

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



Desarrollo de un Algoritmo Memético basado en
VMO.

Autor: Carlos Alberto Martínez Pérez

Tutor: Dr. Amilkar Yudier Puris Cáceres

Santa Clara, 2010

A esa gigantesca familia que tengo.

A aquellos que en algún momento de sus vidas me han tenido en cuenta:

*A esa familia que tengo por un montón de lugares y que me ha apoyado siempre
de manera incondicional.*

*A mis abuelos Kiko y Manolo, que daban cualquier cosa por
hacerme feliz.*

*A mis amigos, que también son familia, pero los quiero mencionar aparte
para que no se disgusten.*

*A mi tutor el Dr. Amilkar, también mi amigo, y quien me brindó su ayuda
en todo momento.*

A todos los profesores que de alguna forma u otra me ayudaron en mi formación.

A todos, muchas gracias.

SÍNTESIS

Los métodos aproximativos para la resolución de problemas de optimización se han venido desarrollando debido a la alta complejidad que dichos problemas han alcanzado. Una buena explotación de las soluciones ya encontradas, así como una correcta exploración del espacio de búsqueda son factores imprescindibles para obtener soluciones de calidad. Esta investigación está encaminada a desarrollar un modelo que combine de manera equilibrada estos factores, tomando como base poblacional para la exploración el modelo basado en Mallas Variables e incorporándole al mismo una etapa de Búsqueda Local encargada de explotar las regiones cercanas a las buenas soluciones ya encontradas. Con el objetivo de presentar un modelo más robusto que el modelo basado en Mallas Variables puro, que obtenga soluciones de mejor calidad que el mismo. Los resultados de esta investigación son la obtención de distintas variantes de un Algoritmo Memético con mejor comportamiento que el modelo basado en Mallas Variables puro para la resolución de problemas continuos.

ABSTRACT

The approximate methods for the resolution of problems of optimization one has come developing due to the high complexity that these problems have reached. A good exploitation of the good solutions and a correct exploration of the search space are indispensable factors to obtain solutions of quality. This investigation is guided to develop a model that combines in a balanced way these factors, taking like populational base, for the exploration the model based on Variable Meshes and incorporating to the same one a stage of Local Search in charge of exploiting the near regions to the good ones already solves. The aim of this research is presenting a more robust model that the based on Variable Meshes pure, that it obtains solutions of better quality that the pure model. The results of this investigation are the obtaining of different variants of a Memetic Algorithm with better behavior that the model based on Variable Meshes pure for the resolution of continuous problems.

TABLA DE CONTENIDOS

SÍNTESIS	I
ABSTRACT	I
INTRODUCCIÓN	1
1 HIBRIDACIÓN ENTRE ALGORITMOS POBLACIONALES Y META HEURÍSTICAS DE TRAYECTORIA	
SIMPLE	7
1.1 BÚSQUEDA LOCAL	8
1.2 META HEURÍSTICAS POBLACIONALES	10
1.4 ALGORITMOS MEMÉTICOS	13
1.4.1 <i>Diseño de AMs</i>	14
1.4.2 <i>Mecanismo de búsqueda global aportado por los APs (exploración)</i>	15
1.4.3 <i>Mecanismo de Búsqueda Local (explotación)</i>	15
1.4.4 <i>Mecanismos para la Hibridación</i>	16
1.4.5 <i>Enfoque de Lamarck vs. Enfoque de Baldwin</i>	17
1.4.6 <i>Ventajas del uso de los Algoritmos Meméticos</i>	17
2 ALGORITMO MEMÉTICO BASADO EN VMO	20
2.1 OPTIMIZACIÓN BASADA EN MALLAS VARIABLES	20
2.1.1 <i>Descripción general de la meta heurística</i>	20
2.2 MÉTODOS DE BÚSQUEDA LOCAL EMPLEADOS	26
2.2.1 <i>Método Solis West</i>	27
2.2.2 <i>Método Simplex de Nelder y Mead</i>	28
2.3 ALGORITMO MEMÉTICO BASADO EN VMO	32
2.3.1 <i>Consideraciones durante el diseño</i>	34
3 ANÁLISIS EXPERIMENTAL DE LOS RESULTADOS	35
3.1 TÉCNICAS ESTADÍSTICAS	35
<i>Descripción de los tests estadísticos utilizados</i>	37
3.2 MARCO EXPERIMENTAL	39
3.3 ANÁLISIS DE LOS RESULTADOS DE LOS TEST ESTADÍSTICOS PARA LAS DISTINTAS VARIANTES DE DISEÑO	40

<i>Comparación con las variantes que aplican la Búsqueda Local a un nodo:</i>	40
<i>Comparación con las variantes que aplican la Búsqueda Local a tres nodos:</i>	44
<i>Comparaciones con el modelo VMO-PURO.</i>	48
CONCLUSIONES	52
RECOMENDACIONES	53
REFERENCIAS BIBLIOGRÁFICAS	54
ANEXOS	59
ANEXO A. TABLAS DE LOS VALORES OBTENIDOS PARA CADA UNA DE LAS FUNCIONES DE PRUEBA DEL CEC´2005 POR LOS MODELOS ESTUDIADOS.	59
ANEXO B. BREVE DESCRIPCIÓN DE LAS FUNCIONES UTILIZADAS COMO PRUEBA.....	68

INTRODUCCIÓN

Hoy en día la optimización es un área que recibe mucho interés pues se presenta en gran cantidad de problemas, tanto en la ciencia como en la ingeniería, los cuales van desde problemas clásicos como los problemas de Viajante de Comercio (Traveling Salesman, TSP) 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, 2007a).

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):

- Objetivo. 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.

Problemas, como el antes mencionado actualmente están siendo abordados mediante procedimientos que, de manera eficiente, encuentran soluciones aproximadas, conocidos

como métodos heurísticos. Numerosas han sido las definiciones de algoritmo heurístico a continuación mostramos una de ellas (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.

Si se presta atención al comportamiento de los métodos heurísticos, se puede apreciar que estos dependen en gran medida del problema para el cual fueron diseñados. Debido a esto se han venido desarrollando 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 da la siguiente definición de meta heurística:

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. Los meta heurísticos proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de la inteligencia artificial, la evolución biológica y los mecanismos estadísticos.

Ante el amplio desarrollo obtenido por los métodos meta heurísticos y heurísticos tradicionales estos pueden ser subdivididos en dos grandes grupos, con características muy distintivas, por un lado están los mecanismos de explotación de las soluciones o métodos de Búsqueda Local (BL) y por otro los mecanismos de exploración de las soluciones o Algoritmos Poblacionales (APs). Los primeros logran con rapidez soluciones precisas en las inmediaciones del punto inicial, poseyendo una gran tendencia a quedar atrapados en óptimos locales y no el óptimoglobal, el cual constituye la mejor solución dentro del espacio de búsqueda. Además estos mecanismos poseen una gran

dependencia del punto inicial siendo algunos de sus representantes los métodos del Descenso/Ascenso (Luenberguer, 1984) los métodos de Cuasi-Newton (Luenberguer, 1984), los métodos del Gradiente (Luenberguer, 1984), el método Simplex de Nelder y Mead (Nelder and Mead, 1965b) , el Solis West (Solis and West, 1981b), la búsqueda Tabú (Glover and Laguna, 1997), el Recocido Simulado (Kirkpatrick et al., 1983), la Búsqueda Local Guiada (Voudouris and Tsang, 1995), etc.

Por otra parte los mecanismos de exploración de las soluciones poseen gran independencia de la naturaleza del espacio de soluciones pudiendo atravesar el espacio de búsqueda con múltiples máximos o mínimos locales y alcanzando una solución global, pero a su vez presentan inconvenientes para alcanzar soluciones precisas con rapidez pues no aprovechan las características locales del espacio de búsqueda. Algunos ejemplos los constituyen: Los Algoritmos Genéticos (AGs) (Goldberg, 1998), la Búsqueda Dispersa (Laguna and Martí, 2003), la Evolución Diferencial (Storn and Price, 1997), Algoritmos Basados en Estimación de Distribuciones (Lozano et al., 2006), la Optimización basada en las Colonias de Hormigas (Dorigo and Caro, 1999), Sistemas de Partículas (Kennedy and Eberhart, 1995) la Optimización Basada en Mallas Variables (VMO) (Puris, 2009a), etc.

Ante a estos dos criterios (exploración y explotación) se ha demostrado que el uso combinado de estas, posee propiedades beneficiosas durante el proceso de búsqueda, pues la incorporación de BL permite abordar la necesidad de precisión durante todo el proceso, a diferencia de un AP en el que el ajuste fino de las soluciones solo se produce en las etapas finales. En los últimos años se han venido desarrollando algoritmos que buscan abordar el problema de la búsqueda (búsqueda global pero con capacidad de alcanzar soluciones precisas rápidamente aprovechando las ventajas de la búsqueda local). Numerosas ha sido las denominaciones para este tipo de modelos que combinan ambos enfoques: Algoritmos Híbridos, Algoritmos Genéticos Locales, etc. Moscato (Moscato, 1999) emplea el término Algoritmos Meméticos para denominar a estos procedimientos, así como plantea la siguiente descripción de los mismos:

Algoritmo Memético es una población de agentes que alternan periodos de auto-mejora (mediante búsqueda local) con periodos de cooperación y competición (mediante selección).

El objetivo de los AMs es hacer que ambas componentes: La Búsqueda Local y el Algoritmo Poblacional trabajen de forma cooperativa para conseguir una sinergia entre ambos que permita mejorar el proceso de búsqueda. Numerosos han sido los diseños de AMs planteados. A continuación mencionamos algunos de los propuestos: El GLS_Based_Memetic_Algorithm propuesto en (Holstein and Moscato, 1999), Genetic_Local_Search (Aarts and Verhoeven, 1997) para resolver problemas del viajante del comercio, el AM con BL basada en el Operador de Cruce (AMCR-BLC²P) (Molina Cabrera, 2007a), Algoritmo híbrido con múltiples colonias (ACOR) (Herrera et al., 2009a), etc.

Basándonos en todo lo antes mencionado y dada las facilidades que muestran los AMs para la resolución de problemas complejos esta investigación estará encaminada al diseño de uno, tomando como AP base uno de los mencionados anteriormente: VMO, meta heurística desarrollada por el grupo de Inteligencia Artificial de la Universidad Central “Marta Abreu” de Las Villas y alguno o algunos de los métodos de BL para lograr mayor precisión en las soluciones.

Siendo nuestro **problema de investigación** el siguiente:

- ¿Cómo encontrar diferentes alternativas de hibridación del modelo basado en Mallas Variables con métodos BL, para obtener soluciones que mejoren en cuanto a calidad a las proporcionadas por el modelo basado en Mallas Variables puro en dominios continuos?

Nuestro problema puede desglosarse en las siguientes **preguntas de investigación**:

- ¿Será posible mejorar el comportamiento algoritmo basado en Mallas Variable incorporando una etapa de Búsqueda Local?

- ¿Qué valores de intensidad y profundidad proporcionan un buen desempeño del Algoritmo Memético?
- ¿La selección adecuada del conjunto de nodos involucrados en la Búsqueda Local influye en la calidad de los resultados obtenidos?

Teniendo en cuenta lo antes mencionado en **objetivo general** de nuestro trabajo será:

- Obtener un Algoritmo Memético basado en VMO con la incorporación de una fase de BL aportada por una meta heurística de trayectoria simple para dominios continuos.

Dicho objetivo general puede subdividirse en los siguientes **objetivos específicos**:

- Seleccionar los métodos de BL más utilizados internacionalmente en el diseño de Algoritmos Meméticos.
- Implementar las distintas alternativas de hibridación del modelo VMO con los métodos de BL seleccionados.
- Estudiar diversos factores como intensidad, profundidad y características de los nodos involucrados en la BL, determinantes en el comportamiento del AM.
- Analizar el comportamiento del Algoritmo Memético frente al modelo VMO puro.

