i(x): \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (3.5)$$

$f_{5-6}(x): Sphere Funtion$

$$f_i(x) = \sum_{i=1}^D x_i^2 \quad (3.12)$$

$f_{7-8}(x): Weierstrass Funtion$

$$f_i(x): \sum_{i=1}^D \left(\sum_{k=0}^{kmax} [a^k \cos(2\pi b^k (x_i + 0.5))] \right) - D \sum_{k=0}^{kmax} [a^k \cos(2\pi b^k * 0.5)] \quad (3.6)$$

$$a = 0.5, b = 3, kmax = 20$$

$f_{9-10}(x): Griewank's Funtion$

$$f_i(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 \quad (3.2)$$

$$\sigma = [1,2,1.5,1.5,1,1,1.5,1.5,2,2] ;$$

$$\lambda = [2 * \frac{5}{32}, \frac{5}{32}, 2 * 1, 1, 2 * \frac{5}{100}, \frac{5}{100}, 2 * 10, 10, 2 * \frac{5}{60}, \frac{5}{60}]$$

M_i todas las matrices de rotación

$$o_{10} = [0,0, \dots, 0]$$

Propiedades:

- Multimodal
- Rotada

- No separable
- Escalable
- Un número grande de óptimo local
- Las propiedades de las funciones están mixtas
- El óptimo local esta fijo en el origen
- El rango de valor de las variables es $x \in [-5,5]^D$ y el óptimo es $F_{18}(x^*) = 10$

F_{19} -Rotated Hybrid Composition Function with a Narrow Basin for the Global Optimum

Es similar a F_{18} excepto:

$$\sigma = [0.1, 2, 1.5, 1.5, 1, 1, 1.5, 1.5, 2, 2];$$

$$\lambda = [0.1 * \frac{5}{32}, \frac{5}{32}, 2 * 1, 1, 2 * \frac{5}{100}, \frac{5}{100}, 2 * 10, 10, 2 * \frac{5}{60}, \frac{5}{60}]$$

Propiedades:

- Multimodal
- No separable
- Escalable
- Un número grande de óptimo local
- Las propiedades de las funciones están mixtas
- El óptimo local esta fijo en el origen
- El rango de valor de las variables es $x \in [-5,5]^D$ y el óptimo es $F_{19}(x^*) = 10$

F_{20} -Rotated Hybrid Composition Function with the Global Optimum on the Bounds

Es similar a F_{18} excepto que después de cargar el fichero de datos ,

$$o_{1(2j)} = 5 \quad j = 1, 2, \dots, \left\lfloor \frac{D}{2} \right\rfloor$$

Propiedades:

- Multimodal
- No separable

- Escalable
- Un número grande de óptimo local
- Las propiedades de las funciones están mixtas
- El óptimo global está en la frontera
- El rango de valor de las variables es $x \in [-5,5]^D$ y el óptimo es $F_{20}(x^*) = 10$

F_{21} -Rotated Hybrid Composition Function

$f_{1-2}(x)$: Rotated Expanded Scaffer's F_6 Funtion

$$F(x, y) = 0.5 + \frac{(\sin(\sqrt{x^2+y^2})-0.5^2)}{(1+0.001(x^2+y^2))^2} \quad (3.10)$$

$$f_i(x) := F(x_1, x_2) + F(x_2, x_3) + \dots + F(x_{D-1}, x_D) + F(x_D, x_1) \quad (3.11)$$

$f_{3-4}(x)$: Rastring's Funtion

$$f_i(x) : \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (3.5)$$

$f_{5-6}(x)$: $F_8 F_2$ Funtion

$$F_8(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 \quad (3.2)$$

$$F_2(x) = \sum_{i=1}^D (100 (x_i^2 + x_{i+1})^2 + (x_i - 1)^2) \quad (3.8)$$

$$f_i(x) = F_8(F_2(x_1, x_2)) + F_8(F_2(x_2, x_3)) + \dots + F_8(F_2(x_{D-1}, x_D)) + F_8(F_2(x_D, x_1)) \quad (3.9)$$

$f_{7-8}(x)$: Weierstrass Funtion

$$f_i(x) : \sum_{i=1}^D \left(\sum_{k=0}^{kmax} \left[a^k \cos\left(2\pi b^k (x_i + 0.5)\right) \right] \right) - D \sum_{k=0}^{kmax} [a^k \cos(2\pi b^k * 0.5)] \quad (3.6)$$

$$a = 0.5, b = 3, kmax = 20$$

$f_{9-10}(x)$: Griewank's Funtion

$$f_i(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 \quad (3.2)$$

$$\sigma = [1, 1, 1, 1, 1, 2, 2, 2, 2, 2],$$

$$\lambda = \left[5 * \frac{5}{100}, \frac{5}{100}, 5 * 1, 1, 5 * 1, 1, 5 * 10, 10, 5 * \frac{5}{200}, \frac{5}{200} \right]$$

M_i todas las matrices ortogonales

Propiedades:

- Multimodal
- Rotada
- No separable
- Escalable
- Un número grande de óptimo local
- Las propiedades de las funciones están mixtas
- El rango de valor de las variables es $x \in [-5,5]^D$ y el óptimo es $F_{21}(x^*) = 360$

F_{22} -Rotated Hybrid Composition Function with High Condition Number Matrix

Similar a F_{21} excepto las M_i 's cuyos números de condiciones son [10 20 50 100 200 1000 2000 3000 4000 5000]

Propiedades:

- Multimodal
- No separable
- Escalable
- Un número grande de óptimo local
- Las propiedades de las funciones están mixtas
- El óptimo global está en la frontera
- El rango de valor de las variables es $x \in [-5,5]^D$ y el óptimo es $F_{22}(x^*) = 360$

