



UNIVERSIDAD CENTRAL "MARTA ABREU" DE LAS VILLAS
VERITATE SOLA NOBIS IMPONETUR VIRILISTOGA. 1948
Facultad de Matemática, Física y Computación,
Centro de Estudios de Informática.



Trabajo de Diploma

***Cruzamiento genético con codificación real en la
exploración desarrollada por VMO.***

Autora: Adis Perla Mariño Rivero.

Tutores: Dr. Amilkar Yudier Puris Cáceres.

Lic. Carlos Alberto Martínez Pérez.

Santa Clara

2011

"Año 53 de la Revolución."



Universidad Central “Marta Abreu” de Las Villas

Facultad de Matemática, Física y Computación,
Centro de Estudios de Informática.



TRABAJO DE DIPLOMA

***Cruzamiento genético con codificación real en la
exploración desarrollada por VMO.***

Autora: Adis Perla Mariño Rivero.

Tutores: Dr. Amilkar Yudier Puris Cáceres.

Lic. Carlos Alberto Martínez Pérez.

Santa Clara,

2011

“Año 53 de la Revolución.”



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

Firma del Autor

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

Firma del Autor

Firma del Jefe de Departamento
donde se defiende el trabajo

Firma del Responsable de
Información Científico-Técnica

PENSAMIENTO



"Lo esencial es invisible a los ojos"

Antoine de Saint-Exupéry

DEDICATORIA

Quisiera dedicar este trabajo a las personas que siempre confiaron en mí, me apoyaron y alentaron en los momentos difíciles, para no desfallecer, y continuar el camino.

- ✓ *A mis padres: Arelis y Guillermo, mis resultados, mi graduación y este trabajo de diploma es por y para ustedes.*
- ✓ *A mi hermano.*
- ✓ *A mi abuela.*
- ✓ *A mi novio, Yandi, por permanecer a mi lado en los buenos y malos momentos, el valor de su ayuda es inmenso, gracias mi amor.*

Mity, gracias, sin ti no soy nadie.

AGRADECIMIENTOS

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

- ✓ *A mis padres por toda su ayuda, confianza y comprensión, a mi hermano y familia por estar conmigo en todo.*
- ✓ *A mi tutor Amilkar por apoyarme en todo momento que sin su ayuda no hubiera sido posible la realización de este trabajo.*
- ✓ *Al Lic. Carlos Alberto Martínez Pérez por las ayudas oportunas y por estar siempre cuando lo necesitaba.*
- ✓ *A mis primas Marelis y Dayana, las cuales quiero mucho y le agradezco profundamente todo lo que me han apoyado y ayudado.*
- ✓ *A mis amigas y amigos en especial a: Lisset F., Brenda, Dailen y Suny por estar a mi lado.*

A todas las personas que de una forma u otra aportaron su grano de arena en la elaboración de este trabajo.

TAREA TÉCNICA

Escriba aquí las tareas que debe ejecutar hasta la confección del informe final (Obligatoria)

Esta página debe estar firmada por el estudiante y por el tutor, expresión del compromiso entre ambos.

Firma del Autor

Firma del Tutor

RESUMEN

Los problemas de optimización reciben una especial atención en la actualidad debido a la alta complejidad que dichos han alcanzado. Los Algoritmos Genéticos representan modelos muy estudiados en la rama de la optimización, ya que permiten, dado un dominio de exploración, realizar una amplia búsqueda, permitiendo no quedar atrapados en óptimos locales. Un gran peso de este comportamiento lo tienen los operadores de cruzamiento utilizados por estos. La Optimización en Mallas Variables surge recientemente en nuestro grupo de investigación, la cual ha sido competitiva en dominios continuos y ha sido poco explorado la búsqueda de nuevas alternativas para mejorar su comportamiento. El objetivo de esta investigación es desarrollar un modelo que combine las ventajas que nos pueden proporcionar los operadores de cruce genéticos y la esta metaheurística poblacional incorporando estos operadores tanto de manera simple como la hibridación de ellos según sus características a la etapa de búsqueda local y global desarrollada por el modelo, con la meta de presentar un modelo más robusto que obtenga soluciones mejores que el original. Los resultados de esta investigación son la obtención de distintas variantes de incorporación de los operadores, no mejorando el comportamiento del original para la resolución de problemas continuos.

.

TABLA DE CONTENIDOS

PENSAMIENTO	i
DEDICATORIA	ii
AGRADECIMIENTOS	iii
TAREA TÉCNICA.....	iv
RESUMEN	v
INTRODUCCIÓN	1
CAPÍTULO 1. Marco teórico.....	7
1.1 Optimización Basada en Mallas Variables.	8
1.1.1 Descripción general de la metaheurística.	8
1.2 Algoritmos Genéticos de Codificación Real.....	17
CAPÍTULO 2. Incorporación de operadores genéticos de cruzamiento con codificación real a VMO.....	21
2.1 Operadores Genéticos de Cruzamiento con codificación real empleados.	21
2.1.1 Operadores de Cruce Basados en Entornos Probabilísticos.	21
2.1.2 Operadores de Cruce Basados en Entornos Deterministas.	25
2.1.3 Principales características y diferencias entre ambos operadores.	26
2.2 Incorporación de los operadores genéticos de cruzamiento con codificación real a la etapa de exploración de VMO.	26

2.3	Descripción de los test estadísticos y funciones de prueba a utilizar.....	29
CAPÍTULO 3. Análisis Experimental de los resultados.....		33
3.1	Técnicas Estadísticas para validar los resultados obtenidos.	33
3.2	Marco Experimental.....	34
3.3	Análisis de los resultados para las distintas variantes de diseño.....	35
3.3.1	Etapa de búsqueda local.....	35
3.3.1.1	Comparaciones de las distintas variantes de aplicar los operadores genéticos de cruzamiento.	36
3.3.1.2	Comparaciones de las distintas variantes de hibridación de los operadores genéticos de cruzamientos.	38
3.3.1.3	Análisis para seleccionar la mejor variante en la búsqueda local.....	39
3.3.2.1	Etapa de búsqueda global.	41
3.3.2.2	Comparaciones de las distintas variantes de aplicar los operadores genéticos de cruzamiento.....	42
3.3.2.3	Comparaciones de las distintas variantes de hibridación de los operadores genéticos de cruzamiento.....	43
3.3.2.4	Análisis para seleccionar la mejor variante en la búsqueda global.....	44
3.4	Comparaciones con el modelo VMO-Puro.	46
Conclusiones.....		49
Recomendaciones		50
REFERENCIAS BIBLIOGRÁFICAS		51
ANEXOS		53
Anexo A. Tablas de los valores obtenidos para cada una de las funciones de prueba por los modelos estudiados.		53

INTRODUCCIÓN

La optimización es, actualmente, un área que recibe mucho interés debido a que se presenta en multitud de problemas de ciencia e ingeniería que, por su propia naturaleza, no pueden ser resueltos mediante un procedimiento de búsqueda exhaustivo (que obtenga la mejor solución tras una búsqueda por todo el dominio de posibles valores). En Matemática y Ciencia de la Computación este concepto se refiere a escoger el "mejor elemento" de un conjunto disponible de alternativas. En el caso más simple, esto significa resolver problemas en los que se busca minimizar (o maximizar) una función real escogiendo sistemáticamente valores de variables reales de un conjunto permitido.

Es pertinente identificar, en forma previa al análisis del concepto de metaheurística, que significa una heurística. La etimología del término heurística proviene de la palabra griega *heuriskein* que se traduce como encontrar. Más allá de las múltiples interpretaciones y de la vastísima bibliografía al respecto, se propone aquí la idea de método heurístico sugerida en [1]:

Un método heurístico es un procedimiento para resolver un problema de optimización bien definido mediante una aproximación intuitiva, en la que la estructura del problema se utiliza de forma inteligente para obtener una buena solución.

En los últimos años ha aparecido un conjunto de métodos bajo el nombre de metaheurísticos [2] con el propósito de obtener mejores resultados que los alcanzados por los heurísticos tradicionales. En algunos textos se puede encontrar la expresión “heurísticos modernos” refiriéndose a los metaheurísticos. Los profesores Osman y Kelly [3] introducen la siguiente definición de procedimientos metaheurísticos:

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

Estos procedimientos se sitúan conceptualmente “por encima” de los heurísticos en el sentido de que son los que guían el diseño de éstos. Así, al enfrentarnos a un problema de optimización, es posible escoger cualquiera de estos métodos para diseñar un algoritmo específico que lo resuelva aproximadamente.

La búsqueda de la solución del problema continuo usualmente se plantea como localizar el extremo (mínimo o máximo) de una función objetivo $F: D \rightarrow \mathbb{R}$; es decir, encontrar un punto $x_0 \in D$ tal que:

$F(x_0) \leq F(x)$ para todo $x \in D$; para el caso de minimizar la función F , y

$F(x_0) \geq F(x)$ para todo $x \in D$; para el caso de maximizar la función F .

El objetivo es asignar valores del dominio, de los permitidos por las restricciones, tal que la función objetivo sea optimizada. Los componentes del problema de optimización son:

- Objetivo – función objetivo (fitness function).
- Conjunto de parámetros (desconocidos) los cuales afectan el valor de la función objetivo.
- Conjunto de restricciones que restringen los valores que se pueden asignar.

Se han desarrollado muchos métodos de optimización global, un aspecto importante de todos estos métodos es su costo computacional, el cual crece polinomialmente como una función del tamaño del problema [4]. La solución de estos problemas por métodos heurísticos son una alternativa para enfrentarlos [5]. Las técnicas de computación evolutivas pertenecen a esta categoría de métodos. Ellas trabajan sobre un conjunto de soluciones potenciales, llamada población, y realizan la exploración del espacio de búsqueda a través de la cooperación y competencia entre las soluciones potenciales;

ejemplos de estas técnicas son los Algoritmos Genéticos(AG) [6], Sistemas de Partículas [7], la Evolución Diferencial [8], Algoritmos Basados en Estimación de Distribuciones[9] etc. Los algoritmos evolutivos, han demostrado su efectividad al escapar de los óptimos locales y descubrir el óptimo global incluso en paisajes adaptativos muy escabrosos y complejos. Ellas también se han denominado modelos computacionales bioinspirados, pues fueron desarrollados a partir de modelos existentes en la naturaleza. Cuando el espacio de búsqueda es demasiado grande para realizar una búsqueda exhaustiva, las técnicas de búsqueda basadas en población pueden ser una buena alternativa; sin embargo, ellas no garantizan encontrar la solución óptima.