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

La hibridación del modelo VMO con distintos Algoritmos de BL trae consigo un mejor comportamiento en cuanto a calidad de las soluciones con respecto al modelo VMO puro.

Esta hipótesis será validada si se comprueba que las técnicas mencionadas anteriormente realmente producen variaciones en la calidad de las soluciones obtenidas, teniendo en cuenta que el costo computacional que traiga consigo la incorporación de estas, sea justificado por la mejoras a los resultados que traen consigo.

Esta tesis está compuesta por un capítulo 1 en el cual se muestra un representación del marco teórico que nos ocupa, refiriéndose al uso de modelos aproximativos en la resolución de problemas de optimización, un capítulo 2 que aborda de manera específica la bases fundamentales y la confección en sí de nuestro modelo memético, un capítulo 3 en el cual se plantea todo el marco experimental utilizado, así como los resultados obtenidos mediante los procedimientos estadísticos empleados. Además esta investigación cuenta con un tópico de conclusiones, uno de recomendaciones y dos anexos donde se presentan datos empleados en la experimentación del capítulo 3.

1 HIBRIDACIÓN ENTRE ALGORITMOS POBLACIONALES Y META HEURÍSTICAS DE TRAYECTORIA SIMPLE

Incontables son los problemas que pueden expresarse como un problema de optimización continua. En muchos de estos encontrar numéricamente de forma exacta un óptimo global resulta imposible. Ante esta problemática se han venido empleando técnicas aproximativas que permiten obtener una solución bastante correcta dentro de límites que pueden ser por ejemplo el tiempo, la cantidad de evaluaciones, etc.

En la búsqueda relacionada a estos problemas existen dos factores fundamentales (Yagiura and Ibraraki, 2001):

- La **exploración** que no es más que el proceso de búsqueda en regiones no exploradas, si este factor en un algoritmo fuese insuficiente se podrían omitir, a la hora de realizar la búsqueda, regiones enteras corriendo el riesgo de nunca encontrar el óptimo global si este estuviese en una de las regiones obviadas.
- La **explotación** que consiste en realizar una búsqueda de manera intensa y minuciosa de mejores soluciones en un entorno cercano a buenas soluciones ya encontradas. Esto garantiza una mayor precisión.

Los Algoritmos Meméticos (AMs) (Moscato P, 1999) combinan ambas componentes haciendo que ambos trabajen de manera cooperativa para conseguir una continua sinergia entre ambos que permita mejorar el proceso de búsqueda (Molina Cabrera, 2007b).

Las bases fundamentales de los modelos Meméticos son las meta heurísticas poblacionales como elemento indispensable para realizar una buena exploración del espacio de búsqueda y los algoritmos de búsqueda local encargados de intensificar la búsqueda en las cercanías de las zonas exploradas.

1.1 Búsqueda Local.

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 1.1), 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 $y \in \text{vecindario}(x)$.

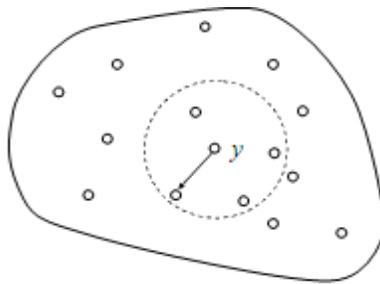


Figura 1.1. Espacio de búsqueda.

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:** 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:** 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:** 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:** 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.

1.2 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 (AEs) 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): Los EDAs 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): Los PSOs se caracterizan en que son modelos inspirados 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.

1.4 Algoritmos Meméticos.

En la práctica a la hora de resolver un problema de optimización ha quedado demostrado, que para obtener buenos resultados, es necesario que el algoritmo que se emplee utilice de manera equilibrada durante el proceso de búsqueda la exploración del espacio de búsqueda y la explotación de las soluciones ya encontradas.

A la hora de realizar una búsqueda las técnicas mencionadas anteriormente (Búsqueda Local y Algoritmos Poblacionales) presentan diferencias entre sí, las cuales constituyen tanto ventajas como desventajas pues la primera potencia la explotación de las soluciones ya encontradas, mientras que la segunda la exploración, lo que indica que con el uso de BL se pueden obtener soluciones precisas de forma eficiente y con los

Algoritmos Poblacionales (APs) se puede realizar una amplia búsqueda en dominios complejos.

El uso combinado de APs con BL ha demostrado traer buenos resultados y ha sido representado en la literatura bajo distintos nombres: Algoritmos Híbridos, Algoritmos Genéticos Locales y el termino más utilizado Algoritmos Meméticos cuya definición planteada por Moscato (Moscato, 1999) mostramos a continuación:

Algoritmo Memético es una población de agentes que alternan periodos de auto-mejora (mediante búsqueda local) con periodos de cooperación (mediante recombinación) y competición (mediante selección).

Los orígenes de los Algoritmos Meméticos (AMs) se remontan a finales de los años ochenta, época en la que el campo de la computación evolutiva estaba comenzando a afianzarse sólidamente. Fue en este escenario en el que surgió la idea básica que sustenta a los AMs: combinar conceptos y estrategias de diferentes meta heurísticas para intentar aunar las ventajas de las mismas.

No fue hasta mediados de los noventa cuando la formulación del así denominado Teorema de **No Free Lunch** por Wolpert y Macready 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.

1.4.1 Diseño de AMs.

Atacar un cierto problema de optimización con AMs requiere instanciar la plantilla genérica descrita anteriormente. Dado que el diseño de un algoritmo de búsqueda efectivo es en general tan complejo como los propios problemas que se desean resolver, nos encontramos ante la tesitura de tener que emplear directrices heurísticas para abordar dicho problema de diseño. A continuación se trataran algunas de estas directrices para algunos de los componentes esenciales de los AMs.

1.4.2 Mecanismo de búsqueda global aportado por los APs (exploración).

Como se ha planteado anteriormente un AM requiere de una componente encargada de realizar la búsqueda global, o sea explorar el espacio de búsqueda dirigido a las zonas más prometedoras.

A la otra de seleccionar un AP para constituir una componente de un AM se debe tener en cuenta que el mismo introduzca la diversidad para favorecer con ello la exploración del espacio de búsqueda, que logre evitar el quedar atrapados en óptimos locales y se produzca convergencia prematura.

Además debe ser capaz de aceptar las nuevas soluciones por el sistema de BL en la población pues por un lado debe ser receptivo a esas soluciones para que el mecanismo de BL pueda aplicarse de forma efectiva, pero al mismo tiempo debe evitar que se reduzca la diversidad de la población al aplicarse BL pues varias aplicaciones de la misma pueden conducir al mismo óptimo local.

Por otro lado es necesario que el AP empleado deba contribuir a que el proceso de búsqueda en cierto grado sea susceptible a una mejora de una parte de la población, o sea que el incorporar las nuevas soluciones precisas permita orientar mejor la búsqueda.

1.4.3 Mecanismo de Búsqueda Local (explotación).

La presencia de Búsqueda Local es una de las características distintivas de los AMs. A diferencia de la componente anterior el objetivo de esta es explotar las soluciones más prometedoras. Está definida por los siguientes elementos (Molina Cabrera, 2007b):

- **Función de transformación:** Permite generar una nueva solución a partir de una solución actual. Dicha función de transformación suele ser no determinista, por lo que a partir de una misma solución, podrían generarse distintas soluciones.
- El número de soluciones generadas a partir de una misma solución actual, de las que se escogerá la nueva siguiente solución. A este número se le denomina **anchura de la búsqueda local.**

- **Criterio de selección de la siguiente solución:** En general, es una práctica común escoger de entre las generadas y la solución actual aquella con mejor valor de la función objetivo (esquema de ascensión de colinas), aunque ciertos algoritmos como el Recocido Simulado permiten que una solución sea sustituida por otra con peor valor de la función objetivo.
- **Criterio de parada:** En optimización continua, dado que no se genera todo el vecindario, sino un subconjunto del mismo (indicado por la anchura), en ningún caso se puede saber si la solución actual es un óptimo local o no, por lo que un criterio factible de parada es un tiempo máximo, un número máximo de iteraciones, o número total de evaluaciones.

La elección del método de BL adecuado es un problema complejo, ya que depende del problema, y también puede depender del estado actual de la búsqueda.

1.4.4 Mecanismos para la Hibridación

A la hora de incorporar una etapa de BL es necesario seleccionar de manera minuciosa tanto aquellas soluciones a las cuales se les será aplicada, como el número máximo de evaluaciones que se emplearan al aplicársele a las mismas.

Para lograr esto es necesario asegurarse que los resultados obtenidos por la combinación seleccionada entre las componentes mencionadas anteriormente (APs y BL), son mejores que los obtenidos por cualquiera de estas de manera separada: o sea que la explotación aportada por BL compense la disminución de la exploración y que el uso de un AP permita aplicar BL a soluciones separadas entre sí, con lo que se reduzca el problema de la convergencia prematura. Por tanto es necesaria la selección de un buen esquema de hibridación que permita a los componentes (AP-BL) trabajar cooperativamente.

Según Hart (Hart, 1994) a la hora de confeccionar un modelo híbrido se deben tener en cuenta los siguientes factores para la aplicación de BL:

- **Frecuencia de aplicación de la BL:** Con qué probabilidad se aplica la BL a los nuevos individuos.
- **Conjunto de individuos sobre los que aplicar la BL:** Si la BL no se aplica a todos los individuos de la población, es necesario seleccionar el conjunto de individuos a mejorar.
- **La intensidad en cada aplicación de la BL:** Determinara el grado de explotación de las soluciones a las cuales se les aplique BL. A la hora de aplicar la BL es conveniente alcanzar un equilibrio entre el porcentaje de individuos sobre los que aplicar la BL y la intensidad de su aplicación.

1.4.5 Enfoque de Lamarck vs. Enfoque de Baldwin

Cuando se aplican métodos de búsqueda local para optimizar soluciones individuales, es necesario decidir de qué forma las soluciones obtenidas mediante estos métodos reemplazarán a las soluciones sobre las que estos fueron aplicados. Esta decisión puede basarse en uno de dos enfoques posibles: el enfoque de Lamarck o el de Baldwin.

En el enfoque de Baldwin el agente (nodo) sometido al proceso de BL recibe el fitness del agente final, pero no es reemplazado. Por otra parte el enfoque Lamarkiano plantea que el agente resultante de la aplicación del proceso de BL a otro, sustituye a este.

1.4.6 Ventajas del uso de los Algoritmos Meméticos.

- **Permite establecer de forma explícita (y ajustable) criterios que equilibren exploración y explotación:** Al definirse cada componente orientado a fomentar únicamente uno de los objetivos, y delegar la responsabilidad de un adecuado equilibrio entre ambos en el modelo de hibridación, se facilita el establecer mecanismos para equilibren el empleo de ambos factores.

- **Incorporación de información específica del problema:** Su estructura facilita el uso de información sobre los problemas para obtener buenos resultados (principalmente en el método de BL, aunque también puede darse en el AP, como el operador de cruce).
- **Incremento de la velocidad de convergencia:** Mientras que en un AP la explotación es realizada principalmente durante las últimas generaciones, en un AM se realiza durante todo el proceso (mediante la aplicación de la BL). Por tanto, el AM puede alcanzar el mismo grado de precisión más rápidamente, con lo que el AM puede alcanzar en menos evaluaciones resultados equivalentes a los obtenidos por el AP.
- **El AM puede alcanzar mejores resultados que el AP:** Los APs suelen presentar problemas de precisión. El uso de la BL permite mejorar rápidamente la precisión, con lo que se introducirían mejores soluciones.
- **Se alimentan de las investigaciones en otros campos:** Al poderse emplear distintos APs o BLs ya propuestos.