F_{23} -Non-Continuous Rotated Hybrid Composition Function

Similar a F_{21} excepto:

$$x_j = \begin{cases} x_j & |x_j - o_{1j}| < 1/2 \\ \text{round}(2x_j)/2 & |x_j - o_{1j}| \geq 1/2 \end{cases} \text{ para } j = 1, 2, \dots, D \quad (3.14)$$

$$\text{round}(x) = \begin{cases} a - 1 & \text{si } x \leq 0 \text{ AND } b \geq 0.5 \\ a & \text{si } b < 0.5 \\ a + 1 & \text{si } x > 0 \text{ AND } b \geq 0.5 \end{cases} \quad (3.15)$$

Donde a es la parte integra de x 's y b es la decimal de x 's

Propiedades:

- Multimodal
- No separable
- Escalable
- Un número grande de óptimo local
- Las propiedades de las funciones están mixtas
- No es continua
- El óptimo global está en la frontera
- El rango de valor de las variables es $x \in [-5,5]^D$ y el óptimo es $F_{23}(x^*) = 360$

F_{24} -Rotated Hybrid Composition Function

$f_1(x)$: *Weierstrass Funtion*

$$f_i(x): \sum_{i=1}^D \left(\sum_{k=0}^{kmax} \left[a^k \cos \left(2\pi b^k (x_i + 0.5) \right) \right] \right) - D \sum_{k=0}^{kmax} [a^k \cos(2\pi b^k * 0.5)] \quad (3.6)$$

$$a = 0.5, b = 3, kmax = 20$$

$f_2(x)$: *Rotated Expanded Scaffer's F6 Funtion*

$$F(x, y) = 0.5 + \frac{(\sin(\sqrt{x^2+y^2})-0.5^2)}{(1+0.001(x^2+y^2))^2} \quad (3.10)$$

$$f_i(x) := F(x_1, x_2) + F(x_2, x_3) + \dots + F(x_{D-1}, x_D) + F(x_D, x_1) \quad (3.11)$$

$f_3(x)$: **F_8F_2** *Funtion*

$$F_8(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 \quad (3.2)$$

$$F_2(x) = \sum_{i=1}^D (100(x_i^2 + x_{i+1})^2 + (x_i - 1)^2) \quad (3.8)$$

$$f_i(x) = F_8(F_2(x_1, x_2)) + F_8(F_2(x_2, x_3)) + \dots + F_8(F_2(x_{D-1}, x_D)) + F_8(F_2(x_D, x_1)) \quad (3.9)$$

$f_4(x)$ = *Ackley's Funtion*

$$f_i(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D z_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi z_i)\right) + 20 + e \quad (3.3)$$

$f_5(x)$: *Rastring's Funtion*

$$f_i(x): \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (3.5)$$

$f_6(x)$: Griewank's Funtion

$$f_i(x) = \sum_{i=1}^D \frac{z_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 \quad (3.2)$$

$f_7(x)$: Non – Continuous Expanded Scaffer's F6 Funtion

$$F(x, y) = 0.5 + \frac{(\sin(\sqrt{x^2+y^2})-0.5^2)}{(1+0.001(x^2+y^2))^2} \quad (3.10)$$

$$f_i(x) := F(y_1, y_2) + F(y_2, y_3) + \dots + F(x_{D-1}, x_D) + F(x_D, x_1) \quad (3.11)$$

$$y_j = \begin{cases} x_j |x_j| < 1/2 \\ \text{round}(2x_j)/2 & |x_j| \geq 1/2 \end{cases} \text{ para } j = 1, 2, \dots, D \quad (3.14)$$

$f_8(x)$: Non – Continuous Rastring's Funtion

$$f_i(x): \sum_{i=1}^D (y_i^2 - 10 \cos(2\pi y_i) + 10) \quad (3.5)$$

$$y_j = \begin{cases} x_j |x_j| < 1/2 \\ \text{round}(2x_j)/2 & |x_j| \geq 1/2 \end{cases} \text{ para } j = 1, 2, \dots, D \quad (3.14)$$

$f_9(x)$: High Conditioned Elliptic Funtion

$$f_i(x): \sum_{i=1}^D (10^6)^{\frac{i-1}{D-1}} x_i^2 \quad (3.15)$$

$f_{5-6}(x)$: Sphere Funtion with Noise Fitness

$$f_i(x) = (\sum_{i=1}^D x_i^2)(1 + 0,1|N(0,1)|) \quad (3.16)$$

$$\sigma_i = 2 \text{ donde } i = 1, 2, \dots, D$$

$$\lambda = [10, \frac{5}{20}, 1, \frac{5}{32}, 1, \frac{5}{100}, \frac{5}{50}, 1, \frac{5}{100}, \frac{5}{100}]$$

las M_i 's son matrices de rotadas cuyos numeros de condiciones son [100 50 30 10 5 5 4 3 2 2]

Propiedades:

- Multimodal
- Rotada
- No separable
- Escalable

- Un número grande de óptimo local
- Las propiedades de las funciones están mixtas
- Las funciones unimodales dan las áreas llanas
- El rango de valor de las variables es $x \in [-5,5]^D$ y el óptimo es $F_{24}(x^*) = 260$

F_{25} -Rotated Hybrid Composition Function without Bounds

Esta función es igual que F_{24} excepto el rango de búsqueda.

Propiedades:

- Multimodal
- No separable
- Escalable
- Un número grande de óptimo local
- Las propiedades de las funciones están mixtas
- Las funciones unimodales dan las áreas llanas
- El óptimo global se encuentra en la frontera
- No tiene fronteras
- El rango de valor de las variables es $x \in [2,5]^D$ y el óptimo es $F_{25}(x^*) = 260$

Técnicas Estadísticas

A la hora de comparar algoritmos con múltiples conjuntos de datos no existe un procedimiento ya establecido, esto se debe en gran medida a que estos mantienen un comportamiento no determinístico, por lo que la diferencia detectada entre los resultados de dos algoritmos podría deberse a factores aleatorios, y no a una mejora real.