Los AG surgen como herramientas para la solución de complejos problemas de búsqueda y optimización, producto del análisis de los sistemas adaptativos en la naturaleza, y como resultado de abstraer la esencia de su funcionamiento. El término AG se usa por el hecho de que estos simulan los procesos de la evolución a través del uso de operadores genéticos que operan sobre una población de individuos que “evoluciona” de una generación a otra. Un AG puede ser visto como una estructura de control que organiza o dirige un conjunto de transformaciones que se ejecutan iterativamente sobre una población de individuos que representan soluciones para el problema a resolver; estas transformaciones se realizan mediante los llamados operadores genéticos y considerando una función de calidad o grado de aptitud de un individuo, la cual es una medida que permite comparar las soluciones para determinar cuál es mejor [6,10].

Entre los operadores genéticos se destacan: el de selección, el cual escoge los individuos de la población que serán utilizados para la reproducción y su meta es dar más oportunidades de escoger a los miembros más aptos de ella; el operador de cruce, que es uno de los más importantes, ya que es el principal operador que introduce diversidad en la población, permitiendo compartir información entre las soluciones y se basa en combinar características de los vectores de varios padres (usualmente dos) para producir nuevas soluciones y el operador de mutación, el cual es capaz de introducir de forma aleatoria diversidad en la población a partir de una única solución y modifica uno o varios genes de una solución , dicha modificación se utiliza para aumentar la diversidad de las poblaciones.

El cruzamiento es el elemento clave que distingue a los algoritmos genéticos de los otros métodos, sin este, cada solución individual va por su cuenta, explorando el espacio de búsqueda en sus inmediaciones sin referencia de lo que el resto de individuos puedan haber descubierto, sin embargo, con el cruzamiento en juego, hay una transferencia de información entre los candidatos prósperos, es decir, los individuos pueden beneficiarse de lo que otros han aprendido, y los esquemas pueden mezclarse y combinarse, con el potencial de producir una descendencia que tenga las virtudes de sus dos padres y ninguna de sus debilidades. Basado en la experiencia del operador de cruce en los AG y el surgimiento en nuestro grupo de investigación de una metaheurística poblacional (Optimización basada en Mallas Variable, VMO) que ha sido competitiva en dominios continuos y poco explorada la búsqueda de nuevas alternativas para mejorar su comportamiento, estos operadores serán incorporados al modelo con el objetivo de mejorar la calidad de las soluciones brindadas por este.

La metaheurística VMO cae en la categoría de algoritmos evolutivos. Un conjunto de nodos que representan soluciones potenciales de un problema de optimización forman una malla que dinámicamente crece y se desplaza por el espacio de búsqueda, para ello en cada ciclo se generan nodos intermedios entre los nodos de la malla y aquellos nodos que son extremos locales (nodos obtenidos de mejor calidad en una vecindad determinada) y el extremo global (nodo obtenido de mejor calidad en todo el proceso desarrollado); así como a partir de los nodos más externos de la malla. Los mejores nodos de la malla resultante son usados como malla inicial para el ciclo siguiente.

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

Estudiar el comportamiento de la metaheurística VMO al incorporarle operadores genéticos de cruzamiento con codificación real al proceso de búsqueda desarrollado por el modelo a fines de mejorar la calidad de las soluciones con respecto al modelo original.

En concreto, el objetivo propuesto se subdivide en los siguientes **objetivos específicos**:

- ❖ Modelar e implementar los operadores de cruzamiento con codificación real más utilizados en la literatura internacional, que son los encargados de la explotación realizada por los AG, a la etapa de exploración local y global de VMO

- ❖ Diseñar nuevas estrategias híbridas de búsqueda mezclando los operadores utilizados según sus características.
- ❖ Comparar los resultados obtenidos por las distintas estrategias frente a las soluciones del modelo original.

Durante el análisis del problema nos surgieron las **preguntas de investigación** siguientes:

- ❖ ¿Cómo se podrán aplicar los operadores genéticos de cruce más utilizados en codificación real en la etapa de exploración del modelo VMO?
- ❖ ¿Cuáles de los operadores utilizados brinda facilidades para conformar nuevas formas de exploración híbridas?
- ❖ ¿Se mejora la calidad de las soluciones obtenidas por el modelo original VMO cuando se incorporan los operadores genéticos de cruce seleccionados?

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

La hibridación del modelo VMO con distintos operadores de cruzamiento con codificación real al proceso de búsqueda desarrollado por la metaheurística trae consigo un mejor comportamiento en cuanto a calidad de las soluciones con respecto al modelo original.

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

El presente trabajo de diploma se estructura en tres Capítulos.

En el Capítulo 1 se abordará en detalles la metaheurística VMO y en forma general los operadores de cruzamiento en los Algoritmos Genéticos de Codificación Real.

En el Capítulo 2 se expondrán detalladamente los Operadores de Cruce Basados en Entornos tanto probabilísticos como deterministas, y como fueron incorporados estos tanto en su manera simple como la hibridación entre ellos según sus características a la etapa de búsqueda desarrollada por el modelo.

Finalmente, en el Capítulo 3 se presentarán los resultados experimentales alcanzados.

Tras el análisis de estos tres capítulos aparecerán las Conclusiones y las Recomendaciones, así como las propuestas de trabajo futuro en el tema.

CAPÍTULO 1. Marco teórico.

Muchos problemas de ingeniería pueden formularse como problemas de optimización sobre una función objetivo definido sobre un dominio D de soluciones alcanzables. Resolverlos supone encontrar dentro de un dominio definido la solución que permita minimizar dicha función objetivo (óptimo global). Dicha problemática se presenta en numerosos problemas de investigación e ingeniería, cuando se necesita obtener, de un conjunto de soluciones, la mejor a un problema dado, y se posee un criterio capaz de medir la calidad de cada solución.

Matemáticamente, se puede expresar que, dada una función $f: D \rightarrow R$, el objetivo del óptimo global es encontrar x^* para el que:

$$x^* = \operatorname{argmin}_{x \in D} f(x)$$

En esta tesis nos centraremos en la problemática de optimización continua (es decir, en la que $D \in \mathbb{R}^N$). Para dicho dominio, el problema de encontrar el óptimo global se puede redefinir en, dado un $\varepsilon \in \mathbb{R}$, $\varepsilon > 0$, encontrar un valor x' tal que:

$$f(x^*) - f(x') \leq \varepsilon$$

En la búsqueda relacionada a estos problemas existen dos factores fundamentales [11]:

- 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.

Obtener métodos que utilicen una continua sinergia entre ambos factores es un elemento a tener en cuenta para conseguir soluciones más precisas. Existen varias técnicas heurísticas para resolver problemas de este tipo, entre las cuales podemos destacar las metaheurísticas poblacionales las cuales suelen apostar por obtener una mayor exploración en la búsqueda.

1.1 Optimización Basada en Mallas Variables.

La Optimización Basada en Mallas Variables (*Variable Mesh Optimization*; VMO) es una metaheurística poblacional con características evolutivas donde un conjunto de P nodos $(n_1, n_2, \dots, n_P)$ 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), que inicialmente se distribuye usando una distribución uniforme. 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, eliminando todos los nodos cercanos usando la función de calidad, con el objetivo de mantener el tamaño de la población y fomentar la diversidad de la malla, donde los mejores representantes de cada zona explotada del espacio de búsqueda son seleccionados como malla inicial para la iteración siguiente. La formulación general de la metaheurística abarca tanto los problemas de optimización continuos como los discretos, en nuestro caso realizaremos énfasis en los problemas continuos.

1.1.1 Descripción general de la metaheurística.

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 $FO(v_1^i, v_2^i, \dots, v_M^i)$. 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. Cada nodo se codifica por un vector de M números flotantes, $n_i = (v_1^i, v_2^i, \dots, v_M^i) = v_j^i, j=1, \dots, M$ que representan la solución al problema de optimización.

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 que se explican en la Tabla 1.1.

Tabla 1.1 Parámetros del algoritmo VMO.

Parámetro	Descripción
P	Cantidad máxima de nodos de la malla en cada ciclo
T	Cantidad de nuevos nodos que requiere el proceso de expansión
k	Tamaño de la vecindad
C	Condición de parada

La malla se mueve mediante un proceso de expansión hacia otras regiones del espacio de búsqueda, para esta acción, nuevos nodos son generados, usando la población actual en cada iteración, según los siguientes pasos:

Paso1: Generación aleatoria de la malla inicial: la población inicial para la primera iteración del algoritmo está compuesta de P nodos que se generan aleatoriamente con una distribución uniforme.

Paso2: Generación de nodos en dirección a los extremos locales: el primer tipo de exploración que se realiza en VMO, se lleva a cabo en las vecindades de cada uno de los nodos (n_i) de la malla inicial. Este paso se encarga de realizar una intensificación de la búsqueda en la vecindad de cada nodo para calcular los vecinos más cercanos (n_i^*) de cada nodo de la malla. Solo cuando n_i^* es mejor que n_i , se genera un nuevo nodo entre n_i y n_i^* . Por esta razón, en este paso Z nodos son creados, donde $Z \leq P - 1$.

Para descubrir el vecino del i -ésimo nodo, se utiliza como función de distancia la euclidiana, definida por:

$$D_{euclidiana}(n_1, n_2) = \sqrt{\sum_{j=1}^M (v_j^1 - v_j^2)^2} \quad (1.1)$$

El nuevo nodo (n_z) es calculado usando la función F definida por la siguiente expresión:

$$n_z = F(n_i, n_i^*, Pr(n_i, n_i^*)) \quad (1.2)$$

donde Pr es el factor de cercanía y representa la relación entre los valores que alcanza la función de calidad en el nodo actual y su extremo correspondiente.