Diversos han sido los modelos de AMs propuestos, a continuación describimos brevemente algunos:

- **Algoritmo híbrido con múltiples colonias (ACOR)** (Herrera et al., 2009a): Se centra en algoritmos de colonias de hormigas para resolver problemas de optimización continua. En concreto se propone una extensión de este tipo de algoritmo hacia un método multi-colonia para incrementar la capacidad de exploración. Así mismo, para mejorar las soluciones de generadas por las hormigas artificiales incorpora métodos de BL como el Simplex y el Solis West.
- **Algoritmo Memético Basado en Encadenamiento de Búsquedas Locales para Problemas de Optimización Continua** (Herrera et al., 2009b): AM diseñado para métodos de BL que adaptan los parámetros para guiar la búsqueda y obtener así soluciones más efectivas. Este proceso de adaptación implica una mayor intensidad de aplicación de la Búsqueda Local lo que

dificulta su uso dentro de los AM. El algoritmo propone aplicar BL a una solución mediante el encadenamiento de consecutivas aplicaciones de BL

- **(AMCR-BLC²P)** (Molina Cabrera, 2007a): Algoritmo Memético que utiliza procedimientos de Búsqueda Local basados en el operador de cruce BLC. Dado que este operador produce hijos alrededor de sus padres puede considerarse como mecanismo para la generación de nuevas soluciones dentro de un método de BL.

2 ALGORITMO MEMÉTICO BASADO EN VMO

A la hora de conformar un Algoritmo Memético, los principales elementos a tener en cuenta son: el Algoritmo Poblacional que se va a encargar de la realización de la búsqueda global, la estrategia de Búsqueda Local y el mecanismo de hibridación que los combine. De momento centraremos nuestra atención en el primero siendo de manera específica el modelo VMO que, como se había dicho, constituye una componente esencial en nuestro diseño.

2.1 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, 2009b) se da una descripción detallada de la misma.

2.1.1 Descripción general de la meta heurística

La esencia del método VMO 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 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).

Seguidamente se muestra una descripción más detallada de cada uno de estos pasos:

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

Generación de nodos en dirección a los extremos locales: para calcular los vecinos más cercanos de cada nodo de la malla se utiliza como función de distancia la euclidiana, definida por:

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

El factor (r), que determina la cercanía del nuevo nodo al nodo actual o al extremo local, se calcula usando la ecuación siguiente:

$$r = \frac{1}{1 + |FO(n) - FO(ne)|} \quad (2.2)$$

esta forma de calcular r se puede utilizar lo mismo para minimizar que para maximizar una función objetivo, ya que solo mide la razón de separabilidad entre dos valores, no interesa el caso de estudio.

La función f para la generación de nuevos nodos a partir de cada nodo de la malla inicial, que no sea extremo local y el mejor vecino, se define por la ecuación:

$$n_f^*(x_j) \begin{cases} \text{ave}(n_i(x_j), nl_i(x_j)), & \text{si } (|\text{ave}(n_i(x_j), nl_i(x_j)) - nl_i(x_j)| > D) \text{ and } U[0,1] \leq r_i \\ nl_i(x_j) + U[-D, D], & \text{si } |\text{ave}(n_i(x_j), nl_i(x_j)) - nl_i(x_j)| \leq D \\ U[n_i(x_j), (n_i(x_j)), nl_i(x_j))] & \text{en otro caso} \end{cases} \quad (2.3)$$

Donde $\text{ave}(n_i(x_j), nl_i(x_j))$ representa el valor medio entre el nodo actual y el extremo local para la i -ésima dimensión y se calcula como:

$$\text{ave}(n(x_j), nl(x_j)) = \frac{n(x_j) + nl(x_j)}{2} \quad (2.4)$$

Además, $U[a(x_j) b(x_j)]$ denota un valor aleatorio en el intervalo $[a, b]$ y D es una cota de distancia adaptativa y se calcula según la ecuación 2.5

$$D = \begin{cases} \frac{am_{DOM}}{4}, & \text{si } l < 15\%M \\ \frac{am_{DOM}}{8}, & \text{si } 15\%M \leq l < 30\%M \\ \frac{am_{DOM}}{16}, & \text{si } 30\%M \leq l < 60\%M \\ \frac{am_{DOM}}{50}, & \text{si } 60\%M \leq l < 80\%M \\ \frac{am_{DOM}}{100}, & \text{si } l \geq 80\%M \end{cases}$$

(2.5)

Donde M denota el valor máximo de evaluaciones de la función objetivo y l la evaluación actual. En dependencia del por ciento del total que represente la evaluación actual, se definen valores de distancia que representan partes del intervalo permitido y am_{DOM} de la siguiente forma:

$$am_{DOM} = |\min_{value}| + |\max_{value}|$$

De forma general, la función f se comporta de manera que en el primer caso se obtiene para la i-ésima componente el valor medio entre el nodo actual y el extremo local en el segundo caso, se explora la vecindad del extremo local utilizando un valor en función de la distancia mínima permitida y, en el último caso, se genera un número aleatorio entre el valor medio y el extremo local.

Generación de nodos en dirección al extremo global: en este paso de la generación se crean nuevos nodos a partir de cada nodo de la malla inicial en dirección al extremo global (ng). El nuevo nodo (n_g^*) es generado usando la ecuación 2.6:

$$n_g^*(x_j) = g(n_i(x_j), ng(x_j), r_i) \quad (2.6)$$

La función g se representa de la siguiente manera:

$$n_g^*(x_j) = \begin{cases} \text{ave}[n_i(x_j), ng_i(x_j)], & \text{si } U[0,1] \leq r_i \\ U[\text{ave}[n_i(x_j), ng_i(x_j)], ng_i(x_j)], & \text{en otro caso} \end{cases}$$

(2.7)

Generación de nodos a partir de los más externos de la malla: En este paso se completa la cantidad total de nodos que debe tener la malla, a partir de los nodos fronteras. Para detectar este tipo de nodos se utiliza, para este caso de estudio, el valor de la norma de cada uno, definida por la ecuación:

$$\|n\| = \sqrt[2]{\sum_{i=1}^{nm} (n(x_j))^2} \quad (2.8)$$

Los nodos de mayor norma son los que están situados en el contorno (puntos más externos) de la malla inicial y los de menor norma se consideran los nodos que más cerca se encuentran del origen 0_m (puntos más internos). La función h permite generar nuevos nodos en dirección de las fronteras definidas para este caso de estudio, mediante las expresiones:

Para los nodos más externos:

$$n_h(x_j)^* = \begin{cases} nv(x_j) + w, & \text{si } nv(x_j) > 0 \\ nv(x_j) - w, & \text{si } nv(x_j) < 0 \end{cases} \quad (2.9)$$

Para los nodos más internos:

$$n_h(x_j)^* = \begin{cases} |nu(i) - w|, & \text{si } nu(x_j) > 0 \\ |nu(i) + w|, & \text{si } nu(x_j) \leq 0 \end{cases} \quad (2.10)$$

donde el desplazamiento w se calcula como:

$$w = (w_0 - w_f) \cdot \frac{M - j}{M} + w_f \quad (2.11)$$

donde el parámetro M y la variable j , están estrechamente relacionados y provocan las variaciones en el valor de w ; el primero representa el número total de iteraciones del algoritmo o un número máximo de evaluaciones de la función objetivo (criterio más utilizado). Por su parte, j denota el valor actual de uno de los criterios mencionados anteriormente. La variable w_0 representa el desplazamiento inicial y w_f el valor final de este (si $w_0 > w_f$ efecto decreciente). Para obtener desplazamientos decrecientes relacionados con las amplitudes de cada función, se desarrolló una propuesta adaptativa, donde $w_0 = am_{DOM}/10$ y $w_f = am_{DOM}/100$.

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

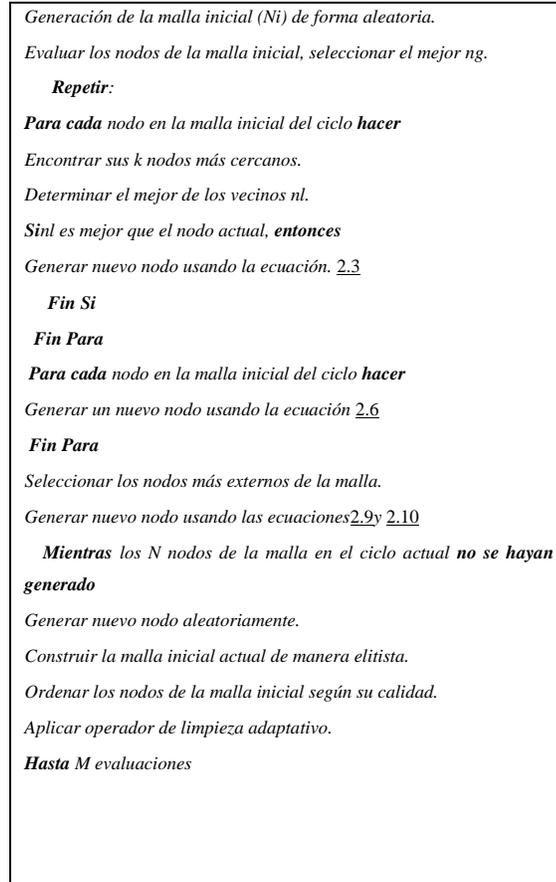


Figura 2.1 Esquema general de modelo VMO

2.2 Métodos de Búsqueda Local Empleados

En el diseño de un AM es necesario el uso de una componente que se encargue de la explotación de las soluciones ya encontradas a continuación mostramos una descripción más detallada de los métodos de Búsqueda Local (BL) empleados en nuestro modelo, son estos: el método Simplex de Nelder y Mead (Nelder and Mead, 1965a) y el método Solis y West (Solis and West, 1981a). Ambos métodos siguen un proceso iterativo en el que en cada paso se genera una nueva solución a partir de una solución actual que remplazara a esta en caso de que la mejore, pero se diferencian en la estrategia que emplean para ir obteniendo las nuevas soluciones. En ambos casos el proceso de búsqueda se repite hasta que se cumpla determinada condición de parada que en nuestro caso será un número máximo de evaluaciones de la función objetivo.

2.2.1 Método Solis West

Este método de BL que sigue un esquema de ascensión de colinas con un tamaño de salto adaptativo. En cada iteración se parte 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. Se caracteriza en varios aspectos:

- Las nuevas soluciones se obtienen mediante la suma de una variable de incremento (generada mediante una distribución normal $N(0, \rho)$ y una variable *bias*. Esta variable *bias* permite mantener una inercia (orientando la búsqueda hacia direcciones exitosas en iteraciones anteriores).
- En cada paso considera una dirección y si el resultado no es mejorado, se toma como nueva dirección de búsqueda, el sentido contrario a la dirección empleada hasta ese momento.
- Cuenta el número de éxitos (soluciones que mejoran a la actual) o fallos (soluciones peores que la actual) consecutivos. En función de unos valores máximos, el parámetro ρ es incrementado o decrementado, para aumentar o disminuir el espacio de búsqueda sobre la solución actual.

En la siguiente figura (2.2) se muestra un esquema del método Solis y West antes mencionado.