Numerosas son las posibles técnicas a aplicar para determinar si un algoritmo difiere de forma significativa respecto a otro. Principalmente existen dos grupos:

- Los tests paramétricos (Luengo et al., 2009), son los más empleados. Se caracterizan en que utilizan por cada algoritmo y función el error medio alcanzado de un conjunto de ejecuciones y lo utilizan para identificar si la

diferencia entre dos algoritmos es estadísticamente significativa. Un ejemplo de estos test son el test de t-Student (para comparar dos algoritmos) y ANOVA (para comparar múltiples algoritmos).

Para aplicar dichos test paramétricos con un mínimo de fiabilidad, los resultados de los algoritmos deben de seguir una distribución normal, y mantener una varianza homogénea (Luengo et al., 2009). Cuando estas características son satisfechas los análisis paramétricos son correctos y seguros, y más sensibles (perceptivos) que los de la siguiente categoría, los test no paramétricos. Sin embargo, los algoritmos de Inteligencia Computacional raramente tienen distribución normal, sobre todo los estocásticos.

➤ Los test no-paramétricos (Sheskin, 2006), que utilizan una representación ordinal de los valores basada en el orden de los algoritmos para cada uno de los problemas. Un ejemplo de estos test son el test de Wilcoxon (Wilcoxon, 1945)(para comparar los resultados de dos algoritmos) y el test de Bonferroni-Dunn o el test de Holm (Holm, 1979)(para comparaciones de varios).

A lo largo la investigación se hará uso del segundo de estos grupos de pruebas, pues los resultados obtenidos por las meta-heurísticas no cumplen las condiciones requeridas para poder usar de forma correcta comparaciones paramétricas, además también se usará algunos estadísticos descriptivos.

Descripción de los tests estadísticos utilizados

En los experimentos computacionales, se ha aplicado varios métodos estadísticos. Ellos son: la mediana, el cálculo de desviación (gap), el test de Wilcoxon (Wilcoxon, 1945).

Seguidamente se muestra una descripción de las pruebas antes mencionadas teniendo en cuenta su funcionalidad básica y la utilidad de su empleo en nuestra investigación.

Cálculo de mediana: elemento central de un conjunto ordenado, esta permite comparar a simple vista el rendimiento de las alternativas implementadas. No se realiza el cálculo de la media puesto que se puede ver afectada por valores extremos.

Cálculo del porcentaje de desviación (gap): El rendimiento de la propuesta, relacionada con la distancia promedio entre los nodos, se evalúa sobre la base de una estadística clásica, porcentaje de desviación. La brecha en el porcentaje de una prueba a una función

multimodal consiste en la desviación relativa entre una solución, tomada como referencia, y la propuesta de solución analizada. El gap entre un par de soluciones puede ser calculado entonces mediante la expresión:

$$g(SA) = \frac{z_{ref} - z_{sa}}{z_{ref}} \times 100$$

Donde z_{ref} es considerado como la solución de referencia (los resultados de VMO reportados) y z_{sa} el enfoque de solución analizada. Un valor negativo de la brecha significa que el enfoque de solución es superior a los resultados considerados como referencia.

Test de Ranking de Signos de Wilcoxon (Wilcoxon, 1945): A la hora de hallar las diferencias entre los resultados de dos algoritmos y calcular un ranking utilizando dicho valor puede emplearse este test. Se trata de una alternativa no paramétrica al t -test de muestras apareadas. Nótese que en este caso, el ranking d va desde 1 hasta N , en vez de hasta k , como era el caso de los tres tests anteriores. Tras sumar los rankings diferenciándolos entre si son negativos o positivos, se obtienen dos valores R^+ y R^- . Si el menor de ellos es menor o igual al valor de la distribución T de Wilcoxon para N grados de libertad (T , se rechaza la hipótesis nula, y el algoritmo asociado al mayor de los valores es el mejor.

Marco experimental

Como se había planteado el uso de topologías dentro del VMO puede influir notablemente en los resultados obtenidos. A continuación se muestra un análisis de las variantes de nuestro modelo siendo analizadas con los test estadísticos mencionados en el apartado anterior. Estos análisis se realizaron para funciones de dimensión 10, 30 y 50:

1. Hallar la variante de menor mediana entre el VMO-Top1 y VMO-Puro.
2. Hallar la variante de menor mediana entre el VMO-Top2 y VMO-Puro.
3. Test de Wilcoxon contrastando VMO-Top1 y VMO-Top2.
4. De existir diferencias significativas, la variante de mejor comportamiento se comparará mediante el test de Wilcoxon con el modelo VMO puro.
5. Si no existiesen dichas diferencias, la de mejor mediana se comparará mediante el test de Wilcoxon con el modelo VMO puro.

Análisis de los resultados de los test estadísticos para las distintas variantes de diseño

Para determinar la mejor de las estrategias de implementación se realizó el procedimiento anterior antes descritos aplicado a los modelos analizados para cada una de las dimensiones (se empleó en todos los casos un enfoque de minimización)

Dimensión 10

La tabla 3.1 muestra los resultados para cada una de las funciones de prueba, los valores en **negrita** indican el mejor valor de la función objetivo para cada una de las instancias. Las tres primeras columnas representan, respectivamente, el código de la función, los resultados de VMO puro y la distancia promedio entre los nodos. El resto de las entradas de la tabla representan el promedio de las soluciones (*sol*, promedio de los resultados obtenidos en 25 corridas), el promedio de las distancias entre los nodos (*aveDist*, promedio de los resultados obtenidos en las 25 corridas) y el gap (*g*) para las propuestas de solución analizadas. Estas propuestas son:

- VMO-Top1: Modelo de VMO con topología sin solapamiento.
- VMO-Top2: Modelo de VMO con topología con solapamiento.