Mientras mayor sea la diferencia entre los valores de FO en los nodos involucrados, mayor será la cercanía o semejanza de n_i a n_i^* , esto lo garantiza el factor Pr . Este toma valores en el rango $[0.1]$, superior cuando n_i tiene mejor calidad y se calcula usando la ecuación siguiente:

$$Pr = \frac{1}{1 + |FO(n_i) - FO(n_i^*)|} \quad (1.3)$$

Esta forma de calcular Pr 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 puede describirse de diferentes maneras. Para este estudio, las componentes v_j^z del nuevo nodo n_z son calculadas por la ecuación:

$$v_j^z = \begin{cases} \overline{m}_j, & \text{si } |\overline{m}_j - v_j^{i*}| > \xi_j \text{ y } U[0, 1] \leq Pr(n_i, n_i^*) \\ v_j^{i*} + U[-\xi_j, \xi_j], & \text{si } |\overline{m}_j - v_j^{i*}| \leq \xi_j \\ U[v_j^{i*}, \overline{m}_j], & \text{en otro caso} \end{cases} \quad (1.4)$$

donde $\overline{m}_j = average(v_j^i, v_j^{i*})$, $U[x, y]$ denota un valor aleatorio (uniforme) en el intervalo $[x, y]$ y ξ_j es el mínimo permitido de distancia para cada componente. Este valor va decreciendo durante la corrida del algoritmo y se calcula según la ecuación:

$$\xi_j = \begin{cases} \frac{range(a_j, b_j)}{4}, & \text{si } c < 0.15\%C \\ \frac{range(a_j, b_j)}{8}, & \text{si } 0.15\%C \leq c < 0.3\%C \\ \frac{range(a_j, b_j)}{16}, & \text{si } 0.3\%C \leq c < 0.6\%C \\ \frac{range(a_j, b_j)}{50}, & \text{si } 0.6\%C \leq c < 0.8\%C \\ \frac{range(a_j, b_j)}{100}, & \text{si } c \geq 80\%C \end{cases} \quad (1.5)$$

donde C denota el valor máximo de evaluaciones de la función objetivo y c la evaluación actual. En la definición $range(a_j, b_j)$ denota la amplitud del dominio (a_j, b_j) para cada componente. En dependencia del por ciento del total que represente la evaluación actual, se definen valores de distancia que representan partes del intervalo permitido.

De forma general, la función F se comporta de manera que en el primer caso se obtiene para la j -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.

En la Figura 1.1 se muestra un ejemplo de este paso, con $k=4$.

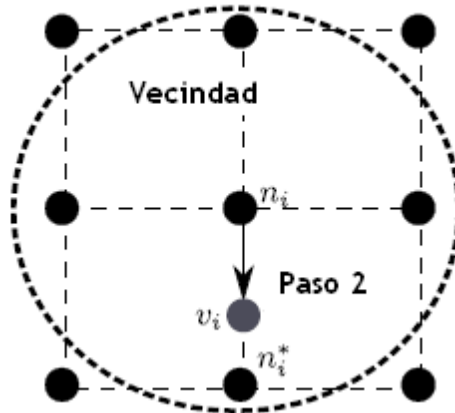


Figura 1.1 Ejemplo del Paso 2.

Paso3: Generación de nodos en dirección al extremo global: este paso se utiliza para acelerar la convergencia del algoritmo. Para esto se explora en la dirección del nodo que tiene mejor evaluación en la población actual llamado extremo global de la malla (n_g). X nuevos nodos se generan ($X=P-1$), uno para cada n_i hacia n_g usando la ecuación:

$$n_x = G(n_i, n_g, \text{Pr}(n_i, n_g)) \quad (1.6)$$

En la expresión 1.7 nosotros presentamos la función G , donde v_j^x representa el valor computacional para cada componente del nuevo nodo n_x representa de la siguiente expresión:

$$n_j^x = \begin{cases} \text{average}(v_j^i, v_j^g) & \text{si } U[0,1] \leq \text{Pr}(n_i, n_g) \\ U[\text{average}(v_j^i, v_j^g), v_j^g], & \text{en otro caso} \end{cases} \quad (1.7)$$

Si hay una gran diferencia (en valor de la función) entre el n_i y n_g , habrá una probabilidad alta para la j -ésima componente de tomar valores próximos a v_j^g . En otro caso, se obtiene el valor medio.

En la Figura 1.2 se muestra este paso.

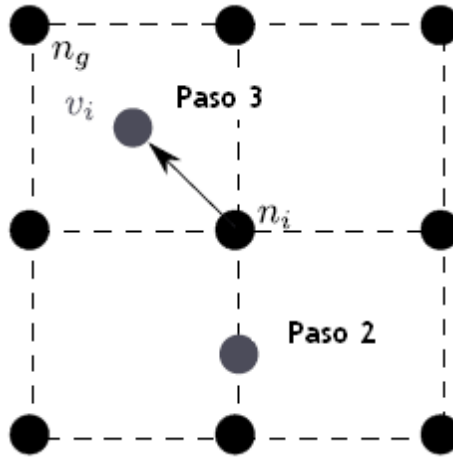


Figura 1.2 Ejemplo del Paso 3.

Paso4: Generación de nodos a partir de los más externos de la malla: En este paso se crean Y nodos los cuales completan la cantidad total de nodos que debe tener la malla, a partir de los nodos fronteras. Este paso solo se corre si el número de nodos creados en los pasos anteriores ($Z + X$) es menor que T , y $Y = T - (Z + X)$ son creados en este paso. Si $Y > P$, solo P nuevos nodos son creados, uno para cada nodo de la malla. En este paso, se

consideran nodos fronteras (n_t). La frontera está compuesta por los nodos que están más cercanos (conocidos como frontera interior o los nodos interiores, n_u) y los más alejados (conocidos como frontera exterior o los nodos externos, n_s) para el punto que representa el centro espacial de búsqueda. Para detectar estos nodos es usada la distancia euclidiana. Los nodos de la distancia más alta componen el grupo n_s y los que tienen una distancia más pequeña componen el grupo n_u . Partiendo de estos grupos, nuevos nodos son creados (uno para cada nodo frontera) usando la función H (ver expresión 1.8).

$$n_h = H(n_t, w) \quad (1.8)$$

Para nuestro estudio, la función H es definida por medio de las ecuaciones (1.9) y (1.10) donde las componentes v_j^h de cada nuevo nodo n_h son calculados dependiendo del tipo de frontera:

En la ecuación (1.9) $[Y/2]$ nodos son creados, usando el grupo n_s (nodos externos):

$$v_j^h = \begin{cases} v_j^s + w_j, & \text{si } v_j^s > 0 \\ v_j^s - w_j, & \text{si } v_j^s < 0 \end{cases} \quad (1.9)$$

En la ecuación (1.10) $Y - [Y/2]$ nodos son creados, usando el grupo n_u (nodos internos):

$$v_j^h = \begin{cases} |v_j^u + w_j|, & \text{si } v_j^u > 0 \\ |v_j^u - w_j|, & \text{si } v_j^u \leq 0 \end{cases} \quad (1.10)$$

donde w_j representa el desplazamiento para cada componente j y es calculado de una manera decreciente según la ecuación:

$$w_j = (w_j^0 - w_j^1) \cdot \frac{C - c}{C} + w_j^1 \quad (1.11)$$

La variable w_j^0 representa el desplazamiento inicial y w_j^1 el valor final. Además, los valores C y c son usados (ver descripción de la expresión 1.5). En orden se obtienen los desplazamientos $w_j^0 > w_j^1$ y estos valores son calculados como: $w_j^0 = \text{range}(a_j, b_j)/10$ y $w_j^1 = \text{range}(a_j, b_j)/100$.

En la Figura 1.3 se muestra un ejemplo de este paso para $Y=4$.

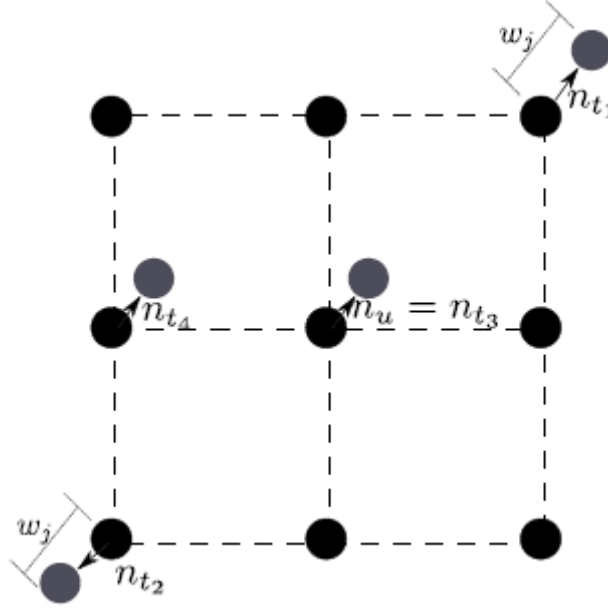


Figura 1.3 Ejemplo del Paso 4.

Paso 5: Todos los nodos de la malla son ordenados dependiendo de su función de calidad (Ascendentemente).

Paso 6: La diferencia entre cada nodo y su sucesor es calculado para cada dimensión. El nodo sucesor con una diferencia mayor a su correspondiente valor ζ_j es eliminado. El ζ_j es calculado por la expresión 1.5. Esta estrategia es llamada limpieza adaptativa, y finalmente nos quedamos con B nodos.

Paso 7: Los nodos con una mejor función de calidad son seleccionados para formar la población de la próxima iteración. Si $B < P$ entonces la malla es completada con nuevos nodos aleatorios.

Este mecanismo considera dos elementos importantes: la calidad del nodo y su lugar en el espacio de solución. Los nodos con mejor adaptabilidad tienen una probabilidad superior de tomar parte en la siguiente población. La limpieza adaptativa permite el método llevar a cabo más exploraciones generales y en un futuro reducir esta frecuencia para enfocar la atención en un área más pequeña del espacio de búsqueda. Este elemento aumenta el nivel de explotación del método y lo hace más firme.

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 usando los parámetros descritos en la Tabla 1.1:

(Definir los parámetros de entrada fijando: P, T, k y C).

Paso 1: P nodos son generados aleatoriamente para componer la población inicial y todos son evaluados para seleccionar el mejor n_g .

Paso 2: Para cada nodo de n_i de la población actual (se crean Z nuevos nodos, $Z < P$):

2.1. Encontrar los k nodos más cercanos usando la distancia euclidiana descrita por la expresión 1.1.

2.2. Seleccionar el mejor nodo n_i^* entre los vecinos.

2.3. Si n_i^* es mejor que n_i ,entonces

- calcular el factor de cercanía Pr entre n_i y n_i^* usando la expresión 1.3.

- genero un nuevo nodo por la expresión 1.2 usando n_i , n_i^* y Pr .

Paso 3: Para cada nodo n_i (X nuevos nodos son generados, $X = P - 1$):

3.1. Calcular el factor de cercanía Pr entre n_i y n_g usando la expresión 1.3.

3.2. Crear un nuevo nodo usando la expresión 1.6 usando n_i , n_g y Pr .

Paso 4: Si $(Z + X) < T$, seleccionar los nodos fronteras(nodos exteriores n_s y nodos interiores n_u)

(Y nuevos nodos son creados $Y = T - (Z + X)$):

4.1. Calcular el vector de desplazamiento w_j usando la expresión 1.11

4.2. Generar $[Y/2]$ nuevos nodos por n_s por la expresión 1.9.

4.3. Generar $Y - [Y/2]$ nuevos nodos por n_u por la expresión 1.10.

Paso 5: Unir y ordenar la población actual con los nodos creados en Pasos 2-4 según su calidad.