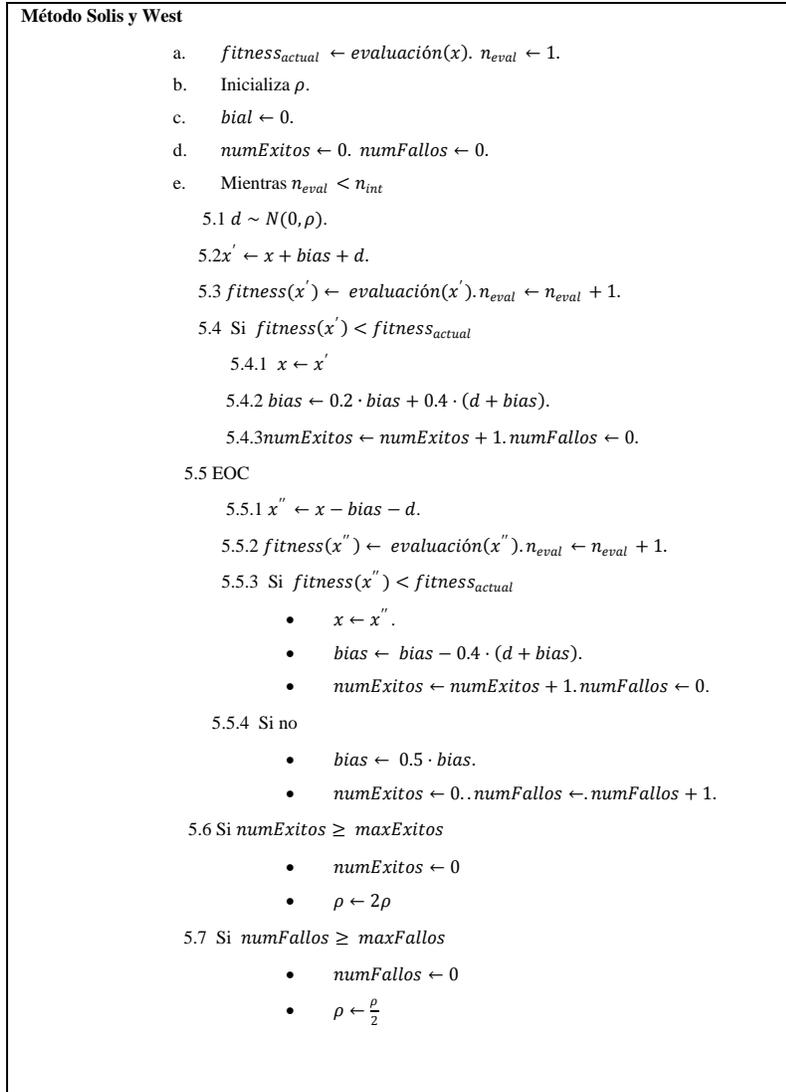


Figura 2.2: Esquema del método Solis-West de BL

2.2.2 Método Simplex de Nelder y Mead

Constituye un algoritmo de optimización ampliamente utilizado. Es debido a Nelder y Mead (Nelder and Mead, 1965b) y es un método para minimizar una función objetivo en un espacio multidimensional.

El método utiliza el concepto de un simplex que es una figura 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. Por ejemplo, un ingeniero que diseñe un puente colgante ha de elegir el grosor de los cables laterales, los cables más largos y del soporte que irá asfaltado. Estos elementos están ligados para un correcto diseño del puente y no es fácil imaginar el efecto en el cambio de cada uno de los espesores. El ingeniero puede usar el método Nelder y Mead para generar diseños de prueba, fijando los grosores de los elementos, que son probados en un modelo de ordenador que tiene en cuenta otros parámetros (vibraciones, vientos, materiales de construcción n...). Así se introduce una función, llamémosla “inestabilidad del puente” que depende de los grosores de los elementos con los que se construye, que interesa hacer mínima ante otros factores externos (vibraciones, vientos, etc.).

Como cada vez que se ejecuta este modelo que tiene en cuenta los factores externos se consume mucho tiempo de cálculo es importante variar los grosores con idea para no malgastar recursos. El método Nelder-Mead genera una nueva posición de prueba (valor de los grosores) extrapolando el comportamiento de la función en los vértices de un simplex. Así no es necesario calcular probar todos los valores posibles de la función (todos los grosores) sino que el algoritmo va reemplazando cada vez uno de los puntos de prueba ajustando con idea para encontrar la solución que minimiza la función más rápidamente. El modo más sencillo de hacerlo es reemplazar el peor punto con un punto reflejado en el resto de $N - 1$ puntos considerados como un plano (de ahí la extrapolación). Si este punto da mejor resultado, el algoritmo prueba a “estirarse” tomando los valores exponencialmente en un línea que contenga este punto. Por otra parte, si este nuevo punto no es mucho mejor que el valor previo, entonces estamos en un valle (buscamos un mínimo, como un gran hoyo) y el algoritmo encoge el simplex hacia el mejor punto.

La naturaleza del método es puramente geométrica y, en ausencia de derivadas, la optimización se efectúa mediante la aplicación sobre el simplex de las operaciones conocidas como reflexión, expansión y contracción. Sobre el espacio de soluciones en N dimensiones, el algoritmo genera iterativamente nuevos vértices del simplex que

reemplazan a los peores y hacen que éste se comprima progresivamente entorno a un mínimo o máximo en la región de búsqueda.

2.2.2.1 Reflexión, Expansión y Contracción

Cada vértice del simplex representa a una posible solución potencial al problema bajo análisis. A su vez, la precisión y calidad de cada uno de los vértices se mide en base a una función de coste o función de fitness. Cada vértice del simplex quedará representado por el vector

$$C = (M_1, F_1, \dots, M_i, F_i, \dots, M_n, F_n) \quad (2.13)$$

Donde M_i , F_i representan la amplitud y fase de la excitación de cada elemento de corriente del modelo equivalente.

Una vez se han inicializado de forma aleatoria los $N+1$ puntos que conforman el simplex, el algoritmo somete los vértices a una serie de operaciones con el objetivo de reemplazar aquellos que representan a las soluciones de menor calidad por vértices más precisos. Posteriormente y en cada iteración del algoritmo se selecciona en base al fitness el mejor de los vértices, C_m , y el peor : C_{p1} y se aplican sobre ellos las operaciones de reflexión, expansión y contracción . Como resultado, de cada una de las operaciones se obtienen nuevos puntos que irán reemplazando al peor de todos los vértices en la iteración en curso.

La operación de reflexión genera un nuevo vértice C^R de la siguiente manera:

$$C^R = (1 + \alpha)C^m - \alpha C_{p1} \quad (2.14)$$

Donde α es el coeficiente de reflexión y C^m es el centroide del simplex calculado excluyendo a C_{p1} . Si el Fitness en C^R mejora respecto al vértice C_{p1} entonces puede pensarse que una expansión más allá del punto de reflexión pudiera mejorar aún más el resultado. Esta situación se dará si estamos en una pendiente ascendente o descendente, según se trate de maximización o minimización, respectivamente. La expansión se rige por la siguiente expresión:

$$C^E = \beta C^R + (1 + \beta)C^M \quad (2.15)$$

Donde β es el coeficiente de expansión. Si el valor del Fitness en C^E mejora el valor obtenido en C_{p1} , se sustituye C_{p1} por C^E en el simplex y se repite el proceso. En caso contrario, si la expansión conduce a una solución peor, entonces es el vértice C^R el que se utiliza para remplazar al C_{p1} .

Ahora bien, en el supuesto de que la reflexión no depare mejoría alguna, entonces se precisa aplicar una contracción sobre el peor punto, dado que lo más probable es que el análisis se esté llevando a cabo sobre un valle. La contracción se rige por la expresión (2.16), donde γ representa el denominado coeficiente de contracción.

$$C^C = \gamma C_{p1} + (1 + \gamma)C^M \quad (2.16)$$

Si aun así existe alguna situación en la que no se contempla que con la contracción se mejore respecto al vértice C_{p1} , Nelder y Mead proponen la contracción del simplex en todas las direcciones y hacia el mejor punto, lo que algunos autores llaman encogimiento hacia C^M .

Las constantes α, β, γ representan el factor con el que se incide sobre el simplex. Los valores típicos $\alpha = 1.0$, $\beta = 2.0$ y $\gamma = 0.5$ demuestran ser los más idóneos y son los propuestos por defecto en las rutinas utilizadas como referencia para implementar el método en este trabajo.

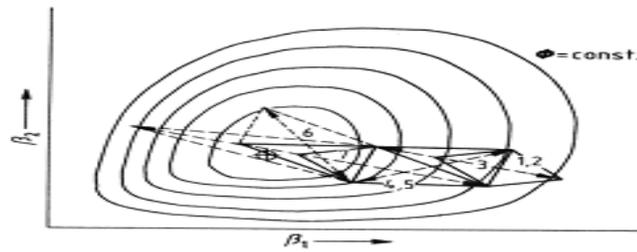


Figura 2.3: Representación del Método Simplex con geometría variable. Los números representan los números de reflexión, expansión o contracción.

2.2.2.2 Ventajas y Desventajas del método.

Como ventajas podemos mencionar que es un método heurístico. Se basa en consideraciones geométricas y no requiere el uso de derivadas de la función objetivo. Es de gran eficacia incluso para ajustar gran número de parámetros. Se puede usar con funciones objetivo muy sinuosas pues en las primeras iteraciones busca el mínimo más ampliamente y evita caer en mínimos locales fácilmente. Es fácil de implementar y usar, y sin embargo tiene una alta eficacia. Por otra parte se dice que converge más lentamente que otros métodos pues requiere mayor número de iteraciones.

2.3 Algoritmo Memético basado en VMO.

Como habíamos planteado a la hora de diseñar un Algoritmo Memético debemos realizar una minuciosa selección del Método Poblacional que servirá como base y garantizará la exploración y un Método de Trayectoria Simple o Búsqueda Local que explote las soluciones ya encontradas justificando la pérdida de exploración que trae consigo.

En nuestro caso el AP seleccionado fue el modelo VMO y los algoritmos de BL: Solis West y Simplex de Nelder y Mead, quedando conformado de la siguiente forma:

- 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.
- Aplicación de la BL a él o los nodos seleccionados.

A continuación se muestra un pseudocódigo del modelo (Figura 2.4):

Generación de la malla inicial (N_i) de forma aleatoria.
Evaluar los nodos de la malla inicial, seleccionar el mejor n_g .
Repetir:
Para cada nodo en la malla inicial del ciclo 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. 2.3
Fin Si
Fin Para
Para cada nodo en la malla inicial del ciclo hacer
Generar un nuevo nodo usando la ecuación 2.6
Fin Para
Seleccionar los nodos más externos de la malla.
Generar nuevo nodo usando las ecuaciones 2.9 y 2.10
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.
Si iteración seleccionada
Aplicar la fase de BL
Hasta M evaluaciones

Figura 2.4: Estructura general del Algoritmo Memético basado en VMO

2.3.1 Consideraciones durante el diseño.

Como habíamos planteado en tópicos anteriores de esta investigación existen aspectos imprescindibles a la hora de aplicar una fase de Búsqueda Local a un Algoritmo Poblacional, la intensidad, la frecuencia de aplicación y la profundidad entran dentro de estos importantes aspectos en el diseño de un Algoritmo Memético.

Luego de un minucioso estudio de la bibliografía y partiendo de la experiencia acumulada por varios creadores de AMs a continuación mostramos nuestras variantes de aplicación de la fase de BL a modelo VMO:

1. Se decidió experimentar con variantes que empleen un 10% y un 20% de las evaluaciones en la fase de BL.
2. Se aplicó dicha fase con profundidades variables de 10, 30.
3. Se diseñaron las siguientes variantes de selección de nodos para ser incorporados a la fase de BL:
 - C4 (Aplicar BL a un nodo seleccionado de manera aleatoria).
 - C2 (Aplicar BL al mejor nodo de cada iteración).
 - C3 (Aplicar BL al peor nodo de cada iteración).
 - C6 (Óptimo local que no generó un nuevo nodo en la función f del modelo VMO).
 - C21 (Aplicar BL a tres nodos seleccionados de manera aleatoria).
 - C19 (Aplicar BL a los tres mejores nodos de cada iteración).
 - C20 (Aplicar BL a los tres peores nodos de cada iteración).
 - C23 (Tres óptimos locales que no generaron un nuevo nodo en la función f del modelo).
 - Se empleo el enfoque Lamarkiano (El agente resultante de la aplicación del proceso de BL a otro sustituye a este en la malla).

3 ANÁLISIS EXPERIMENTAL DE LOS RESULTADOS

En este capítulo presentamos el análisis de los resultados de las distintas variantes de diseño de nuestro AM, así como una descripción de los test estadísticos utilizados. Además nos referimos a todo el marco experimental y el proceso experimentación desarrollado en este trabajo.