Tabla 3.1 Resultados para dimensión 10

<i>func</i>	VMO		VMO-Top1			VMO-Top2		
	<i>sol</i>	<i>aveDist</i>	<i>sol</i>	<i>-g</i>	<i>aveDist</i>	<i>sol</i>	<i>-g</i>	<i>aveDist</i>
F6	2,964E+01	4,895E+00	2,933E+01	7,711E+02	4,264E+01	2,118E+01	8,289E+02	4,547E+01
F7	1,267E+03	2,821E+01	1,268E+03	9,628E+01	5,538E+01	1,268E+03	9,204E+01	5,418E+01
F8	2,033E+01	6,621E+01	2,037E+01	4,069E+01	9,314E+01	2,036E+01	4,750E+01	9,765E+01
F9	1,217E+01	5,030E+00	1,226E+01	9,205E+01	9,661E+00	1,279E+01	9,055E+01	9,586E+00

F10	6,375E+00	1,960E-01	4,866E+00	2,250E+02	6,370E-01	4,464E+00	1,870E+02	5,624E-01
F11	2,104E+00	2,964E-02	7,855E-01	1,420E+02	7,172E-02	7,461E-01	3,207E+02	1,247E-01
F12	3,528E+02	1,765E-01	1,146E+01	2,653E+02	6,449E-01	7,134E+01	3,262E+02	7,524E-01
F13	9,202E-01	3,824E-01	9,981E-01	2,830E+02	1,465E+00	1,039E+00	3,048E+02	1,548E+00
F14	2,667E+00	1,083E+01	2,555E+00	2,739E+02	4,049E+01	2,442E+00	2,369E+02	3,649E+01
F15	2,437E+02	2,576E-01	2,084E+02	1,804E+02	7,224E-01	2,497E+02	2,143E+02	8,097E-01
F16	1,087E+02	2,099E-01	9,655E+01	2,174E+02	6,661E-01	9,759E+01	2,082E+02	6,468E-01
F17	1,166E+02	2,380E-01	1,018E+02	2,357E+02	7,989E-01	1,052E+02	2,345E+02	7,960E-01
F18	6,611E+02	2,040E-01	5,001E+02	1,103E+02	4,292E-01	4,801E+02	1,096E+02	4,276E-01
F19	6,201E+02	2,024E-01	5,623E+02	1,093E+02	4,237E-01	6,039E+02	1,071E+02	4,192E-01
F20	6,771E+02	2,025E-01	6,039E+02	1,227E+02	4,509E-01	6,658E+02	1,070E+02	4,190E-01
F21	4,600E+02	1,915E-01	4,840E+02	1,127E+02	4,074E-01	5,000E+02	1,122E+02	4,065E-01
F22	7,600E+02	4,800E-01	7,607E+02	1,456E+02	1,179E+00	7,603E+02	1,532E+02	1,215E+00
F23	5,775E+02	5,664E-01	5,691E+02	2,510E+02	1,988E+00	5,659E+02	1,663E+02	1,508E+00
F24	2,240E+02	1,914E-01	2,000E+02	1,155E+02	4,125E-01	2,000E+02	1,129E+02	4,074E-01
F25	2,000E+02	1,894E-01	2,120E+02	1,173E+02	4,115E-01	2,000E+02	1,147E+02	4,067E-01

En la siguiente tabla se muestran los resultados de la mediana en las alternativas implementadas:

Tabla 3.2 Mediana

Configuraciones	Mediana
VMO-Puro	2,000E+02
VMO-Top1	1,018E+02
VMO-Top2	1,052E+02

F6	2,690E+03	5,471E+00	7,057E+02	1,206E+02	1,207E+01	1,117E+03	8,289E+02	1,193E+01
F7	4,712E+03	1,490E+01	4,719E+03	9,593E+01	2,918E+01	4,721E+03	9,873E+01	2,960E+01
F8	2,099E+01	1,074E+02	2,100E+01	5,347E+01	1,648E+02	2,099E+01	5,521E+01	1,667E+02
F9	1,374E+02	5,785E+00	1,220E+02	1,024E+02	1,171E+01	1,189E+02	9,955E+01	1,154E+01
F10	9,213E+01	2,315E-01	3,776E+01	1,095E+02	4,851E-01	4,098E+01	1,089E+02	4,836E-01
F11	2,351E+01	2,491E-02	1,510E+01	8,836E+01	4,692E-02	1,400E+01	9,186E+01	4,779E-02
F12	2,985E+04	1,489E-01	2,069E+04	9,135E+01	2,850E-01	1,966E+04	9,729E+01	2,938E-01
F13	8,453E+00	3,426E-01	6,196E+00	2,034E+02	1,039E+00	5,629E+00	1,472E+02	8,469E-01
F14	1,221E+01	1,452E+01	1,226E+01	2,952E+02	5,736E+01	1,232E+01	3,848E+02	7,038E+01
F15	4,424E+02	2,284E-01	3,609E+02	1,036E+02	4,650E-01	3,770E+02	1,010E+02	4,591E-01
F16	1,672E+02	2,334E-01	2,040E+02	1,060E+02	4,808E-01	2,289E+02	1,051E+02	4,786E-01
F17	2,609E+02	2,659E-01	2,289E+02	2,512E+02	9,339E-01	2,597E+02	2,934E+02	1,046E+00
F18	9,182E+02	3,302E-01	9,087E+02	6,736E+01	5,526E-01	9,097E+02	6,981E+01	5,607E-01
F19	9,186E+02	3,238E-01	9,088E+02	7,154E+01	5,555E-01	9,083E+02	7,036E+01	5,517E-01
F20	9,159E+02	3,147E-01	9,086E+02	7,257E+01	5,430E-01	9,087E+02	7,390E+01	5,472E-01
F21	5,756E+02	2,313E-01	5,120E+02	9,887E+01	4,599E-01	5,000E+02	9,929E+01	4,609E-01
F22	9,849E+02	7,653E-01	9,455E+02	1,440E+02	1,867E+00	9,575E+02	1,935E+02	2,246E+00
F23	6,760E+02	3,902E-01	6,266E+02	1,760E+02	1,077E+00	6,317E+02	1,482E+02	9,683E-01
F24	2,320E+02	2,335E-01	2,001E+02	9,873E+01	4,641E-01	2,000E+02	1,006E+02	4,685E-01
F25	2,314E+02	2,290E-01	2,000E+02	1,011E+02	4,605E-01	2,000E+02	1,027E+02	4,643E-01