Paso 6: Aplicar el operador de limpieza adaptativo, eliminando los nodos más cercanos. Obtenemos B nodos para la próxima iteración

Paso 7: Si $B < P$ entonces

- seleccione los nodos de B y complete la población con $P - B$ los nuevos nodos aleatorios.

Por otra parte ($B \geq P$) entonces

- seleccione los P nodos mejores (calidad) para componer la población de la próxima iteración.

Paso 8: Ir al **Paso 2** si la condición de parada no es cumplida ($c < C$).

Figura 1.4 Pasos de aplicar el algoritmo VMO.

1.2 Algoritmos Genéticos de Codificación Real.

La obtención de individuos nuevos a partir de los que existen es una de las características más interesantes e importantes en el trabajo de los AG. El cruzamiento es el principal operador genético, representa la reproducción sexual, opera sobre dos cromosomas a la vez para generar dos descendientes donde se combinan las características de ambos cromosomas padres. Normalmente, este se aplica sobre parejas de padres, generando dos hijos para cada una de ellas, los cuales se introducen en la población. Sin embargo, también se han presentado operadores de cruce con múltiples descendientes, que generan más de dos descendientes por cada grupo de padres. En este caso, un mecanismo de selección de descendientes limita el número de hijos que entran a formar parte de la nueva población. Estos operadores, que están inspirados en la naturaleza, tratan de sacar más beneficio de los padres mediante el muestreo de un mayor número de posibles soluciones resultantes de su recombinación. Todos los descendientes pueden crearse utilizando el mismo mecanismo de generación de hijos o mediante diferentes mecanismos de generación de descendientes.

Estos algoritmos han manifestados como métodos flexibles y robustos para la solución de los complejos problemas de optimización y su eficiencia ya está probada. El operador de cruce en estos modelos puede verse como un método de búsqueda local, esta ventaja pudiera aprovecharse en la etapa de exploración que realiza el modelo VMO.

Para la codificación de los cromosomas en AG existen distintas formas:

- I. Codificación binaria: En la formulación original de los AG, las soluciones se han codificado usando el alfabeto binario, cada componente del cromosoma está formado por unos o ceros en dependencia del estado y han sido las más utilizadas por la simplicidad de su implementación.
- II. Codificación discreta: En esta representación cada componente del cromosoma está formado un número entero que representa un estado, es decir, representa algún aspecto particular de la solución.
- III. Codificación real: adecuada para problemas con parámetros en dominios continuos. Un cromosoma es un vector de números reales con un tamaño

igual a la longitud del vector solución del problema. Los AG basados en este tipo de codificación se denominan AG con codificación real (AGCR) [12] que serán objeto de estudio en esta investigación ya que nuestro problema está definido en un dominio continuo.

Los métodos II y III permite una mayor precisión y complejidad que el método comparativamente restringido de utilizar sólo números binarios, y a menudo ``está intuitivamente más cerca del espacio de problemas".

El operador de cruce juega un papel central en los AGCR, de hecho puede considerarse como una de las características que definen a estos algoritmos y es uno de los componentes a tener en cuenta para mejorar su eficacia. En estos algoritmos, este operador influye decisivamente sobre el nivel de diversidad en la población, y por ello, es un factor determinante para evitar el problema de la convergencia prematura [13]. Esto explica que el principal esfuerzo en la investigación desarrollada para mejorar los AGCR se centre en la propuesta y estudio de nuevos operadores de cruce.

El operador de cruzamiento en estos modelos se agrupan en diferentes categorías en base a la forma en que generan los genes de los hijos a partir de los genes de los padres como son:

I. Operadores de cruce basados en entornos (OCEs) [14].

Estos operadores generan los genes de los descendientes a partir de valores tomados del entorno asociado a los padres, X_i , Y_i , normalmente a partir de un intervalo $[a_i, b_i]$ cuyos extremos dependen de los valores de los genes de los cromosomas padres. En esta categoría podemos especificar que existen dos subcategorías: los operadores probabilísticos, los cuales generan los genes de los descendientes a partir de valores tomados del entorno asociado a los progenitores mediante distribuciones de probabilidad, normalmente a partir de un intervalo cuyos extremos dependen de los valores de los genes de los cromosomas padres y los deterministas los cuales generan de manera única los descendientes a partir de los valores de los padres.

II. Operadores de cruce híbridos [14].

Una idea interesante en el desarrollo de operadores de cruce consiste en la aplicación simultánea de diversos operadores sobre la población. De hecho, existen diversos estudios dirigidos al estudio de la sinergia producida por la combinación de distintos operadores. Su objetivo es investigar cuando una combinación de operadores de cruce es mejor que el mejor de los operadores que la componen.

Cada operador de cruce dirige la búsqueda hacia una zona diferente del entorno de los padres. La calidad de los elementos de la región visitada depende del problema particular a resolver. Diferentes operadores de cruce se comportarán de manera diferente con respecto a diferentes problemas, incluso en diferentes estados del proceso en el mismo problema. De esta forma, la aplicación simultánea de diferentes operadores de cruce sobre la población podría proporcionar modelos efectivos que serían aplicados a muchos problemas prácticos.

La efectividad final del operador de cruce va a depender del equilibrio entre la diversidad asociada a los mecanismos de generación de descendientes y la presión selectiva derivada del mecanismo de selección de descendientes.

No hay evidencia teórica o experimental que indique la existencia de un operador de cruzamiento universal que obtenga los mejores resultados para todo problema, siendo necesario determinar qué operador se ajusta a las características particulares de cada problema y de cada codificación para obtener buenos resultados.

En nuestro modelo un cromosoma es representado por un vector de números reales con un tamaño igual a la longitud del vector solución del problema, estos vectores son el conjunto de nodos que representan soluciones potenciales del problema de optimización que forman la malla y un cruzamiento entre dos cromosomas padres para generar el descendiente representará el cruzamiento entre dos nodos de nuestra población para la generación de un nuevo nodo.

Basado en esta motivación nuestro estudio se centrará en la incorporación de los operadores de cruce con codificación real más utilizados en la literatura a la etapa de exploración que desarrolla VMO con el objetivo de mejorar las soluciones encontradas por este modelo en dominios continuos.

CAPÍTULO 2. Incorporación de operadores genéticos de cruzamiento con codificación real a VMO.

El cruzamiento permite el intercambio de código genético entre los individuos introduciendo así instancias de nuevas combinaciones de genes. En este capítulo se especificarán los operadores genéticos de cruce que serán incorporados a la etapa de búsqueda desarrollada por el modelo, los detalles que tienen en común así como las diferencias. Estas características servirán para brindar una mayor diversidad a la calidad de las soluciones y de igual modo se describirán las distintas variantes resultantes de incorporar dichos operadores ya sea de manera simple o de forma híbrida a VMO.

2.1 Operadores Genéticos de Cruzamiento con codificación real empleados.

En este apartado se comentarán los operadores de cruce que serán incorporados a nuestro modelo, como son: BLX- α , BLX- α - β , PBX- α , Heurístico, LinearBGA, Geométrico y Aritméticos, para los cuales utilizaremos los acrónimos Op1, Op2, Op3, Op4, Op5, Op6 y Op7 respectivamente para realizar un trabajo más cómodo en la conformación de las tablas. Estos operadores tienen una gran importancia en el desarrollo de esta investigación. Se verán al final de este epígrafe algunas características y diferencias que servirán para entender mejor el comportamiento de los mismos.

2.1.1 Operadores de Cruce Basados en Entornos Probabilísticos.

Los OCEs probabilísticos (ver Figura 2.1) generan los genes de los descendientes a partir de valores tomados del entorno asociado a los padres mediante distribuciones de probabilidad, normalmente a partir de un intervalo cuyos extremos dependen de los valores

de los genes de los cromosomas padres. El grado de diversidad de estos operadores puede variar dependiendo del valor del parámetro asociado al operador. A mayor valor del parámetro, mayor será la varianza (diversidad) introducida en la población. En la formulación original de estos operadores se generan dos descendientes pero para el caso de estudio solo uno es necesario.

Veremos ahora ejemplos de esta categoría:

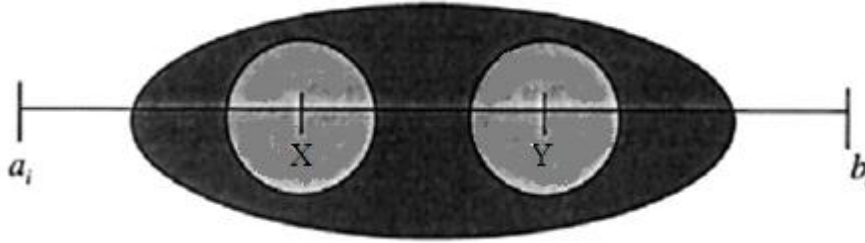


Figura 2.1 Efectos de los OCEs.

Sean $X = (x_1, x_2, \dots, x_n)$ y $Y = (y_1, y_2, \dots, y_n)$ dos cromosomas que fueron seleccionados para aplicarles estos operadores, que para nuestro problema representan nodos. A continuación se describe la forma en la que cada operador de cruce trabaja.

- **Cruzamiento BLX $-\alpha$ (Pino y otros, 2009).**

Este operador llamado también mezcla entrega resultados que no generan pérdida de diversidad, lo cual matemáticamente se refiere a que el valor de α genera un intervalo de ocurrencia para el hijo que puede estar entre los dos padres (en magnitud) o posiblemente en los extremos de este intervalo, es decir puede ser mayor o menor que ambos padres. Seleccionando aleatoriamente una dimensión $i \in [1, 2, \dots, n]$, se obtiene un hijo $H = (h_1, h_2, \dots, h_i, \dots, h_n)$ donde h_i se selecciona de manera (uniformemente) aleatoria del intervalo $[XY_{min} - I * \alpha, XY_{max} + I * \alpha]$ siendo $XY_{max} = \max(X_i, Y_i)$, $XY_{min} = \min(X_i, Y_i)$, e $I = XY_{max} - XY_{min}$. En la Figura 2.2 se muestra el efecto de este operador.