3.1 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 tests son el test de t-Student (para comparar dos algoritmos) y ANOVA (para comparar múltiples algoritmos).

Para poder aplicar dichos tests paramétricos con un mínimo de fiabilidad, los resultados de los algoritmos deben de seguir una distribución normal, y mantener una varianza homogénea. Cuando estas características son satisfechas los análisis paramétricos son correctos y seguros, y más sensibles (perceptivos) que los de la siguiente categoría, los tests no paramétricos.

- Los tests 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 tests son el test de Wilcoxon (para comparar los resultados de dos algoritmos) y el test de Bonferroni-Dunn o el test de Holm (para comparaciones de varios).

A lo largo de nuestra investigación haremos 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.

A continuación describimos el proceso para determinar diferencias significativas entre los modelos diseñados. Se trabajará con un margen de error de un 5% o lo que es lo mismo 0.05, en caso de emplearse otro valor será especificado. Más adelante se dará una descripción detallada de los test estadísticos mencionados:

1. Se aplica la prueba de Iman-Davenport (Iman and Davenport, 1980) para detectar diferencias entre un conjunto de algoritmos.
2. Si no se detectan diferencias se puede concluir que los algoritmos involucrados obtienen resultados que no difieren significativamente entre sí.
3. En otro caso, se pueden utilizar las pruebas de comparaciones múltiples (post-hoc). En este trabajo se aplica específicamente el test de Holm (Holm, 1979) para comparar una muestra de control (menor valor de ranking) con el conjunto restante de algoritmos (más de dos).
4. En caso de que sea necesario establecer una comparación entre dos algoritmos, se realiza una prueba de Wilcoxon (Wilcoxon, 1945).

En todos los test de comparación y con el fin de identificar el mejor algoritmo del grupo para futuras comparaciones, se decide escoger el modelo de mejor comportamiento (menor Ranking) en caso que las diferencias no sean significativas.

Descripción de los tests estadísticos utilizados.

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

Test de Friedman: Constituye un análisis de varianza de segunda vía para variables apareadas o de carácter no paramétrico. Haya el orden de los resultados observados por algoritmo (r_j para el algoritmo j con k algoritmos) para cada función, asignando al mejor de ellos el orden 1, y al peor el orden k . Bajo la hipótesis nula, que se forma a partir de suponer que los resultados de los algoritmos son equivalentes y, por tanto, sus rankings son similares, el estadístico de Friedman:

$$x_F^2 = \frac{12N}{k \cdot (k+1)} \left[\sum jR_j^2 - \frac{k \cdot (k+1)^2}{4} \right],$$

se distribuye aproximadamente acorde a χ^2 con $k - 1$ grados de libertad, siendo $R_j = \frac{1}{N} \sum_i r_i^j$, y N el número de casos. Los valores críticos del estadístico de Friedman coinciden exactamente con los establecidos en la distribución χ^2 cuando $N > 10$ y $k > 5$.

En nuestro trabajo consideramos aplicar una variante de esta prueba conocida como el **Test de Iman-Davenport**, pues sus autores demostraron que la función de Friedman x_F^2 era excesivamente conservativa y definieron una variante con un mejor comportamiento, indicada en la ecuación siguiente:

$$F_F = \frac{(N-1)x_F^2}{N \cdot (k-1) - x_F^2}$$

Bajo las mismas condiciones de equivalencia el valor de Iman-Davenport se distribuye acorde a una distribución F con $k - 1$ y $(k - 1)(N - 1)$ grados de libertad. Por tanto si el valor del test es mayor que el valor correspondiente de la distribución F , entonces esto quiere decir que existe al menos una diferencia estadística relevante entre los algoritmos considerados. En nuestro caso llegamos a esta misma conclusión comparando el valor ajustado del Test (p -value) con nivel de confianza (α), de manera que si $p\text{-value} < \alpha$ existe diferencia significativa en el grupo de control, en caso contrario no la hay.

Este test , como el de Friedman, sólo muestra que existe alguna diferencia significativa, pero no indica cual o cuales algoritmos la poseen con respecto a otros, para eso, existe una serie de test de comparaciones múltiples (post-hoc) que pueden ser aplicados en ese caso para identificar entre qué algoritmos existe una diferencia relevante. Dos de estos tests son los siguientes.

Test de Holm: Un test que prueba secuencialmente las hipótesis ordenadas según su significación. Los valores de p se ordenan por p_1, p_2, \dots , de tal forma que $p_1 \leq p_2 \leq \dots \leq p_k - 1$. El método de Holm compara cada p_{ki} con $\alpha/(k - i)$ comenzando desde el valor de p más significativo. Si p_1 es menor que $\alpha/(k-1)$, la correspondiente hipótesis se rechaza y nos permite comparar p_2 con $\alpha/(k-2)$. Si la segunda hipótesis se rechaza, se continúa el proceso de comparación. En cuanto una determinada hipótesis no puede ser rechazada, todas las restantes se mantienen como aceptadas. El estadístico para comparar el algoritmo i -ésimo con el j -ésimo es:

$$z = (R_i - R_j) / \sqrt{\frac{k \cdot (k+1)}{6 \cdot N}}$$

El valor z se utiliza para encontrar la probabilidad correspondiente a partir de la Tabla de la distribución normal, la cual es comparada con el correspondiente valor de α .

Test de Ranking de Signos de Wilcoxon: A la hora 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.

3.2 Marco experimental.

Como se había planteado el conjunto de nodos involucrados, la intensidad y profundidad con que se aplique la búsqueda local pueden influir considerablemente en los resultados obtenidos. A continuación mostramos un análisis de algunos de las variantes de nuestro modelo, teniendo en cuenta estos factores y analizados con los test estadísticos mencionados en el apartado anterior. Para dichos análisis, se confeccionaron los siguientes grupos de comparación tanto para modelo que emplea un 10% de las evaluaciones en la fase de BL, como los que emplean un 20% con ambos métodos de Búsqueda Local (Solis West y Simplex) :

1. Hallar la mejor variante de hibridación con ambos métodos de BL (Solis West y Simplex) cuando es seleccionado un solo nodo para aplicársele la fase de BL.
2. Hallar la mejor variante de hibridación con ambos métodos de BL (Solis West y Simplex) cuando son seleccionados tres nodos para aplicárseles la fase de BL.
3. Los dos mejores de “1”, los dos mejores de “2” y el VMO puro sin la incorporación de una fase de BL.

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

Los métodos de Búsqueda Local: Solis-West y Simplex, como se reflejó en el Capítulo 2 fueron los seleccionados para conjuntamente con el modelo VMO conformar las bases de nuestro Algoritmo Memético.

Seguidamente mostramos los estudios realizados sobre las diferentes variables de aplicación de los mismos en el diseño del AM. Para ello se utilizarán como prueba las funciones del **CEC 2005**¹ (Anexo B).

Comparación con las variantes que aplican la Búsqueda Local a un nodo:

Se realizó un estudio comparativo entre las distintas variantes diseñadas de aplicación de la fase de BL empleando el método *Simplex* para intensidades de 10% y 20% en ambos casos con una profundidad de 30 (Anexo A, Tabla 1, Tabla 2).

La Tabla 3.1 muestra los valores de los rankings de estos modelos destacándose el de menor valor en cada grupo (se empleará la notación planteada en el apartado 2.3.1 de capítulo anterior para referirse a las variantes de diseño).

<i>Algoritmo</i>	<i>Intensidad de un 10%</i>	<i>Intensidad de un 20%</i>
VMO-C2	2.675	2.900
VMO-C3	1.950	2.325
VMO-C4	2.450	2.000
VMO-C6	2.925	2.775

Tabla 3.1 Ranking de los modelos VMO-Simplex.

¹ Congress on Evolutionary Computation 2005 organizado por la *IEEE*.

Luego la Tabla 3.2 muestra los resultados del test de Iman-Davenport para cada grupo, notándose que no existen diferencias significativas entre las variantes algorítmicas comparadas para cada intensidad ($p\text{-value} > \alpha$).

<i>Intensidad</i>	<i>p-value</i>	<i>Valor de α</i>	<i>Hipótesis</i>
10%	0.0994	0.050	Aceptada
20%	0.0994	0.050	Aceptada

Tabla 3.2 Resultados de la prueba de Iman-Davenport.

A pesar de no diferir significativamente de el resto en sus respectivos grupos decidimos establecer una comparación entre los modelo VMO-C3 cuando se emplea un 10% de las iteraciones en la fase de BL y VMO-C4 cuando se emplea un 20% en dicha fase (menor ranking dentro de cada grupo). La Tabla 3.3 muestra los resultados de la prueba de Wilcoxon para $\alpha=0.05$. Además en dicha Tabla aparecen los mencionados R^+ (suma de rankings positivo) y R^- (suma de rankings negativo) quienes comparten la columna con las variantes de diseño a las cuales favorecen respectivamente.

R^-	R^+	Sig.	Hipótesis
VMO-C4 (20%)	VMO-C3 (10%)	Asintótica	
74.000	97.000	0.616	Aceptada

Tabla 3.3 Resultados de la prueba de Wilcoxon.

Puede apreciarse entre los dos modelo las diferencias no son significativas, aun así posee un mejor comportamiento el modelo VMO-C3, por tanto a la hora de aplicar una fase de BL mediante el método Simplex al modelo VMO seleccionando un nodo para la aplicación de la misma, la mejor variante entre las analizadas es aplicar dicha fase con un 10% de intensidad, un 30 de profundidad al peor nodo de la iteración.

Para el método *Solis West* se realizaron los mismos estudios para las variantes de aplicación de la fase de BL sobre un nodo obteniéndose los resultados siguientes (*Anexo A, Tabla 5, Tabla 6*):

<i>Algoritmo</i>	<i>Intensidad de un 10%</i>	<i>Intensidad de un 20%</i>
VMO-C2	2.925	3.100
VMO-C3	2.325	2.150
VMO-C4	2.525	1.975
VMO-C6	2.225	2.775

Tabla 3.4 Ranking de los modelos VMO-Solis-West

Como refleja la Tabla 3.4 para un intensidad de un 10% y profundidad de 30 la alternativa VMO-C6 (Óptimo local que no generó un nuevo nodo en la función f del modelo VMO) presenta mejor rankings, por otra parte si se aplica BL con una intensidad de un 20% la variante VMO-C4 (Aplicar BL a un nodo seleccionado de manera aleatoria) posee el mejor de los valores.

Al aplicárseles la prueba de Iman-Davenport para detectar diferencias dentro de los grupos de comparación, se detecto que en el primero de estos (intensidad de un 10%) no existen diferencias significativas, pero en el segundo de estos sí se detectaron. La Tabla 3.5 refleja lo antes planteado.

<i>Intensidad</i>	<i>p-value</i>	<i>Valor de α</i>	<i>Hipótesis</i>
10%	0.333	0.050	Aceptada
20%	0.015	0.050	Rechazada

Tabla 3.5 Resultados de la prueba de Iman-Davenport.

Al ser rechazada la hipótesis nula en el segundo grupo, se acepta la hipótesis alternativa, o sea, la existencia de diferencias significativas dentro del grupo, para detectar entre que modelos existen tales diferencias de significación se aplicó el test de Holm cuyos resultados aparecen en la Tabla 3.6 en la cual se puede apreciar que el modelo VMO-C4 para un 20% de intensidad (seleccionado como algoritmo de control al poseer menor ranking) difiere significativamente del VMO-C2 para iguales valores de intensidad.

Algoritmo	$Z = (R_i - R_j) / \sqrt{\frac{k - (k+1)}{6 - N}}$	p-value	Holm	Hipótesis
VMO-C2	2.756	0.006	0.017	Rechazada
VMO-C6	1.960	0.050	0.025	Aceptada
VMO-C3	0.429	0.668	0.05	Aceptada

Tabla 3.6 Tabla de Holm para $\alpha=0.05$.