Como se observa en la Tabla 3.4 *Resultados para dimensión 30* para dimensión 30 las variantes presentadas VMO-Top1 y VMO-Top1 superan, en cuanto a calidad de resultados, al modelo VMO puro. Este último solamente es superior en 4 de las 20 instancias, también observamos el comportamiento de las distancias entre los nodos (aveDist) de las diferentes variantes topológicas, donde siguen siendo muy superiores a las obtenidas por el VMO puro.

Así como también en la **Tabla 3.5 Mediana** aprecia que la mejor de las variantes es el VMO-Top1.

Tabla 3.5 Mediana

Configuraciones	Mediana
VMO-Puro	2,609E+02
VMO-Top1	2,289E+02
VMO-Top2	2,597E+02

Al aplicarse la prueba de Wilcoxon (Tabla 3.6 **Resultados del test de Wilcoxon para dimensión 30**), sucede algo similar a lo ocurrido para dimensión 10: los modelos híbridos basados en ESC no difieren entre sí, sin embargo respecto al VMO puro, sí lo hacen de manera positiva.

Tabla 3.6 Resultados del test de Wilcoxon para dimensión 30

	Comparación	p-value	Hipótesis
dimensión $n = 10$	VMO-Top1 vs VMO-Top2	0.235	Aceptada
	VMO vs VMO-Top1	0.002	Rechazada
	VMO vs VMO-Top2	0.003	Rechazada

Dimensión 50

Para dimensión 50 (Tabla 3.7 **Resultados para dimensión 50**) puede apreciarse que VMO-puro solamente supera a la variante VMO-Top1 en una de las instancias, mientras que es superior a la segunda variante en solo dos. Al igual que para las dimensiones anteriores la distancia entre nodos (aveDist) de ambas variantes topológicas sigue un comportamiento muy superior que el modelo puro garantizando una mayor exploración del espacio de búsqueda. Aquí también como en las dimensiones anteriores la mediana del VMO-Top1 sigue comportándose mejor que las demás (Tabla 3.8 **Mediana**).

Tabla 3.7 Resultados para dimensión 50

<i>func</i>	VMO		VMO-Top1			VMO-Top2		
	<i>sol</i>	aveDist	<i>sol</i>	<i>-g</i>	aveDist	<i>sol</i>	<i>-g</i>	aveDist
F6	3,216E+03	5,156E+00	1,956E+03	8,031E+01	9,297E+00	1,570E+03	8,288E+01	9,429E+00
F7	6,246E+03	1,357E+01	6,266E+03	1,170E+02	2,945E+01	6,267E+03	1,106E+02	2,858E+01
F8	2,120E+01	1,361E+02	2,118E+01	5,839E+01	2,156E+02	2,118E+01	5,673E+01	2,134E+02
F9	4,354E+02	5,976E+00	2,903E+02	1,057E+02	1,230E+01	2,971E+02	1,089E+02	1,248E+01
F10	2,665E+02	2,287E-01	1,188E+02	1,053E+02	4,695E-01	1,170E+02	9,850E+01	4,539E-01
F11	5,139E+01	2,447E-02	3,648E+01	9,394E+01	4,746E-02	3,608E+01	9,784E+01	4,841E-02
F12	2,444E+05	1,573E-01	1,442E+05	8,999E+01	2,988E-01	1,606E+05	9,306E+01	3,036E-01
F13	2,727E+01	3,373E-01	1,761E+01	2,081E+02	1,039E+00	1,853E+01	2,365E+02	1,135E+00
F14	2,193E+01	1,788E+01	2,186E+01	3,328E+02	7,740E+01	2,198E+01	3,420E+02	7,905E+01
F15	5,348E+02	2,294E-01	3,442E+02	1,046E+02	4,694E-01	3,396E+02	9,936E+01	4,574E-01
F16	2,538E+02	2,277E-01	1,984E+02	1,048E+02	4,664E-01	2,162E+02	1,087E+02	4,752E-01
F17	3,276E+02	3,039E-01	2,715E+02	7,515E+02	2,587E+00	2,465E+02	3,389E+02	1,334E+00
F18	9,897E+02	2,634E-01	9,520E+02	7,718E+01	4,666E-01	9,522E+02	8,209E+01	4,796E-01
F19	1,004E+03	2,516E-01	9,496E+02	8,979E+01	4,775E-01	9,539E+02	9,008E+01	4,783E-01
F20	9,948E+02	2,505E-01	9,432E+02	9,294E+01	4,833E-01	9,527E+02	8,504E+01	4,635E-01
F21	6,793E+02	2,338E-01	5,395E+02	9,928E+01	4,660E-01	5,396E+02	9,302E+01	4,514E-01
F22	1,028E+03	8,035E-01	1,004E+03	2,058E+02	2,457E+00	1,002E+03	2,063E+02	2,461E+00
F23	1,040E+03	5,552E-01	1,002E+03	1,971E+02	1,650E+00	1,007E+03	1,348E+02	1,304E+00
F24	7,269E+02	3,087E-01	5,502E+02	1,113E+02	6,521E-01	7,641E+02	3,767E+02	1,472E+00
F25	7,403E+02	2,605E-01	6,521E-01	2,471E+02	9,042E-01	5,984E+02	2,644E+02	9,493E-01

Tabla 3.8 Mediana

Configuraciones	Mediana
VMO-Puro	6,793E+02
VMO-Top1	3,442E+02
VMO-Top2	5,396E+02

Nuevamente sucede, al aplicarse el test de Wilcoxon, algo similar. Las alternativas híbridas propuestas en nuestra investigación superan estadísticamente a las variantes VMO puro y VMO con BL aplicada a priori y no difieren entre sí (Tabla 3.9 *Resultados del test de Wilcoxon para dimensión 50*).