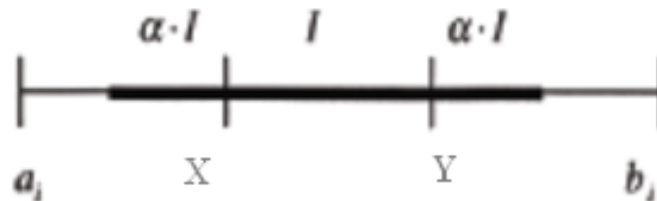


Figura 2.2 Operador BLX $-\alpha$.

- **Cruzamiento BLX - α - β (Figura 2.3).**

Supongamos que X es el padre más idóneo, se genera un hijo $H = (h_1, \dots, h_i, \dots, h_n)$, $I = XY_{max} - XY_{min}$, donde h_i es un valor tomado al azar escogido del intervalo $[X_i - I^*\alpha, Y_i + I^*\beta]$ si $X_i \leq Y_i$, o $[X_i - I^*\beta, Y_i + I^*\alpha]$ en otro caso.

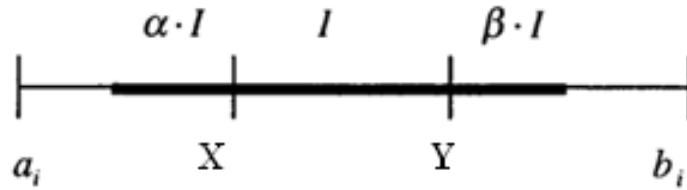


Figura 2. 3 Operador BLX - α - β .

- **Cruzamiento PBX - α (Figura 2.4).**

Es similar al operador de cruzamiento BLX- α , sabiendo que $X, Y \in [a_i, b_i]$, se obtiene un hijo $H = (h_1, h_2, \dots, h_i, \dots, h_n)$ donde h_i se selecciona aleatoriamente del intervalo $[l_i, u_i]$ siendo $l_i = \max(a_i, x_i - I^*\alpha)$, $u_i = \min(b_i, x_i + I^*\alpha)$. Este operador genético obtiene soluciones cercanas al segundo nodo y_i con α en el intervalo de 0.5 a 1.

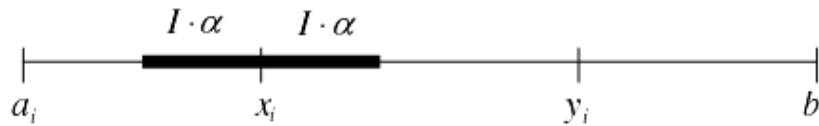


Figura 2. 4 Operador PBX - α .

- **Cruzamiento heurístico (Figura 2.5).**

Asumimos que X es mejor que el cromosoma Y en cuanto a la función de calidad. Entonces, se genera $H = (h_1, \dots, h_i, \dots, h_n)$, donde $h_i = u^*(X_i - Y_i) + X_i$, donde u es un valor aleatorio en el intervalo de 0 a 1.

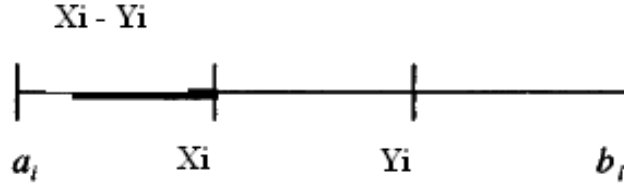


Figure 2.5 Operador Heurístico.

- **Cruzamiento LinearBGA(Figura 2.6)**

Teniendo las mismas consideraciones previamente mencionadas $H_i = X_i + r_i * \gamma * A$, en esta expresión el signo de “+” o “-” es utilizando con una probabilidad de 0.9, $r_i = 0.5 * (b_i - a_i)$,

$\gamma = \sum_{k=0}^{15} \alpha_k * 2^{-k}$, donde α es un valor aleatorio en el intervalo $[0,1]$ con $p(\alpha=1)=1/16$ y $A = [(Y_i - X_i) / |X - Y|]$.

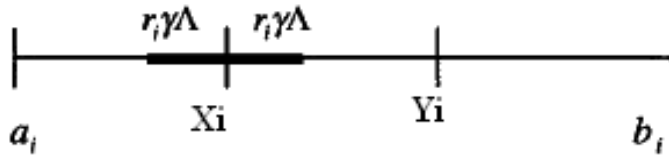


Figure 2.6 Operador LinearBGA.

Todos ellos muestran dos importantes características:

- En su definición incluyen parámetros que determinan la extensión asociada con las distribuciones de probabilidad utilizadas para generar los hijos. El grado de diversidad inducido por estos operadores se puede ajustar variando el valor de estos parámetros.
- Definen una distribución de probabilidad de los hijos basada en la distancia de los padres. Si los padres están muy cerca el uno del otro, los hijos generados podrían distribuirse densamente alrededor de los padres. En cambio, si los padres están alejados entre sí, los hijos apenas se distribuirán a su alrededor.

Una vez que el AGCR ha encontrado zonas prometedoras del espacio de búsqueda, busca solo sobre una pequeña porción del vecindario alrededor de cada punto del espacio, con lo que se realizan múltiples exploraciones sobre puntos individuales en paralelo sobre sucesivas generaciones de una población. Con el paso de las generaciones, el AGCR pierde

diversidad debido a la presión selectiva. Bajo esta circunstancia, la naturaleza adaptativa de los OCEs probabilistas (pueden generar descendientes adaptativamente de acuerdo a la distribución de los padres sin ningún parámetro adaptativo) permite la creación de hijos distribuidos densamente alrededor de los padres, induciendo una efectiva búsqueda local.

2.1.2 Operadores de Cruce Basados en Entornos Deterministas.

Dentro de la categoría de operadores de codificación real tenemos igualmente los operadores de cruce deterministas que generan los genes de los descendientes de forma determinista (no hay componente aleatorio) a partir de valores tomados de los genes de los padres:

- **Cruzamiento geométrico (Figura 2.7).**

Se genera un hijo $H=(h_1, \dots, h_i, \dots, h_n)$, donde $h_i = x_i^w * y_i^{1-w}$, donde $w \in [0,1]$.

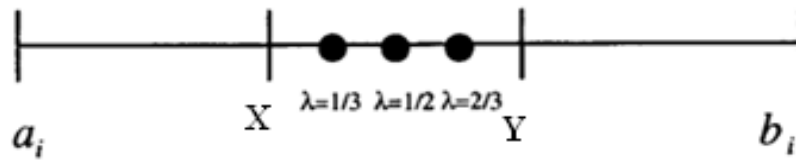


Figura 2. 7 Operador Geométrico para diferentes valores de w .

- **Cruzamiento aritmético (Figura 2.8).**

Se selecciona aleatoriamente una constante $\lambda \in [0,1]$ se obtienen un hijo de la siguiente forma $H=(h_1, \dots, h_i, \dots, h_n)$ donde $H_i = \lambda * X_i + (1-\lambda) * Y_i$.

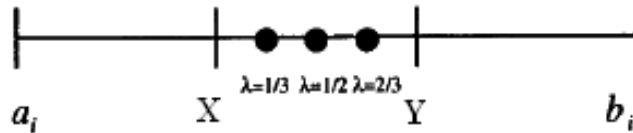


Figura 2.8 Operador Aritmético para diferentes valores de λ .

La constante λ puede no cambiar durante todo el proceso, en ese caso, se trata de un cruzamiento aritmético uniforme, en caso contrario, es un cruzamiento aritmético no

uniforme. Con $\lambda=0.5$ es un ejemplo claro de un operador explotador de cruzamiento, contrariamente este operador mostrará exploración con $\lambda > 1$ y $\lambda < 0$.

2.1.3 Principales características y diferencias entre ambos operadores.

A continuación veremos algunas características y diferencias entre los operadores mencionados anteriormente:

- I. Entre los operadores de cruce existe una diferencia muy importante: los OCEs probabilísticos, a diferencia de los determinísticos, son operadores con componentes aleatorios. Esto quiere decir que, dado un mismo par de padres los primeros puede generar un conjunto de soluciones distintas, mientras que los últimos, dado el mismo par de soluciones padres, siempre se genera el mismo descendiente.
- II. En los operadores determinísticos siempre se reducen el radio de distribución de las nuevas soluciones, mientras que, los probabilísticos, el parámetro determina el grado de distribución de los nuevos individuos, pudiéndose reducir o aumentar la diversidad de la población.
- III. Los probabilísticos tiene un comportamiento autoadaptativo en el que la generación de los descendientes se realiza en función de la distribución de los padres, adaptándose a ésta, sin necesidad de ningún parámetro adaptativo. El parámetro en los operadores deterministas adolece de dicho comportamiento.

2.2 Incorporación de los operadores genéticos de cruzamiento con codificación real a la etapa de exploración de VMO.

A la hora del diseño de la incorporación de los distintos operadores de cruzamiento es importante preservar la idea original de VMO manteniendo la manera organizada en que se realiza la exploración. En lo que sigue se explica la forma en que se utilizaron los operadores de cruce en la exploración local y global de VMO.

Cambios en la etapa de exploración local y global en el modelo VMO con la incorporación de los operadores genéticos de cruzamiento.

Para analizar el comportamiento del modelo VMO se realizarán diferentes variantes cambiando las funciones en los pasos 2 y 3 del algoritmo por los operadores descritos anteriormente ya sea de manera simple o la hibridación de ellos según sus características. Dado que, en AGCRs, los niveles de varianza en la población y por tanto de diversidad difieren del uso de unos operadores de cruce a otros, puede ser interesante combinar distintos operadores que aporten características complementarias en un intento de conseguir una mejor calidad de las soluciones. Los operadores de cruce híbridos combinan distintos operadores y facilitan el estudio de la sinergia entre dichos operadores.

Los pertenecientes al grupo de OCEs tanto probabilista como determinista proporcionan diferentes estilos de búsqueda, de forma que la combinación de diferentes OCEs nos permiten generar grupos de descendientes con una distribución de probabilidad diferente, para crear los genes de los hijos en zonas restringidas del espacio de búsqueda alrededor de las regiones marcadas por los genes de los padres. Con PNX cada gen del hijo se genera en el vecindario del correspondiente gen de uno de los padres. En cambio, BLX no muestra esta clara preferencia hacia zonas cercanas a los padres. Al combinar estos operadores se conseguirá una mayor diversidad y permitirán hacer una mayor profundización a la hora de la investigación aportando características complementarias.