Se aplicó la prueba de Wilcoxon a los modelos VMO-C6 con intensidad de 10% y el modelo VMO-C4 con 20% para detectar qué modelo VMO-Solis-West con BL aplicada a un nodo posee mejor comportamiento y detectar si ese comportamiento es significativamente notable.

R⁻	R⁺	Sig. Asintótica	Hipótesis
VMO-C4 (20%)	VMO-C6 (10%)		
81.500	89.500	0.862	Aceptada

Tabla 3.7 Resultados de la prueba de Wilcoxon.

Como se puede apreciar los comportamientos son muy similares presentándose una pequeña mejoría por parte de la variante VMO-C6 con intensidad y profundidad de 10%.

Hasta el momento podemos concluir que aplicando la fase de BL al modelo VMO las mejores variantes de las estudiadas para cada uno de los métodos de BL son:

1. VMO-Simplex con BL aplicada con un 10% de intensidad al peor nodo de la iteración.
2. VMO-SOLIS-WEST con BL aplicada con intensidad de 10% a un nodo que no fue utilizado para generar otro al aplicársele el primer operador del modelo VMO.

Comparación con las variantes que aplican la Búsqueda Local a tres nodos:

A continuación mostramos un proceso de experimentación similar al anterior, algunas incrementando ahora el número de nodos involucrados en la fase de BL de uno a tres y disminuyendo la intensidad a 10.

Se aplicó la fase de BL con el método *Simplex*, ahora incorporando dos nodos más a ser sometidos a dicha fase y como puede apreciarse en la Tabla 3.9 los modelos VMO-C19 para una intensidad de un 10% y VMO-C21 para una intensidad de 20% muestran los mejores rankings en sus respectivos grupos de comparación (*Anexo A, Tabla 3, Tabla 4*).

Algoritmo	Intensidad de un 10%	Intensidad de un 20%
VMO-C19	2.325	2.800
VMO-C20	2.550	2.475
VMO-C21	2.525	2.125
VMO-C23	2.600	2.600

Tabla 3.9 Ranking de los modelos VMO-Simplex.

Al aplicarse la prueba de Iman-Davenport fueron aceptadas las hipótesis nulas en ambos grupos y por tanto se comprobó la no existencia de diferencias significativas entre

los modelos involucrados. Eso es corroborado por la Tabla 3.10 la cual muestra los resultados de dicho test.

<i>Intensidad</i>	<i>p-value</i>	<i>Valor de α</i>	<i>Hipótesis</i>
10%	0.918	0.050	Aceptada
20%	0.416	0.050	Aceptada

Tabla 3.10 Resultados de la prueba de Iman-Davenport.

Siguiendo procedimiento establecido de análisis se aplico el test de Wilcoxon para determinar el modelo VMO-Simplex con BL aplicada a tres nodos de mejor comportamiento. En la Tabla 3.11 puede apreciarse que a pesar de no ser significativa hay una mejora a favor del modelo que incorpora la BL con un 20% de intensidad.

R^-	R^+	Sig.	<i>Hipótesis</i>
VMO-C21 (20%)	VMO-C19 (10%)	Asintótica	
123.000	48.000	0.102	Aceptada

Tabla 3.11 Resultados de la prueba de Wilcoxon.

A la hora de incorporar la fase de BL aplicada a tres nodos mediante el método de trayectoria simplex *Solis West* pudo detectarse (Tabla 3.12) que para una intensidades de un 10% y un 20% obtuvieron mejores rankings dentro sus grupos de control las variantes VMO-C23 (aplicar BL a tres nodos que no fueron utilizados para generar otros al aplicársele el primer operador del modelo VMO) y VMO-C21 (Aplicar BL a tres nodos seleccionados de manera aleatoria) (*Anexo A, Tabla 7, Tabla 8*).

<i>Algoritmo</i>	<i>Intensidad de un 10%</i>	<i>Intensidad de un 20%</i>
VMO-C19	2.575	2.975
VMO-C20	2.725	2.300
VMO-C21	2.425	1.900
VMO-C23	2.275	2.825

Tabla 3.12 Ranking de los modelos VMO-Solis-West.

Se aplicó la prueba de Iman-Davenport para detectar diferencias en los grupos detectándose la existencia de las mismas solamente en el segundo de estos. La siguiente Tabla (3.13) muestra los resultados de dicha prueba.

<i>Intensidad</i>	<i>p-value</i>	<i>Valor de α</i>	<i>Hipótesis</i>
10%	0.727	0.050	Aceptada
20%	0.020	0.050	Rechazada

Tabla 3.13 Resultados de la prueba de Iman-Davenport.

Al detectarse por la prueba de Iman-Davenport diferencias significativas existentes en el grupo en el cual se aplicó la fase de BL con una intensidad de 20% fue necesario aplicar el test de Holm para detectar entre cuales modelos existía tal diferencia seleccionándose el VMO-C21 como algoritmo de control. La siguiente Tabla muestra los resultados de dicha prueba para $\alpha=0.05$.

Algoritmo	$Z = (R_i - R_j) / \sqrt{\frac{k - (k+1)}{6 - N}}$	p-value	Holm	Hipótesis
VMO-C19	2.633	0.008	0.016	Rechazada
VMO-C23	2.266	0.023	0.025	Aceptada
VMO-C20	0.980	0.327	0.050	Aceptada

Tabla 3.14 Tabla de Holm para $\alpha=0.05$.

Como se aprecia el modelo VMO-C21 difiere de manera significativa solamente de la variante VMO-C19 (Aplicar BL a los tres mejores nodos de cada iteración).

Luego de este análisis se aplicó la prueba de Wilcoxon para determinar el modelo de mejor comportamiento dentro de los comprendidos en el grupo de variantes VMO-SOLIS-WEST con BL aplicada a tres nodos. Para ello se utilizaron las mejores variantes para intensidades de 10% y 20% (VMO-C23 y VMO-C21 respectivamente) obteniéndose los siguientes resultados mostrados en la Tabla 3.15.

R^-	R^+	Sig. Asintótica	Hipótesis
VMO-C21 (20%)	VMO-C23 (10%)		
125.000	46.000	0.085	Aceptada

Tabla 3.15 Resultados de la prueba de Wilcoxon

Como se aprecia es aceptada la hipótesis nula o sea no hay diferencias significativas, pero se comporta de mejor manera la variante VMO-C21 para una intensidad de un 20%.

Se puede de manera parcial concluir entonces que para la hibridación con ambos métodos las mejores alternativas son:

1. VMO-Simplex con BL con BL aplicada con intensidad de 20% a tres nodos seleccionados de manera aleatoria.
2. VMO-SOLIS-WEST con BL aplicada con intensidad de 20% a tres nodos seleccionados de manera aleatoria.

Comparaciones con el modelo VMO-PURO.

Seguidamente con fase final de la experimentación presentada en esta memoria presentamos una comparación entre quienes resultaron las mejores variantes de diseño de hibridación presentadas en los apartados 3.2.1.1 y 3.2.1.2 con el modelo VMO-PURO o sea sin la incorporación una fase de BL.

Comparación con las variantes de hibridación con BL aplicada a un nodo:

Se creó grupo de comparación integrado por: la mejor variante de VMO-Simplex (C3 para un 10% de intensidad y profundidad de 30), la mejor variante de VMO-Solis-West (C6 para un 10% de intensidad y profundidad de 30) y el modelo VMO-Puro. La siguiente Tabla muestra los respectivos rankings para las alternativas mencionadas (*Anexo A, Tabla 1, Tabla 5, Tabla 9*).

Algoritmo	Ranking
VMO-Puro	2.075
VMO-Simplex	1.950
VMO-Solis-West	1.975

Tabla 3.16 Ranking de los modelos VMO-Puro y las mejores variantes de VMO con la fase de BL aplicada a un nodo.

Como se aprecia ambas variantes de hibridación poseen mejor ranking que el modelo sin la incorporación de la fase de BL destacándose el modelo que emplea el método Simplex.

Seguidamente se aplicó una prueba de Iman-Davenport para detectar si las diferencias entre el grupo son estadísticamente significativas. La Tabla 3.17 muestra que no existen diferencias de significación.

<i>p-value</i>	<i>Valor de α</i>	<i>Hipótesis</i>
0.920	0.050	Aceptada

Tabla 3.17 Resultados de la prueba de Iman-Davenport.

Comparación con las variantes de hibridación con BL aplicada tres nodos:

Este grupo de comparación lo integraron los modelo VMO-Simplex (C21 para un 20% de intensidad y profundidad de 10), VMO-Solis-West (C21 para un 20% de intensidad y profundidad de 10), y el modelo VMO-Puro.

La Tabla de rankings que a continuación mostramos refleja que nuevamente ambas variantes de hibridación poseen mejor ranking que el modelo puro siendo esta vez la variante que emplea el método de BL Solis West la de mejor valor (*Anexo A, Tabla 4, Tabla 8, Tabla 9*).

<i>Algoritmo</i>	<i>Ranking</i>
VMO-Puro	2.050
VMO-Simplex	2.000
VMO-Solis-West	1.950

Tabla 3.18 Ranking de los modelos VMO-Puro y las mejores variantes de VMO con la fase de BL aplicada a tres nodos.

Al aplicarse la prueba de Iman-Davenport (Tabla 3.18) se pudo comprobar que las diferencias encontradas en el grupo no eran estadísticamente significativas.

<i>p-value</i>	<i>Valor de α</i>	<i>Hipótesis</i>
0.953	0.050	Aceptada

Tabla 3.19 Resultados de la prueba de Iman-Davenport.

Se puede concluir que si se aplica la fase de BL al modelo VMO las mejores variantes de aplicación (agrupadas por número de agentes involucrados) son:

- 1. Un nodo:** VMO-Simplex con BL aplicada al peor nodo de la malla para un 10% de intensidad y profundidad de 30.
- 2. Tres nodos:** VMO-Solis-West con BL aplicada a tres nodos seleccionados de manera aleatoria para un 20% de intensidad y profundidad de 10.

Selección de la mejor variante de hibridación dentro de las estudiadas:

Como se planteó anteriormente las mejores variantes de aplicación de la fase de BL al modelo VMO agrupadas por cantidad de agentes involucrados son: para un nodo VMO-Simplex con BL aplicada al peor nodo de la malla para un 10% de intensidad y profundidad de 30 y para tres nodos VMO-Solis-West con BL aplicada a tres nodos seleccionados de manera aleatoria para un 20% de intensidad y profundidad de 10. A continuación mostramos una comparación entre ambas para detectar la de mejor comportamiento y si difiere significativamente de la otra. La Tabla 3.20 muestra los resultados del test de Wilcoxon para esta comparación.

R^-	R^+	Sig.	Hipótesis
VMO-Solis-West-C21 (20%)	VMO-Simplex-C3 (10%)	Asintótica	
95.000	76.000	0.679	Aceptada

Tabla 3.20 Resultados de la prueba de Wilcoxon.

Como se aprecia entre ambas variantes las diferencias no son estadísticamente significativas lo que demuestra que ambas poseen comportamientos semejantes.

CONCLUSIONES

Como resultado de esta investigación se desarrollaron diversas alternativas de diseño para un Algoritmo Memético basado en el modelo VMO; cumpliéndose de esta forma el objetivo general planteado, ya que:

1. Se realizó un estudio minucioso de las posibles estrategias de BL a emplear en nuestro diseño de AM, seleccionando para el mismo los métodos estocásticos: Solis West y Simplex de Nelder y Mead.
2. Se diseñaron las distintas variantes de hibridación del modelo VMO con cada uno de los métodos de BL seleccionados.
3. Se establecieron distintos valores para factores influyentes en los resultados del AM como intensidad, profundidad, nodos involucrados.
4. Se obtuvieron las mejores variantes de hibridación con ambos métodos de BL para los distintos valores de intensidad, profundidad y nodos involucrados.
5. Se comprobó que las variantes de hibridación presentan un mejor comportamiento que el modelo puro, aunque no de manera significativa.
6. Se pudo determinar que no existen diferencias significativas entre la mejor variante de aplicación de la fase de BL sobre UN NODO respecto a la mejor variante cuando dicha fase es aplicada sobre TRES NODOS.