Tabla 3.9 Resultados del test de Wilcoxon para dimensión 50

	Comparación	p-value	Hipótesis
dimensión $n = 10$	VMO-Top1 vs VMO-Top2	0.171	Aceptada
	VMO vs VMO-Top1	0.000	Rechazada
	VMO vs VMO-Top2	0.001	Rechazada

Conclusiones parciales

Después de haber realizado los test estadísticos descritos en las secciones anteriores y analizar los comportamientos en el modelo estructurado para las dimensiones 10, 30 y 50, es posible concluir que para la mediana su comportamiento es el mismo, la primera variante topológica VMO-Top1 posee un mejor rendimiento. Mientras que utilizando el porcentaje de desviación estándar (*gap*) el valor negativo en cada una de las dimensiones indica que el enfoque de la solución VMO-Top1 y VMO-Top2 es superior al VMO-Puro. El promedio de las distancias entre nodos (*aveDist*) revela que en ambas variantes estructuradas hay una mayor exploración en el espacio de búsqueda; y finalmente el test de Wilcoxon muestra que aunque no existen diferencias significativas entre las variantes estructuradas, si existen diferencias significativas con respecto al modelo puro y esto conlleva a que el modelo estructurado sea superior al modelo puro. Además se observó que las variantes topológicas muestran un mejor desempeño para las funciones con dimensión 50.

CONCLUSIONES

Como resultado de la presente investigación, se diseñó e implementó una nueva variante del algoritmo VMO continuo con topologías estructuradas, incorporando estas estrategias en el proceso de generación de nodos. Donde el diseño de esta nueva variante se inspira en otras estrategias locales poblacionales (ej. Topología de Anillo en Optimización de Enjambres de Partículas); también se implementaron las dos alternativas distintas del modelo teniendo en cuenta la información a compartir.

A partir de lo antes expuesto se pudo comprobar mediante la aplicación de varios análisis estadísticos: Mediana, Porcentaje de desviación y el test de Ranking de Signos de Wilcoxon, que las variantes del modelo VMO-Estructurado estudiadas e implementadas presentan, de manera significativa, una mejoría en cuanto a calidad de resultados a los arrojados por el modelo VMO puro.

En lo que se refiere a cuál alternativa del modelo VMO-Estructurado presentó los mejores resultados, también se puede concluir que de manera general no presentan diferencias significativas entre sí.

RECOMENDACIONES

La optimización es un campo que no se agota con la investigación realizada, por tanto para proseguir con la misma existen varios aspectos a recomendar, entre los cuales destacan:

1. Estudiar el comportamiento de la meta-heurística con diferentes subpoblaciones.
2. Estudiar el rendimiento de la meta-heurística para otras variantes topológicas (ej. Von Newman).
3. Comparar las nuevas propuestas con otros algoritmos estructurados (ej. PSO con topología de anillo).

REFERENCIAS BIBLIOGRÁFICAS

- BÄCK, T., FOGEL, D. Y MICHALEWICZ, Z. 1997. *Evolutionary Computation 1. IOP Press.*
- BATISTA, B. M. & MORENO-VEGA, J. M. 2013. *Metaheurísticas en Logística Portuaria y Problemas Afines.*
- BELLO, R. & PURIS, A. 2009. *Optimización basada en Mallas Dinámicas. Su aplicación en la solución de problemas de optimización continuos.*
- BHARAT, T. V., SIVAPULLAIAH, P. V. & ALLAM, M. M. 2009. Swarm intelligence-based solver for parameter estimation of laboratory through-diffusion transport of contaminants. *Computers and Geotechnics*, 36, 984-992.
- BRATTON, D. & KENNEDY, J. Defining a standard for particle swarm optimization. *Swarm Intelligence Symposium, 2007. SIS 2007. IEEE, 2007. IEEE*, 120-127.
- CLARK, A., ALMADA-LOBO, B. & ALMEDER, C. 2011. Lot sizing and scheduling: industrial extensions and research opportunities. *International Journal of Production Research*, 49(9).
- CHRISTODOULOU, S. Ant colony optimization in construction scheduling. *Proceedings of the 2005 ASCE International Conference on Computing in Civil Engineering*, July, 2005. 11-17.
- DARWIN, C. 1859. *On the Origin of Species.* .
- DE CASTRO, L. N. 2007. Fundamentals of natural computing: an overview. *Physics of Life Reviews* 4, 1, 1–36.
- DELGADO, E. M. 2010. *Supernova: un algoritmo novedoso de optimización global.* Universidad nacional de Colombia.
- DORIGO, M. & CARO, G. D. 1999. *The Ant Colony Optimization meta-heuristic.*
- DORIGO, M. Y. S., T. 2003. *Handbook of Metaheuristics. In: SPRINGER (ed.).*
- ENGELBRECHT, A. P. 2007. *Computational intelligence: an introduction*, John Wiley & Sons.
- ESMORÍS, A. V. 2013. *Algoritmos heurísticos en optimización* Maestria, Universidad de Santiago de Compostela.
- FEOKTISTOV, V. 2006. *Differential Evolution. Springer.*
- GLOVER, F. 1986. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13, 533-549.
- GLOVER, F. & KOCHENBERGER, G. 2003. *Handbook of Metaheuristics.*