Las notaciones manipuladas en este epígrafe son equivalentes a las utilizadas a la hora de declaración de estas funciones originalmente en el capítulo anterior. Seguidamente se muestran los cambios sufridos los pasos 2 y 3 para conformar las distintas variantes de experimentación:

Paso2: Generación de nodos en dirección a los extremos locales: En esta etapa de exploración que realiza VMO 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, es sustituida por la ecuación:

$$n_z = \text{OCEs}(n_i, n_i^*) \quad (2.1)$$

donde OCEs es sustituido por cualquier operador ya sea probabilista o determinista descrito en el epígrafe anterior.

De la misma manera para conformar otras variantes de estudio la esta función es suplida por el operador híbrido definido por la expresión:

$$v_j^z = \begin{cases} \text{OperadorDter}(n_i, n_i^*), & \text{si } |\overline{m}_j - v_j^{i*}| > \xi_j \text{ y } U[0, 1] \leq \text{Pr}(n_i, n_i^*) \\ \text{OperadorProbabilístico}(n_i, n_i^*), & \text{si } |\overline{m}_j - v_j^{i*}| \leq \xi_j \\ \text{OperadorProbabilístico}(n_i, n_i^*), & \text{en otro caso} \end{cases} \quad (2.2)$$

Paso3: Generación de nodos en dirección al extremo global: En este paso se generan nodos intermedios entre los nodos iniciales de la malla y el extremo global, en este caso de estudio la función H es sustituida por la expresión:

$$n_x = \text{OCEs}(n_i, n_g) \quad (2.3)$$

donde OCEs es sustituido como en el caso de la búsqueda local, por cualquier operador descrito en el epígrafe anterior.

De igual manera que en el Paso2, la función H es cambiada por el operador híbrido definido por la expresión:

$$n_j^x = \begin{cases} \text{OperadorDter}(v_j^i, v_j^g), & \text{si } U[0, 1] \leq \text{Pr}(n_i, n_g) \\ \text{OperadorProbabilístico}(v_j^i, v_j^g), & \text{en otro caso} \end{cases} \quad (2.4)$$

Se creó otro operador híbrido para conformar una variante distinta de estudio con características diferentes a los operadores definidos anteriormente con el objetivo de ampliar la variedad, la cual fue insertada en el Paso2 y en el Paso3 del modelo definido por la expresión:

$$n_j^{Hib} = \begin{cases} \text{Aritmético}(n_i, n_d), & \text{si } va = 0 \\ \text{Geométrico}(n_i, n_d), & \text{si } va = 1 \\ BLX - \alpha (n_i, n_d), & \text{si } va = 2 \\ BLX - \alpha - \beta(n_i, n_d), & \text{si } va = 3 \\ PBX - \alpha (n_i, n_d), & \text{si } va = 4 \\ \text{Heurístico}(n_i, n_d), & \text{si } va = 5 \\ \text{LinearBGA}(n_i, n_d), & \text{si } va = 6 \end{cases} \quad (2.5)$$

donde n_j^{Hib} representa la j -ésima componente del nodo híbrido creado, va es una función aleatoria que devuelve números enteros entre 0 y 6 con igual probabilidad y n_d es un nodo que es cambiado por el mejor vecino (n_i^*) si insertamos el operador híbrido en el Paso2 o por el extremo global (n_g) si lo insertamos en el Paso3.

Los operadores híbridos tienen esta estructura para seleccionar solo aquellos operadores de cruce que tuvieran un comportamiento similar al que tenía la función antes de la incorporación de estos, para no perder la funcionalidad que tienen tanto la función de generación local como la global dentro del modelo. En estas variantes se aplica un cambio únicamente, es decir, solo se incorpora uno de los dos pasos anteriormente expuestos, el otro se mantiene sin alterar para observar cómo influye este cambio en el comportamiento del modelo. Las variantes serán separadas para analizarlas: las locales y globales, para seleccionar las mejores en cada uno de estos grupo y en cada una de las dimensiones estudiadas para después compararlas con el modelo original.

2.3 Descripción de los test estadísticos y funciones de prueba a utilizar.

Seguidamente se mostrará una descripción de las pruebas que se realizarán en el próximo capítulo teniendo en cuenta su funcionalidad básica y la utilidad de su empleo en esta investigación y al finalizar el epígrafe mostraremos las funciones utilizadas para la experimentación.

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 χ_F^2 [17] era excesivamente conservativa y definieron una variante con un mejor comportamiento, indicada en la ecuación siguiente:

$$F_F = \frac{(N-1)\chi_F^2}{N \cdot (k-1) - \chi_F^2} \quad (2.6)$$

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:

$$z = (R_i - R_j) / \sqrt{\frac{k \cdot (k+1)}{6 \cdot N}} \quad (2.7)$$

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.

Como problema de prueba para estudiar el caso continuo, se presenta la aproximación (minimización) de un conjunto de 20 funciones multimodales (F_6 - F_{25}), publicadas como marco experimental obligatorio de una sección especial organizada por la IEEE en el Congreso de Computación Evolutiva (CEC 2005) [21] celebrado en Londres y, desde entonces, de obligada utilización para el estudio actual del comportamiento de las meta-heurísticas para dominios continuos. Seguidamente se enumeran las funciones utilizadas para la experimentación, así como el dominio de cada dimensión.

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

Todas las funciones han sido desplazadas para asegurar que nunca se encuentre su óptimo en el centro del espacio de búsqueda y para que sean escalables. Además, en F_7 el óptimo no se encuentra dentro del rango permitido de cada variable y en F_{12} el dominio de búsqueda no está limitado.

CAPÍTULO 3. Análisis Experimental de los resultados.

En este capítulo se presentan los análisis de los resultados de las distintas variantes del diseño de VMO con la incorporación de los operadores genéticos de cruce con codificación real descritos anteriormente, así como todo el marco experimental y el proceso de experimentación desarrollado en este trabajo. Dado que la elección de un operador puede poseer mucha influencia en los resultados, se van a comparar algoritmos que se diferencian en los operadores empleados. Comparando estos algoritmos entre sí se estudia la conveniencia de cada uno de esos operadores para cada valor de dimensión de las estudiadas. De esa forma, se puede no sólo recomendar un esquema VMO con la inserción de los operadores, sino también la combinación de operadores que consideremos más óptima, es decir el operador híbrido ya sea de la búsqueda local o global de mejor comportamiento.

Cada una de las variantes con la incorporación del un operador de cruce a nuestro modelo ya sea en su forma simple o de manera híbrida se representará con la notación VMO-OPERADOR y se simbolizará el modelo original como VMO-PURO, o sea sin la incorporación de ningún operador.

3.1 Técnicas Estadísticas para validar los resultados obtenidos.

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. Entre las más utilizadas están los test paramétricos [15], que son los más empleados y los test no paramétricos [16], los cuales haremos uso en esta investigación pues los resultados obtenidos por las metaheurísticas no cumplen las condiciones requeridas para poder usar de forma correcta comparaciones paramétricas, como es la distribución normal y mantener una varianza homogénea.

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.

A continuación se presenta el marco estadístico utilizado para validar los resultados:

1. Se aplica la prueba de Iman-Davenport [17] para detectar diferencias entre un conjunto de algoritmos.
2. Si no se detectan deferencias 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 [18] 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 [19].

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. Este modelo experimental de comparaciones ha sido utilizado anteriormente en distintos trabajos [20].

3.2 Marco Experimental.

A continuación mostramos un análisis experimental de los resultados obtenidos por las distintas alternativas utilizadas. Para dichos análisis, se confeccionaron los siguientes grupos de comparación tanto para dimensión 10, 30 y 50:

1. **Etapa de búsquedas local.**
 - a. Seleccionar el mejor OCEs.
 - b. Seleccionar la mejor variante híbrida.
 - c. Análisis para escoger la variante de mejor comportamiento en la búsqueda local comparando las seleccionadas en: 1a y 1b.
2. **Etapa de búsqueda global.**
 - a. Seleccionar el mejor OCEs.
 - b. Seleccionar la mejor variante híbrida.
 - c. Análisis para escoger la variante de mejor comportamiento en la búsqueda global comparando las seleccionadas en: 2a y 2b.
3. Seleccionar el mejor modelo para cada una de las dimensiones que de mejor comportamiento a nuestro problema comparando las variantes 1c, 2c y el modelo original.

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

Los operadores de codificación real basados en entornos tanto probabilistas como los deterministas como se reflejó en el Capítulo 2 fueron los seleccionados para conjuntamente con el modelo VMO conformar las distintas variantes de experimentación. A continuación se muestran los estudios realizados, para ello se utilizarán como prueba las funciones presentadas en el capítulo anterior.

3.3.1 Etapa de búsqueda local.

En este epígrafe cada uno de los operadores serán incorporados al Paso 2 encargado de la búsqueda local del modelo para observar la calidad de las soluciones brindadas por las distintas variantes que se forman.

Se analizará primero el comportamiento de los operadores en su forma simple, y se seleccionará la mejor variante para cada una de las dimensiones, de igual manera se hará el análisis para los híbridos y al final se comparará las dos mejores variantes (simple e híbrida) para seleccionar la de mejor comportamiento en esta etapa de búsqueda.

3.3.1.1 Comparaciones de las distintas variantes de aplicar los operadores genéticos de cruzamiento.

Se realizó un estudio comparativo entre las distintas variantes diseñadas de incorporación de los operadores de cruce en su forma simple a VMO para dimensión 10, 30 y 50 (*Anexo A, Tabla 1, Tabla 2, Tabla 3*).

La Tabla 3.1 muestra los resultados del test de Iman-Davenport para cada grupo notándose que existen diferencias significativas entre las variantes algorítmicas comparadas ($p\text{-value} < \alpha$), destacándose para cada dimensión, el modelo con la incorporación del operador PBX- α (VMO-PBX- α) por tener menor ranking.

Tabla 3.1 Resultados de la prueba de Iman-Davenport.

<i>Dimensión</i>	<i>p-value</i>	<i>Valor de α</i>	<i>Hipótesis</i>
10	1.47E-11	0.050	Rechazada
30	5.59E-16	0.050	Rechazada
50	6.00E-14	0.050	Rechazada

Al ser rechazada la hipótesis nula en el todos los grupos, se acepta la hipótesis alternativa, o sea, la existencia de diferencias significativas dentro de cada grupo.

Para detectar entre que modelos existen tales diferencias se aplicó el test de Holm cuyos resultados para el primer grupo aparecen en la Tabla 3.2 en la cual se puede apreciar que el modelo VMO-PBX- α para dimensión 10 (seleccionado como algoritmo de control al poseer menor ranking) difiere significativamente de los modelos VMO-Geométrico, VMO-LinearBGA, VMO-Aritmético, VMO-BLXAlphaBeta y VMO-BLXAlpha. En el caso del modelo VMO-Heurístico su comportamiento no difiere significativamente con el tomado como control.