RECOMENDACIONES

Este campo de investigación 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 del AM basado en VMO con la utilización de otros métodos de BL (métodos numéricos).
2. Analizar nuevas alternativas de selección de los nodos, intensidad y profundidad de la búsqueda local.
3. Estudiar operadores de cruce basados en entorno, con características propias de explotación, con el fin de ser incorporados en el proceso de expansión del modelo original.

REFERENCIAS BIBLIOGRÁFICAS

1. Aarts, E. and G. Verhoeven (1997). HandBook of Evolutionary Computation, Oxford University Press.
2. Ballester, P. and J. Carter (2004). An effective real-parameter genetic algorithm with parent centric normal crossover for multimodal optimization Genetic and Evolutionary Computation Conference LNCS.
3. Bello, R. and A. Puris (2009). Optimización basada en Mallas Dinámicas. Su aplicación en la solución de problemas de optimización continuos. CD de Memoria del VI Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB'09). Málaga: 441-448.
4. Coigt, H., H. Muhlenbein, et al. (1995). Fuzzy recombination for the breeder genetic algorithm. Sixth International Conference on Genetic Algorithms, San Mateo CA: Morgan Kaufmann.
5. Darwin, C. (1859). On the Origin of Species, London: John Murray.
6. Deb, K. and R. B. Agrawal (1995). "Simulated Binary Crossover for Continuous Search Space." Complex Systems **9**: 115-148.
7. Dorigo, M. and G. D. Caro (1999). The Ant Colony Optimization meta-heuristic. New Ideas in Optimization. D. Corne, M. Dorigo and F. Glover. London UK, McGraw-Hill: 11-32.
8. Dunn, O. J. (1961). "Multiple comparisons among means." Journal of the American Statistical Association **56**: 52-64.
9. Eiben A., Hinterding R., et al. (1999). "Parameter Control in Evolutionary Algorithms." IEEE: Transactions on Evolutionary Computation **3**(2): 124-141.
10. García, S., D. Molina, et al. (2009). "A Study on the Use of Non-Parametric Tests for Analyzing the Evolutionary Algorithms' Behaviour: A Case Study on the CEC'2005 Special Session on Real Parameter Optimization." Journal of Heuristics **In press**.

11. Glover, F. (1977). "Heuristics for Integer Programming Using Surrogate Constraints." Decision Sciences **8**(1): 156-166.
12. Glover, F. and G. Kochenberger, Eds. (2003). Handbook of Metaheuristics., Kluwer Academic Publishers.
13. Glover, F. and M. Laguna (1997). Tabu Search. MA, Kluwer Academic Publishers.
14. Goldberg, D. E., Ed. (1998). Genetic Algorithms in Search. Optimization and Machine Learning. University of Alabama, Addison-Wesley Publishing Company.
15. Hart, W. (1994). Adaptive Global Optimization With Local Search. San Diego Universidad de California.
16. Herrera, F., P. Cardoso, et al. (2009). ACOR híbrido con multiples colonias para problemas de optimizacion continua. CD de Memoria del VI Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB'09). Málaga: 465-472.
17. Herrera, F., M. Lozano, et al. (2003). "A Taxonomy for the Crossover Operator for Real Coded Genetic Algorithms: An Experimental Study." International Journal of Intelligent Systems **18**: 309-338.
18. Herrera, F., D. Molina Cabrera, et al. (2009). Algoritmo Memético Basado en Encadenamiento de Busquedas Locales para Problemas de Optimización Continua. CD de Memoria del VI Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB'09). Málaga: 481-486.
19. Holm, S. (1979). "A simple sequentially rejective multiple test procedure." Scand Journal Stat **6**: 65-70.
20. Holstein, D. and P. A. Moscato (1999). Memetic algorithms using guided local search: A case study. . New Ideas in Optimization.
21. Iman, R. L. and J. M. Davenport (1980). "Approximations of the critical region

- of the Friedman statistic." Commun Stat **18**: 571-595.
22. Kennedy, J. and R. C. Eberhart (1995). Particle swarm optimization. IEEE International Conference on Neural Networks, Piscataway, New York, USA.
23. Kirkpatrick, S., C. Gellat, et al. (1983). "Optimization by simulated annealing." Science **220**(4598): 671–680.
24. Krasnogor, N. and J. E. Smith (2000). A Memetic Algorithm with Self-Adapting a Local Search: TSP as a Case Study. Proceedings of the 200 International Conference on Genetic and Evolutionary Computation.
25. Laguna, M. and R. Martí (2003). Scatter Search. Methodology and Implementation in C, Kluwer Academic Publishers.
26. Lozano, J., P. Larrañaga, et al., Eds. (2006). Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms., Springer Verlag.
27. Luenberguer, D. (1984). "Programación Lineal y no Lineal".
28. Luengo, J., S. García, et al. (2009). "A Study on the Use of Statistical Tests for Experimentation with Neural Networks: Analysis of Parametric Test Conditions and Non-Parametric Tests." Expert Systems with Applications **36**: 7798-7808.
29. Martínez, Y., A. Puris, et al. (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.
30. Merz, P. (2000). Memetic Algorithms for Combinational Optimization Problems: Fitness Landscapes and Effective Search Strategies. Gesamthochschule Siegen.
31. Molina Cabrera, D. (2007). Algoritmos Meméticos con Aplicación Adaptativa de la Búsqueda Local para Optimización Continua. Departamento de Ciencias de la Computación e Inteligencia Artificial. Granada, Universidad de Granada: 258.

32. Moscato, P. A. (1999). Memetic Algorithms: a Short Introduction. D. Corne, M. Dorigo and F. Glover. London, McGraw-Hill.
33. Muhlenbein, H., M. Schomisch, et al. (1991). The parallel genetic algorithm as function optimizer. Fourth International Conference on Genetic Algorithms, San Mateo, California, Morgan Kaufmann.
34. Nagata, Y. and S. Kobayashi (1997). "Edge assembly crossover: a highpower genetic algorithm for the traveling salesman problem." Proceedings of the seventh International Conference on Genetic Algorithms: 450-457.
35. Nelder, J. and R. Mead (1965). "A simplex method for functions minimizations." Computer Journal 7 7(4): 308–313.
36. Puris, A. Y. (2009). Desarrollo de meta-heurísticas poblacionales para la solución de problemas complejos. Departamento de Inteligencia Artificial. Santa Clara, Universidad Central "Marta Abreu" de las Villas.
37. Seront, G. and H. Bersini (2000). A new ga-local search hybrid for continous optimization based on multi level single linkaga clustering. of the Genetic and Evolutionary Computation Conference 2000.
38. Sheskin, D. (2006). Handbook of parametric and nonparametric statistical procedures. CRC. London/West Palm Beach, Chapman & Hall. **1736**.
39. Solis, F. J. and R. J. West (1981). "Minimization by Random Search Techniques." Mathematical Operations Research. **6**: 19–30.
40. Storn, R. and K. Price (1997). "Diferential Evolution. A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces." Journal of Global Optimization **11**: 341-359.
41. Voss, S., S. Martello, et al., Eds. (1999). Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization. Boston, MA, Kluwer Academic Publishers.
42. Voudouris, C. and E. Tsang (1995). Guided local search. University of Essex, Department of Computer Science.

43. Wilcoxon, F. (1945). "Adjusted p-values for simultaneous inference." Biometrics
48: 1005-1013.

ANEXOS

Anexo A. Tablas de los valores obtenidos para cada una de las funciones de prueba del CEC'2005 por los modelos estudiados.

DataSet	VMO-C2	VMO-C3	VMO-C4	VMO-C6
F6	2,06E+02	4,45E+01	4,81E+01	1,00E+02
F7	6,39E+02	2,03E+02	1,83E+02	6,53E+02
F8	2,04E+01	2,03E+01	2,04E+01	2,04E+01
F9	1,36E+01	1,14E+01	1,17E+01	1,42E+01
F10	8,01E+00	7,97E+00	7,13E+00	1,00E+01
F11	1,79E+00	1,62E+00	2,14E+00	1,49E+00
F12	3,43E+02	4,80E+02	5,22E+02	3,73E+02
F13	1,12E+00	9,04E-01	1,03E+00	1,12E+00
F14	2,75E+00	2,62E+00	2,70E+00	2,69E+00
F15	2,35E+02	2,52E+02	2,72E+02	2,61E+02
F16	1,10E+02	1,04E+02	1,10E+02	1,15E+02
F17	1,14E+02	1,13E+02	1,11E+02	1,16E+02
F18	6,51E+02	6,67E+02	6,79E+02	6,75E+02
F19	5,29E+02	5,84E+02	6,84E+02	6,60E+02
F20	6,60E+02	6,49E+02	6,20E+02	6,40E+02
F21	4,76E+02	4,68E+02	4,76E+02	5,08E+02
F22	7,70E+02	7,62E+02	7,50E+02	7,68E+02
F23	5,66E+02	6,14E+02	5,98E+02	5,71E+02
F24	2,12E+02	2,36E+02	2,12E+02	2,12E+02
F25	2,12E+02	2,00E+02	2,00E+02	2,00E+02

Tabla 1. Valores obtenidos para cada una de las funciones con los modelos VMO-Simplex con intensidad 10%, profundidad 30 y la fase de BL aplicada sobre un nodo.

DataSet	VMO-C2	VMO-C3	VMO-C4	VMO-C6
F6	2,21E+02	1,71E+02	1,50E+02	3,78E+02
F7	5,76E+02	2,04E+02	2,09E+02	6,21E+02
F8	2,03E+01	2,04E+01	2,03E+01	2,04E+01
F9	1,64E+01	1,38E+01	1,26E+01	1,44E+01
F10	8,81E+00	7,49E+00	7,05E+00	8,97E+00
F11	1,96E+00	1,75E+00	1,64E+00	1,85E+00
F12	5,06E+02	5,57E+02	1,43E+02	2,22E+02
F13	1,09E+00	9,11E-01	9,94E-01	1,06E+00
F14	2,77E+00	2,75E+00	2,68E+00	2,81E+00
F15	3,11E+02	2,71E+02	2,34E+02	2,36E+02
F16	1,12E+02	1,07E+02	1,11E+02	1,10E+02
F17	1,13E+02	1,13E+02	1,14E+02	1,18E+02
F18	6,84E+02	6,96E+02	6,25E+02	5,86E+02
F19	6,53E+02	6,64E+02	6,64E+02	6,05E+02
F20	6,58E+02	6,44E+02	6,84E+02	6,20E+02
F21	4,87E+02	4,76E+02	4,84E+02	4,82E+02
F22	7,70E+02	7,51E+02	7,64E+02	7,75E+02
F23	5,54E+02	5,72E+02	5,80E+02	5,97E+02
F24	2,00E+02	2,12E+02	2,00E+02	2,00E+02
F25	2,00E+02	2,12E+02	2,00E+02	2,00E+02

Tabla 2. Valores obtenidos para cada una de las funciones con los modelos VMO-Simplex con intensidad 20%, profundidad 10 y la fase de BL aplicada sobre un nodo.