- GLOVER, F. & LAGUNA, M. 1997. *Tabu Search*, MA, Kluwer Academic Publishers.
- GOLDBERG, D. E. (ed.) 1998. *Genetic Algorithms in Search*, University of Alabama: Addison-Wesley Publishing Company.
- GUILLÉN, G., BADELL, M. & PUIGJANER, L. 2007. A holistic framework for short-term supply chain management integrating production and corporate financial planning. *International Journal of Production Economics*, 106(1).
- HATAMLOU, A. 2013. Black hole: A new heuristic optimization approach for data clustering. *Information Sciences*, 222(1).
- HOLM, S. 1979. A simple sequentially rejective multiple test procedure. *Scand Journal Stat*, 6, 65-70.
- HSIEH, S.-T., SUN, T.-Y., LIU, C.-C. & TSAI, S.-J. Solving large scale global optimization using improved particle swarm optimizer. *Evolutionary Computation*, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on, 2008. IEEE, 1777-1784.
- JANG, J., SUN, C. Y MIZUTAN, E. 1996. *Neuro-fuzzy and soft computing*. Prentice Hall.
- KENNEDY, J. & EBERHART, R. C. Particle swarm optimization. *IEEE International Conference on Neural Networks*, 1995 Piscataway, New York, USA. , 1942-1948.
- LAGUNA, M. & MARTÍ, R. 2003. Scatter Search. Methodology and Implementation in C.
- LAMIRI, M., GRIMAUD, F. & XIE, X. 2009. Optimization methods for a stochastic surgery planning problem. *International Journal of Production Economics*, 120(2).
- LIAO, T. W. 2010. Two hybrid differential evolution algorithms for engineering design optimization. *Applied Soft Computing*, 10(4).
- LOURENÇO, H. R., PAIXÃO, J. P. & PORTUGAL, R. 2001. Multiobjective metaheuristics for the bus driver scheduling problem. *Transportation Science*, 35, 331-343.
- LOZANO, J., LARRAÑAGA, P., INZA, I. & BENGOTXEA, E. (eds.) 2006. *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms.*: Springer Verlag.
- LUENBERGUER, D. 1989. *Programación lineal y no lineal*, Luenberguer., Addison-Wesley Iberoamericana.
- LUENGO, J., GARCÍA, S. & HERRERA, F. 2009. *A Study on the Use of Non-Parametric Tests for Analyzing the Evolutionary Algorithms' Behaviour*.
- MARTÍNEZ, Y., PURIS, A. & BELLO, R. 2006. Aplicación de métodos heurísticos en la solución de problemas de configuración. *Cd con las memorias del*

V Conferencia Internacional de Ciencias Empresariales CICE 2006. Santa Clara, Cuba.

MICHALEWICZ, Z. Y. S., M. 2002. Evolutionary Optimization. In: PRESS, K. A. (ed.).

MOLINA CABRERA, D. 2007. *Algoritmos Meméticos con Aplicación Adaptativa de la Búsqueda Local para Optimización Continua*. Universidad de Granada.

MUÑOZ, A. D. 2007. *Metaheurísticas*, Librería-Editorial Dykinson.

NICKEL, S., SCHRÖDER, M. & STEEG, J. 2012. Mid-term and short-term planning support for home health care services. *European Journal of Operational Research*, 219(3).

NIETO, J. M. G. 2006. Algoritmos Basados en Cúmulos de Partículas Para la Resolución de Problemas Complejos.

PAPADAKOS, N., TZALLAS-REGAS, G., RUSTEM, B. & THOMS, J. 2011. Risky traveling salesman problem. *European Journal of Operational Research*, 212(1).

POLI, R., KENNEDY, J. & BLACKWELL, T. 2007. Particle swarm optimization. *Swarm intelligence*, 1, 33-57.

POLYA, G. 2014. *How to solve it: A new aspect of mathematical method*, Princeton university press.

PURIS, A., BELLO, R., MOLINA, D. & HERRERA, F. 2012. Variable mesh optimization for continuous optimization problems. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 16, 511-525.

PURIS, A. Y. 2009. *Desarrollo de meta-heurísticas poblacionales para la solución de problemas complejos*

Universidad Central "Marta Abreu" de las Villas.

RADA-VILELA, J., ZHANG, M. & SEAH, W. A performance study on synchronous and asynchronous updates in particle swarm optimization. Proceedings of the 13th annual conference on Genetic and evolutionary computation, 2011. ACM, 21-28.

RAO, S. 1996. *Engineering Optimization, Theory and practice*. Wiley.

RODRÍGUEZ ORTIZ, C. 2010. Algoritmos heurísticos y metaheurísticos para el problema de localización de regeneradores.

SCHILDE, M., DOERNER, K. F. & HARTL, R. F. 2011. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. *Computers & Operations Research*, 38(12).

SHARIFF, S. S. R., MOIN, N. H. & OMAR, M. 2012. Location allocation modeling for healthcare facility planning in Malaysia. *Computers & Industrial Engineering*, 62(4).