Tabla 3.2 Tabla de Holm para $\alpha=0.05$.

<i>Algoritmo</i>	$Z = (R_i - R_j) / \sqrt{\frac{k - (k+1)}{6 - N}}$	<i>p-value</i>	<i>Holm</i>	<i>Hipótesis</i>
VMO-Op6	5.744	9.20E-9	0.008	Rechazada
VMO-Op5	4.730	2.23E-6	0.01	Rechazada

<i>Algoritmo</i>	$Z = (R_i - R_j) / \sqrt{\frac{k - (k+1)}{6 - N}}$	<i>p-value</i>	<i>Holm</i>	<i>Hipótesis</i>
VMO-Op7	4.392	1.11E-5	0.0125	Rechazada
VMO-Op2	3.324	0.001	0.016	Rechazada
VMO-Op1	2.365	0.018	0.025	Rechazada
VMO-Op4	1.314	0.188	0.05	Aceptada

Se puede observar el análisis para el segundo grupo (dimensión 30) en la Tabla 3.3 en la que se refleja que el modelo VMO-PBX- α (seleccionado como algoritmo de control al poseer menor ranking) difiere significativamente de los modelos VMO-LinearBGA, VMO-Geométrico, VMO-Aritmético y VMO-BLXAlphaBeta, no siendo así con los modelos VMO-BLX- α y VMO-Heurístico .

Tabla 3.3 Tabla de Holm para $\alpha=0.05$.

<i>Algoritmo</i>	$Z = (R_i - R_j) / \sqrt{\frac{k - (k+1)}{6 - N}}$	<i>p-value</i>	<i>Holm</i>	<i>Hipótesis</i>
VMO-Op5	5.938	2.87E-9	0.007	Rechazada
VMO-Op6	5.002	5.65E-7	0.008	Rechazada
VMO-Op7	4.776	1.78E-6	0.01	Rechazada
VMO-Op2	3.905	9.41E-5	0.012	Rechazada
VMO-Op1	2.323	0.020	0.016	Aceptada
VMO- Op4	1.710	0.087	0.025	Aceptada

En la Tabla 3.4 ilustra el estudio para el tercer grupo(dimensión 50), demostrando que el modelo VMO-PBX- α difiere significativamente de los modelos VMO-LinearBGA, VMO-Geométrico, VMO-Aritmético, VMO-BLXAlphaBeta y VMO-BLXAlpha. Se observa que las variantes con la incorporación de los operadores BLX-Alpha y Heurístico obtienen resultados que no difieren significativamente con el algoritmo tomado de control.

Tabla 3.4 Tabla de Holm para $\alpha=0.05$.

<i>Algoritmo</i>	$Z = (R_i - R_j) / \sqrt{\frac{k - (k+1)}{6 - N}}$	<i>p-value</i>	<i>Holm</i>	<i>Hipótesis</i>
------------------	--	----------------	-------------	------------------

VMO-Op5	5.744	9.19E-9	0.00714	Rechazada
VMO-Op6	4.873	1.09E-6	0.00833	Rechazada
VMO-Op7	3.066	1.78E-6	0.01	Rechazada
VMO-Op2	3.905	0.002	0.015	Rechazada
VMO-Op1	2.485	0.012	0.016	Rechazada
VMO-Op4	2.097	0.035	0.025	Aceptada

En este análisis de los operadores genéticos de manera simple incorporados a la búsqueda local se observa que el operador PBX- α fue el de mejor comportamiento para todas las dimensiones analizadas.

3.3.1.2 Comparaciones de las distintas variantes de hibridación de los operadores genéticos de cruzamientos.

A continuación mostramos un proceso donde se realiza un análisis similar al anterior, ahora utilizaremos los mejores operadores seleccionados en el primer proceso de comparación (PBX- α e Heurístico) y reajustaremos la función F encargada de la búsqueda local en nuestro modelo siguiendo las expresiones 2.2, en la cual utilizaremos dichos operadores con un operador determinístico diferente y la 2.5 que haremos uso de todos para conformar las distintas variantes de estudio (Anexos A, Tabla 7, Tabla 8, Tabla 9).

Se aplicó la prueba de Iman-Davenport para detectar diferencias en los grupos de las variantes híbridas. La Tabla 3.5 muestra los resultados de dicha prueba mostrando la aceptación de la hipótesis nula en los tres grupos, o sea, no existen diferencias significativas entre las variantes algorítmicas comparadas para cada dimensión ($p\text{-value} > \alpha$).

Tabla 3.5 Resultados de la prueba de Iman-Davenport.

<i>Dimensión</i>	<i>p-value</i>	<i>Valor de α</i>	<i>Hipótesis</i>
10	0.204	0.050	Aceptada
30	0.704	0.050	Aceptada
50	0.584	0.050	Aceptada

Para cada una de la dimensiones se formaron tres variantes híbridas con las estructuras mencionadas en el capítulo anterior, dos siguiendo la expresión 2.2, cada una con un operador determinista diferente, lo que varía son los operadores probabilistas que en cada dimensión se eligieron de los que obtuvieron mejores resultados de los comparados en su forma simple. Para dimensión 10 los operadores probabilistas fueron PBX-Alpha y Heurístico, se formaron dos variantes: VMO-HíbridoLocal1 y VMO-HíbridoLocal2 respectivamente, y una variante **VMO-HíbridoLocal3** la cual sigue la expresión 2.5 que resultó ser la de mejor comportamiento.

En el caso de la dimensión 30 los operadores probabilistas coincidieron con los de la dimensión 10 y se formaron VMO-HíbridoLocal4 y HíbridoLocal5 también con los diferentes operadores deterministas respectivamente y resultó ser de mejor comportamiento el **VMO-HíbridoLocal6** la cual seguía la expresión 2.5.

En el caso de la dimensión 50 se formaron **VMO-HíbridoLocal7**, VMO-HíbridoLocal8 y VMO-HíbridoLocal9 como se ha explicado anteriormente, también coincidieron los operadores probabilistas, en este análisis el de mejor comportamiento fue el primero mencionado anteriormente, formado por los operadores dichos y el operador determinista geométrico.

3.3.1.3 Análisis para seleccionar la mejor variante en la búsqueda local.

Ahora haremos un análisis de los operadores de mejor comportamiento para cada una de las dimensiones, comparando la mejor variante de operador simple y la mejor híbrida. La Tabla 3.6 muestra las mejores variantes seleccionadas anteriormente en cada uno de los análisis.

Tabla 3.6 Mejores variantes en la etapa de Búsqueda Local

<i>Dimensión</i>	<i>Operador Simple</i>	<i>Operador Híbrido</i>
10	PBX- α	VMO-HíbridoLocal3
30	PBX- α	VMO-HíbridoLocal6
50	PBX- α	VMO-HíbridoLocal7

La Tabla 3.7 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. Se le aplicó la prueba a los modelos VMO-PBX- α y VMO-HíbridoLocal3 (Ver Tabla 3.6) para detectar qué variante posee mejor comportamiento para la búsqueda local para dimensión 10 y detectar si ese comportamiento es significativamente notable.

Puede apreciarse que en entre los dos modelo las diferencias son significativas, con un mejor comportamiento el modelo VMO-PBX- α , este operador fue menor en mayores ocasiones que el otro y como estamos minimizando nos quedaremos con este.

Tabla 3.7 Resultados de la prueba de Wilcoxon para dimensión 10.

R^-	R^+	Sig. Asintótica	Hipótesis
VMO-Op3	VMO-HíbridoLocal3		
189.00	21.00	0.002	Rechazada

Haremos un análisis similar pero ahora comparando las dos mejores variantes de dimensión 30: la simple VMO-PBX- α y la híbrida VMO-HíbridoLocal6 como se ilustra en la Tabla 3.6, en la siguiente tabla se muestran los resultados de aplicar dicho test.

Tabla 3.8 Resultados de la prueba de Wilcoxon para dimensión 30.

R^-	R^+	Sig. Asintótica	Hipótesis
VMO-Op3	VMO-HíbridoLocal6		

204.00	6.00	0.000	Rechazada
---------------	-------------	--------------	------------------

En este caso la hipótesis principal es rechazada, por lo que hay diferencias significativas entre los modelos comparando, se puede apreciar una mejora a favor del modelo que utiliza el operador PBX- α en la búsqueda local.

Haremos un análisis similar pero ahora comparando las dos mejores de dimensión 50, como se muestra en la Tabla 3.6 para la búsqueda local resultó ser el PBX- α y el VMO-HíbridoLocal7 como mejor variante híbrida. La Tabla 3.8 muestra los resultados.

Tabla 3.9 Resultados de la prueba de Wilcoxon para dimensión 50.

R⁻	R⁺	Sig. Asintótica	Hipótesis
VMO-Op3	VMO- HíbridoLocal7		
171.00	0.00	0.000	Rechazada

En este caso también la hipótesis nula se rechaza por lo que existen diferencias entre los grupos, con un comportamiento significativamente notable a favor del operador PBX- α incorporado a la búsqueda local.

Hasta el momento podemos concluir que el operador PBX- α incorporado a la búsqueda local es la mejor variante de las comparadas en este trabajo para cada una de las dimensiones analizadas.

3.3.2.1 Etapa de búsqueda global.

En este epígrafe haremos un análisis similar al realizado anteriormente para el Paso2 donde los operadores descritos en el capítulo anterior serán incorporados a la etapa de búsqueda global, o sea al Paso 3 del modelo. Los operadores híbridos en este epígrafe serán conformados siguiendo las expresiones 2.4 y 2.5.

3.3.2.2 Comparaciones de las distintas variantes de aplicar los operadores genéticos de cruzamiento.

A continuación mostramos un proceso de experimentación similar a los realizados con anterioridad, en este caso los operadores de cruce se incorporan en su forma simple a la etapa de búsqueda global (*Anexo A, Tabla 4, Tabla 5, Tabla 6*).

Al aplicarse la prueba de Iman-Davenport no fueron aceptadas las hipótesis nulas en los dos primeros grupos y por tanto se comprobó la existencia de diferencias notables entre los modelos involucrados, no siendo así en el tercer grupo en el cual fue aceptada esta hipótesis, o sea, las variantes algorítmicas obtienen resultados que no difieren significativamente entre sí. Eso es corroborado por la Tabla 3.10 la cual muestra los resultados de dicho test.

Tabla 3.10 Resultados de la prueba de Iman-Davenport.