DataSet	VMO-C19	VMO-C20	VMO-C21	VMO-C23
F6	2,46E+02	1,77E+02	1,12E+02	5,61E+01
F7	4,19E+02	2,24E+02	2,23E+02	4,16E+02
F8	2,03E+01	2,04E+01	2,04E+01	2,03E+01
F9	1,43E+01	1,27E+01	1,20E+01	1,67E+01
F10	8,21E+00	7,85E+00	7,21E+00	6,89E+00
F11	1,79E+00	1,47E+00	1,70E+00	2,24E+00
F12	3,90E+02	3,05E+02	2,48E+02	2,45E+02
F13	9,46E-01	1,02E+00	9,96E-01	1,10E+00
F14	2,61E+00	2,55E+00	2,70E+00	2,79E+00
F15	2,58E+02	2,35E+02	2,59E+02	2,63E+02
F16	1,06E+02	1,07E+02	1,08E+02	1,09E+02
F17	1,13E+02	1,17E+02	1,16E+02	1,12E+02
F18	5,80E+02	7,52E+02	6,90E+02	6,24E+02
F19	6,10E+02	6,36E+02	6,33E+02	6,45E+02
F20	7,42E+02	6,68E+02	6,88E+02	5,87E+02
F21	4,90E+02	5,11E+02	5,06E+02	4,90E+02
F22	7,62E+02	7,56E+02	7,72E+02	7,73E+02
F23	5,76E+02	6,18E+02	5,87E+02	5,81E+02
F24	2,12E+02	2,12E+02	2,12E+02	2,24E+02
F25	2,12E+02	2,12E+02	2,48E+02	2,00E+02

Tabla 3. Valores obtenidos para cada una de las funciones con los modelos VMO-Simplex con intensidad 10%, profundidad 30 y la fase de BL aplicada sobre tres nodos.

DataSet	VMO-C19	VMO-C20	VMO-C21	VMO-C23
F6	4,79E+01	4,82E+01	7,14E+01	7,80E+01
F7	3,45E+02	8,93E+01	1,83E+02	3,24E+02
F8	2,03E+01	2,03E+01	2,03E+01	2,03E+01
F9	1,29E+01	1,43E+01	1,44E+01	1,31E+01
F10	7,37E+00	8,13E+00	6,47E+00	7,97E+00
F11	1,99E+00	1,69E+00	1,74E+00	1,91E+00
F12	7,21E+02	4,24E+02	2,91E+02	3,48E+02
F13	1,02E+00	9,01E-01	1,02E+00	1,10E+00
F14	2,63E+00	2,53E+00	2,64E+00	2,49E+00
F15	2,49E+02	2,60E+02	2,37E+02	2,59E+02
F16	1,11E+02	1,10E+02	1,07E+02	1,06E+02
F17	1,11E+02	1,14E+02	1,09E+02	1,14E+02
F18	6,63E+02	6,44E+02	6,89E+02	6,04E+02
F19	6,04E+02	6,49E+02	6,74E+02	5,83E+02
F20	6,38E+02	6,28E+02	5,84E+02	6,36E+02
F21	5,00E+02	4,95E+02	4,80E+02	5,03E+02
F22	7,33E+02	7,55E+02	7,10E+02	7,73E+02
F23	6,74E+02	6,23E+02	5,79E+02	6,00E+02
F24	2,00E+02	2,00E+02	2,00E+02	2,12E+02
F25	2,24E+02	2,12E+02	2,12E+02	2,00E+02

Tabla 4. Valores obtenidos para cada una de las funciones con los modelos VMO-Simplex con intensidad 20%, profundidad 10 y la fase de BL aplicada sobre tres nodos.

DataSet	VMO-C2	VMO-C3	VMO-C4	VMO-C6
F6	1,04E+02	4,24E+01	8,28E+01	9,96E+01
F7	1,27E+03	1,27E+03	1,27E+03	1,27E+03
F8	2,03E+01	2,03E+01	2,03E+01	2,03E+01
F9	9,67E+00	1,17E+01	1,16E+01	1,27E+01
F10	7,17E+00	8,17E+00	8,05E+00	9,08E+00
F11	1,69E+00	1,60E+00	1,63E+00	1,44E+00
F12	6,09E+02	6,06E+02	2,84E+02	4,14E+02
F13	9,02E-01	9,69E-01	1,01E+00	9,03E-01
F14	2,64E+00	2,54E+00	2,60E+00	2,57E+00
F15	2,11E+02	2,71E+02	2,50E+02	2,83E+02
F16	1,12E+02	1,08E+02	1,14E+02	1,08E+02
F17	1,20E+02	1,12E+02	1,11E+02	1,12E+02
F18	7,13E+02	5,80E+02	5,86E+02	6,65E+02
F19	6,89E+02	6,85E+02	6,64E+02	6,73E+02
F20	7,05E+02	6,72E+02	7,38E+02	6,52E+02
F21	4,84E+02	5,00E+02	4,92E+02	4,60E+02
F22	7,72E+02	7,54E+02	7,61E+02	7,52E+02
F23	5,82E+02	5,81E+02	6,02E+02	5,66E+02
F24	2,12E+02	2,12E+02	2,12E+02	2,12E+02
F25	2,12E+02	2,00E+02	2,00E+02	2,00E+02

Tabla 5. Valores obtenidos para cada una de las funciones con los modelos VMO-Solis-West con intensidad 10%, profundidad 30, y la fase de BL aplicada sobre un nodo.

DataSet	VMO-C2	VMO-C3	VMO-C4	VMO-C6
F6	1,30E+02	6,90E+01	8,84E+01	1,08E+02
F7	1,27E+03	1,27E+03	1,27E+03	1,27E+03
F8	2,03E+01	2,04E+01	2,03E+01	2,03E+01
F9	1,27E+01	1,24E+01	1,21E+01	1,18E+01
F10	9,16E+00	7,89E+00	6,85E+00	8,01E+00
F11	2,16E+00	1,78E+00	1,76E+00	1,80E+00
F12	7,30E+02	5,69E+02	4,45E+02	3,16E+02
F13	9,94E-01	8,84E-01	8,50E-01	8,74E-01
F14	2,71E+00	2,62E+00	2,59E+00	2,72E+00
F15	3,23E+02	2,54E+02	2,61E+02	2,23E+02
F16	1,13E+02	1,08E+02	1,10E+02	1,11E+02
F17	1,16E+02	1,13E+02	1,13E+02	1,19E+02
F18	7,69E+02	6,88E+02	6,20E+02	6,93E+02
F19	6,97E+02	4,40E+02	6,40E+02	6,64E+02
F20	6,44E+02	6,11E+02	7,04E+02	6,72E+02
F21	4,76E+02	4,92E+02	4,76E+02	4,83E+02
F22	7,48E+02	7,42E+02	7,62E+02	7,67E+02
F23	6,03E+02	6,09E+02	5,74E+02	6,24E+02
F24	2,00E+02	2,00E+02	2,00E+02	2,12E+02
F25	2,00E+02	2,00E+02	2,12E+02	2,00E+02

Tabla 6. Valores obtenidos para cada una de las funciones con los modelos VMO-Solis-West con intensidad 20%, profundidad 10 y la fase de BL aplicada sobre un nodo.

DataSet	VMO-C19	VMO-C20	VMO-C21	VMO-C23
F6	4,31E+01	8,71E+01	1,54E+02	8,15E+01
F7	1,27E+03	1,27E+03	1,27E+03	1,27E+03
F8	2,03E+01	2,04E+01	2,03E+01	2,03E+01
F9	1,18E+01	1,34E+01	1,22E+01	1,06E+01
F10	8,13E+00	7,25E+00	7,77E+00	8,52E+00
F11	1,43E+00	1,88E+00	1,94E+00	1,61E+00
F12	3,86E+02	3,90E+02	4,68E+02	3,85E+02
F13	9,94E-01	1,03E+00	8,51E-01	9,13E-01
F14	2,44E+00	2,57E+00	2,46E+00	2,70E+00
F15	2,99E+02	2,84E+02	2,54E+02	2,69E+02
F16	1,11E+02	1,09E+02	1,09E+02	1,06E+02
F17	1,10E+02	1,09E+02	1,13E+02	1,09E+02
F18	6,46E+02	7,13E+02	6,02E+02	6,94E+02
F19	7,25E+02	6,58E+02	6,64E+02	6,44E+02
F20	6,99E+02	6,75E+02	6,60E+02	6,49E+02
F21	4,92E+02	4,84E+02	4,92E+02	4,92E+02
F22	7,65E+02	7,41E+02	7,56E+02	7,50E+02
F23	5,82E+02	5,74E+02	5,59E+02	6,03E+02
F24	2,12E+02	2,24E+02	2,00E+02	2,36E+02
F25	2,00E+02	2,24E+02	2,12E+02	2,12E+02

Tabla 7. Valores obtenidos para cada una de las funciones con los modelos VMO-Solis-West con intensidad 10%, profundidad 30, y la fase de BL aplicada sobre tres nodos.

DataSet	VMO-C19	VMO-C20	VMO-C21	VMO-C23
F6	4,79E+01	4,82E+01	7,14E+01	7,80E+01
F7	3,45E+02	8,93E+01	1,83E+02	3,24E+02
F8	2,03E+01	2,03E+01	2,03E+01	2,03E+01
F9	1,29E+01	1,43E+01	1,44E+01	1,31E+01
F10	7,37E+00	8,13E+00	6,47E+00	7,97E+00
F11	1,99E+00	1,69E+00	1,74E+00	1,91E+00
F12	7,21E+02	4,24E+02	2,91E+02	3,48E+02
F13	1,02E+00	9,01E-01	1,02E+00	1,10E+00
F14	2,63E+00	2,53E+00	2,64E+00	2,49E+00
F15	2,49E+02	2,60E+02	2,37E+02	2,59E+02
F16	1,11E+02	1,10E+02	1,07E+02	1,06E+02
F17	1,11E+02	1,14E+02	1,09E+02	1,14E+02
F18	6,63E+02	6,44E+02	6,89E+02	6,04E+02
F19	6,04E+02	6,49E+02	6,74E+02	5,83E+02
F20	6,38E+02	6,28E+02	5,84E+02	6,36E+02
F21	5,00E+02	4,95E+02	4,80E+02	5,03E+02
F22	7,33E+02	7,55E+02	7,10E+02	7,73E+02
F23	6,74E+02	6,23E+02	5,79E+02	6,00E+02
F24	2,00E+02	2,00E+02	2,00E+02	2,12E+02
F25	2,24E+02	2,12E+02	2,12E+02	2,00E+02

Tabla 8. Valores obtenidos para cada una de las funciones con los modelos VMO-Solis-West con intensidad 20%, profundidad 10, y la fase de BL aplicada sobre tres nodos.

DataSet	VMO-Puro
F6	6,44E+01
F7	1,27E+03
F8	2,03E+01
F9	1,15E+01
F10	6,89E+00
F11	1,57E+00
F12	2,57E+02
F13	9,34E-01
F14	2,65E+00
F15	2,34E+02
F16	1,08E+02
F17	1,15E+02
F18	6,49E+02
F19	6,29E+02
F20	5,40E+02
F21	5,15E+02
F22	7,75E+02
F23	5,73E+02
F24	2,12E+02
F25	2,12E+02

Tabla 9. Valores obtenidos para cada una de las funciones para el modelo VMO-Puro.

Anexo B. Breve Descripción de las funciones utilizadas como prueba

➤ Siete funciones básicas

- ✓ F_6 : Función de Rosenbrock desplazada, $x \in [-100,100]^D$
- ✓ F_7 : Función de Griewank desplazada y rotada sin fronteras, $x \in [0,600]^D$
- ✓ F_8 : Función de Ackley desplazada y rotada con óptimo local en las fronteras, $x \in [-32,32]^D$
- ✓ F_9 : Función de Rastrigin desplazada, $x \in [-5,5]^D$
- ✓ F_{10} : Función de Rastrigin desplazada y rotada, $x \in [-5,5]^D$
- ✓ F_{11} : Función de Weierstrass desplazada y rotada, $x \in [-0.5,0.5]^D$
- ✓ F_{12} : Problema 2.13 de Schwefel, $x \in [-\pi,\pi]^D$

➤ Dos funciones expandidas

- ✓ F_{13} : Función de Griewank expandida y extendida más la función de Rosenbrock $x \in [-3,1]^D$
- ✓ F_{14} : Función de Scaffer desplazada, rotada y expandida $x \in [-100,100]^D$

➤ Once funciones híbridas

- ✓ F_{15}, \dots, F_{25} : $x \in [-5,5]^D$