- SHESKIN, D. 2006. *Handbook of parametric and nonparametric statistical procedures*.
- SHI, X. H., LIANG, Y. C., LEE, H. P., LU, C. & WANG, Q. 2007. Particle swarm optimization-based algorithms for TSP and generalized TSP. *Information Processing Letters*, 103, 169-176.
- SOLIS, F. J. & WETS, R. J. 1981. Minimization by Random Search Techniques. *Mathematical Operation Research*, 6, 19-30.
- SPENDLEY, W., HEXT, G. R. & HIMSWORTH, F. R. 1962. Sequential Application of Simplex Designs in Optimizacion and Evolutionary Operation. *Technometrics*, 4, 441-457.
- STORN, R. & PRICE, K. 1997. Differential Evolution. A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11, 341-359.
- SUGANTHAN, P., HANSEN, N., LIANG, J. J., DEB, K., CHEN, Y.-P., AUGER, A. & TIWARI, S. 2005a. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization. Nanyang Technological University.
- SUGANTHAN, P. N., HANSEN, N., LIANG, J. J., DEB, K., CHEN, Y.-P., AUGER, A. & TIWARI, S. 2005b. Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization.
- VALERO, F. L. 2008. *Metaheurísticas avanzadas para problemas reales en redes de telecomunicaciones*. Universidad de Málaga.
- WILCOXON, F. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6), 80-83.
- WOLPERT, D. H. & MACREADY, W. G. 1997. No Free Lunch Theorems for Optimization. *IEEE Transactions of Evolutionary Computation*, 1, 67-82.
- ZHAO, S.-Z., LIANG, J. J., SUGANTHAN, P. N. & TASGETIREN, M. F. Dynamic multi-swarm particle swarm optimizer with local search for large scale global optimization. *Evolutionary Computation*, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on, 2008. IEEE, 3845-3852.

**ANEXO A TABLAS DE LOS VALORES OBTENIDOS PARA
CADA UNA DE LAS FUNCIONES DE PRUEBA DEL
CEC'2005 POR LOS MODELOS ESTUDIADOS.**

Tabla A.1 Resultados del modelo VMO puro

Funciones	VMO-puro Dimensión 10	VMO-puro Dimensión 30	VMO-puro Dimensión 50
F6	2,964E+01	2,690E+03	3,216E+03
F7	1,267E+03	4,712E+03	6,246E+03
F8	2,033E+01	2,099E+01	2,120E+01
F9	1,217E+01	1,374E+02	4,354E+02
F10	6,375E+00	9,213E+01	2,665E+02
F11	2,104E+00	2,351E+01	5,139E+01
F12	3,528E+02	2,985E+04	2,444E+05
F13	9,202E-01	8,453E+00	2,727E+01
F14	2,667E+00	1,221E+01	2,193E+01
F15	2,437E+02	4,424E+02	5,348E+02
F16	1,087E+02	1,672E+02	2,538E+02
F17	1,166E+02	2,609E+02	3,276E+02
F18	6,611E+02	9,182E+02	9,897E+02
F19	6,201E+02	9,186E+02	1,004E+03
F20	6,771E+02	9,159E+02	9,948E+02
F21	4,600E+02	5,756E+02	6,793E+02
F22	7,600E+02	9,849E+02	1,028E+03
F23	5,775E+02	6,760E+02	1,040E+03
F24	2,240E+02	2,320E+02	7,269E+02
F25	2,000E+02	2,314E+02	7,403E+02

Tabla A.2 *Resultados del modelo VMO-Top1*

Funciones	VMO-Top1 Dimensión 10	VMO-Top1 Dimensión 30	VMO-Top1 Dimensión 50
F6	2,933E+01	7,057E+02	1,956E+03
F7	1,268E+03	4,719E+03	6,266E+03
F8	2,037E+01	2,100E+01	2,118E+01
F9	1,226E+01	1,220E+02	2,903E+02
F10	4,866E+00	3,776E+01	1,188E+02
F11	7,855E-01	1,510E+01	3,648E+01
F12	1,146E+01	2,069E+04	1,442E+05
F13	9,981E-01	6,196E+00	1,761E+01
F14	2,555E+00	1,226E+01	2,186E+01
F15	2,084E+02	3,609E+02	3,442E+02
F16	9,655E+01	2,040E+02	1,984E+02
F17	1,018E+02	2,289E+02	2,715E+02
F18	5,001E+02	9,087E+02	9,520E+02
F19	5,623E+02	9,088E+02	9,496E+02
F20	6,039E+02	9,086E+02	9,432E+02
F21	4,840E+02	5,120E+02	5,395E+02
F22	7,607E+02	9,455E+02	1,004E+03
F23	5,691E+02	6,266E+02	1,002E+03
F24	2,000E+02	2,001E+02	5,502E+02
F25	2,120E+02	2,000E+02	6,521E-01

Tabla A.3 *Resultados del modelo VMO-Top2*

Funciones	VMO-Top2	VMO-Top2	VMO-Top2
	Dimensión 10	Dimensión 30	Dimensión 50
F6	2,118E+01	1,117E+03	1,570E+03
F7	1,268E+03	4,721E+03	6,267E+03
F8	2,036E+01	2,099E+01	2,118E+01
F9	1,279E+01	1,189E+02	2,971E+02
F10	4,464E+00	4,098E+01	1,170E+02
F11	7,461E-01	1,400E+01	3,608E+01
F12	7,134E+01	1,966E+04	1,606E+05
F13	1,039E+00	5,629E+00	1,853E+01
F14	2,442E+00	1,232E+01	2,198E+01
F15	2,497E+02	3,770E+02	3,396E+02
F16	9,759E+01	2,289E+02	2,162E+02
F17	1,052E+02	2,597E+02	2,465E+02
F18	4,801E+02	9,097E+02	9,522E+02
F19	6,039E+02	9,083E+02	9,539E+02
F20	6,658E+02	9,087E+02	9,527E+02
F21	5,000E+02	5,000E+02	5,396E+02
F22	7,603E+02	9,575E+02	1,002E+03
F23	5,659E+02	6,317E+02	1,007E+03
F24	2,000E+02	2,000E+02	7,641E+02
F25	2,000E+02	2,000E+02	5,984E+02