<i>Dimensión</i>	<i>p-value</i>	<i>Valor de α</i>	<i>Hipótesis</i>
10	9.44E-7	0.050	Rechazada
30	6.81E-5	0.050	Rechazada
50	0.954	0.050	Aceptada

Para detectar entre que modelos existen tales diferencias de significación para el primer grupo se aplicó el test de Holm cuyos resultados aparecen en la Tabla 3.11 en la cual se puede observar que el modelo VMO-PBX- α para una dimensión 10 (seleccionado como algoritmo de control al poseer menor ranking) difiere significativamente de los modelos VMO-Heurístico y VMO-Aritmético no siendo así con las otras variantes comparadas.

Tabla 3.11 Tabla de Holm para $\alpha=0.05$.

<i>Algoritmo</i>	$Z = (R_i - R_j) / \sqrt{\frac{k - (k+1)}{6 - N}}$	<i>p-value</i>	<i>Holm</i>	<i>Hipótesis</i>
VMO-Op4	4.841	1.29E-6	0.007	Rechazada
VMO-Op7	2.678	0.007	0.01	Rechazada
VMO-Op5	1.904	0.056	0.012	Aceptada
VMO-Op1	1.710	0.087	0.016	Aceptada

VMO-Op2	1.129	0.258	0.025	Aceptada
VMO-Op6	0.871	0.383	0.05	Aceptada

La Tabla 3.12 muestra que el modelo VMO-PBX- α para dimensión 30 difiere significativamente de los modelos VMO-Heurístico, VMO-BLXAlphaBeta y VMO-Geométrico.

Tabla 3.12 Tabla de Holm para $\alpha=0.05$.

<i>Algoritmo</i>	$Z = (R_i - R_j) / \sqrt{\frac{k-(k+1)}{6-N}}$	<i>p-value</i>	<i>Holm</i>	<i>Hipótesis</i>
VMO-Op4	3.614	3.00E-4	0.0083	Rechazada
VMO-Op2	3.066	0.002	0.01	Rechazada
VMO-Op6	2.937	0.003	0.012	Rechazada
VMO-Op1	1.516	0.129	0.016	Aceptada
VMO-Op7	1.387	0.165	0.025	Aceptada
VMO- Op5	0.677	0.497	0.05	Aceptada

3.3.2.3 Comparaciones de las distintas variantes de hibridación de los operadores genéticos de cruzamiento.

A continuación se muestra un proceso donde se realiza un análisis análogo al preliminar, ahora se utiliza el mejor operador seleccionado en el proceso anterior, reajustando la función H siguiendo las expresiones 2.4 y 2.5 como se plasmó en el epígrafe 2.2 del capítulo anterior (*Anexo A, Tabla 10, Tabla 11, Tabla 12*).

Se aplicó la prueba de Iman-Davenport para detectar diferencias en los grupos, la Tabla 3.13 ilustra los resultados de dicha prueba. Se puede observar que no existen diferencias entre las variantes comparadas.

Tabla 3.13 Resultados de la prueba de Iman-Davenport.

<i>Dimensión</i>	<i>p-value</i>	<i>Valor de α</i>	<i>Hipótesis</i>
10	0.723	0.050	Aceptada

30	0.666	0.050	Aceptada
50	0.829	0.050	Aceptada

Para esta comparación se crearon las variantes híbridas de igual manera que en la búsqueda local. Para la dimensión 10 se formaron los modelos **VMO-HíbridoGlobal1** y VMO-HíbridoGlobal2 siguiendo la expresión 2.4 con los operadores deterministas: geométrico y aritmético respectivamente y el operador probabilista es PBX-Alpha, ya que fue el de mejor comportamiento en el análisis de los operadores simples y la variante VMO-HíbridoGlobal3 siguiendo la expresión 2.5, en este caso el VMO-HíbridoGlobal1 resultó ser el más efectivo.

Para la dimensión 30 las variantes fueron VMO-HíbridoGlobal4, VMO-HíbridoGlobal5 y VMO-HíbridoGlobal6 con la misma estructura que las anteriores ya que el operador probabilista coincidió, para esta dimensión el de mejor comportamiento fue el **VMO-HíbridoGlobal5**.

Para la dimensión 50 se crearon las variantes de hibridación VMO-HíbridoGlobal7, VMO-HíbridoGlobal8 y VMO-HíbridoGlobal9 siguiendo la misma estructura que las anteriores, en este caso el operador probabilista fue el LinearBGA para las dos primeras que siguen la expresión 2.4, la de mejor comportamiento fue el **VMO-HíbridoGlobal7**.

3.3.2.4 Análisis para seleccionar la mejor variante en la búsqueda global.

Ahora haremos un análisis del operador de mejor comportamiento para cada una de las dimensiones, comparando el mejor operador simple y la mejor variante híbrida incorporadas al modelo en este caso a la etapa de búsqueda global. La Tabla 3.14 muestra a manera de resumen dichos mejores.

Tabla 3.14 Mejores variantes en la etapa de Búsqueda Global.

Dimensión	Operador Simple	Operador Híbrido
10	PBX- α	VMO-HíbridoGlobal1
30	PBX- α	VMO-HíbridoGlobal5

50	LinearBGA	VMO-HíbridoGlobal7
-----------	------------------	---------------------------

La Tabla 3.15 muestra los resultados de la prueba de Wilcoxon para $\alpha=0.05$ como se realizó anteriormente. Al aplicarle la prueba a los modelos con mejor comportamiento para dimensión 10: VMO-PBX- α y VMO-HíbridoGlobal1 como se muestra en la Tabla 3.14, en este caso se puede observar que hay una mejora a favor del operador simple, y que existen diferencias significativas entre ambos modelos.

Tabla 3.15 Resultados de la prueba de Wilcoxon para dimensión 10.

R⁻	R⁺	Sig. Asintótica	Hipótesis
VMO-Op3	VMO- HíbridoGlobal1		
137.00	34.00	0.025	Rechazada

Haremos un análisis similar pero ahora con los dos mejores de dimensión 30, en este caso coincide el VMO-PBX- α , y la variante híbrida es VMO-HíbridoGlobal5 como se mostró en la Tabla 3.16, en la siguiente tabla se muestran los resultados de dicho test.

Tabla 3.16 Resultados de la prueba de Wilcoxon para dimensión 30.

R⁻	R⁺	Sig. Asintótica	Hipótesis
VMO-Op3	VMO- HíbridoGlobal5		
134.00	36.50	0.033	Rechazada

La hipótesis principal es rechazada, por lo que existen diferencias significativas entre los modelos comparando, se puede apreciar hay una mejora a favor del modelo que utiliza el operador PBX- α en la búsqueda global para esta dimensión.

Para dimensión 50, se compararon el LinearBGA(mejor operador simple) y el VMO-HíbridoGlobal7 como mejor variante híbrida como se pudo apreciar en la Tabla 3.14. En este caso la hipótesis nula se acepta por lo que no existen diferencias entre los grupos, comportándose mejor el operador VMO-HíbridoGlobal7 en la búsqueda global para esta dimensión. La Tabla 3.17 muestra los resultados.

Tabla 3.17 Resultados de la prueba de Wilcoxon para dimensión 50.

R^-	R^+	Sig. Asintótica	Hipótesis
VMO-Op5	VMO- HíbridoGlobal7		
92.00	118.00	0.627	Aceptada

Hasta ahora podemos concluir que el operador PBX- α presenta el de mejor comportamiento de todas las variantes incorporadas a la búsqueda global para dimensión 10 y 30, para la 50 fue el VMO-HíbridoGlobal7.

3.4 Comparaciones con el modelo VMO-Puro.

Escriba Seguidamente como fase final de la experimentación presentada en esta memoria presentamos una comparación entre quienes resultaron las mejores variantes de diseño en cada uno de las secciones anteriores y el modelo VMO-PURO, o sea, sin la incorporación de operadores de cruce a las diferentes etapas de búsqueda de nuestro modelo para cada una de las dimensiones.

Comparación de las variantes de mejor comportamiento con el modelo original.

En este epígrafe haremos una comparación entre los mejores en cada una de las búsquedas y el modelo original. Para dimensión 10 las mejores variantes son: el PBX- α de mejor comportamiento en la búsqueda local y el PBX- α en la global (*Anexo A, Tabla 1, Tabla 4*); para la dimensión 30 el PBX- α en la búsqueda local y el PBX- α en la global (*Anexo A, Tabla 2, Tabla 5*) y para la 50 el PBX- α incorporado a la búsqueda local y el VMO-HíbridoGlobal7 en la búsqueda global (*Anexo A, Tabla 3, Tabla 12*). En la Tabla 3.18 se ilustra cada grupo de comparación (según la dimensión) con las variantes que lo conforman.

Tabla 3.18 Resultados de la prueba de Iman-Davenport.

<i>Dimensión</i>	<i>Búsqueda Local</i>	<i>Búsqueda Global</i>	<i>Modelo Original</i>
10	VMO-PBX- α	VMO-PBX- α	VMO-PURO
30	VMO-PBX- α	VMO-PBX- α	VMO-PURO
50	VMO-PBX- α	VMO-HíbridoGlobal7	VMO-PURO

En la Tabla 3.19 se muestra la aplicación de la prueba de Iman-Davenport para detectar si existen diferencias entre cada uno de los integrantes del grupo.

Se puede observar que existen diferencias de significativas entre los modelos comparados, ya que se rechaza la hipótesis nula ($p\text{-value} > \alpha$) con un mejor comportamiento de modelo VMO-PURO.

Tabla 3.19 Resultados de la prueba de Iman-Davenport.

<i>Dimensión</i>	<i>p-value</i>	<i>Valor de α</i>	<i>Hipótesis</i>
10	9.952E-4	0.050	Rechazada
30	4.634E-8	0.050	Rechazada
50	1.450E-8	0.050	Rechazada

Para detectar entre que modelos existen tales diferencias de significación se aplicó el test de Holm cuyos resultados aparecen en la Tabla 3.20 en la cual se puede observar que el modelo VMO-PURO para una dimensión 10 (seleccionado como algoritmo de control al poseer menor ranking) difiere significativamente de las distintas variantes seleccionadas como las mejores en cada una de las comparaciones realizadas anteriormente.

Tabla 3.20 Tabla de Holm para $\alpha=0.05$ y dimensión 10.

<i>Algoritmo</i>	$Z = (R_i - R_j) / \sqrt{\frac{k - (k+1)}{6 - N}}$	<i>p-value</i>	<i>Holm</i>	<i>Hipótesis</i>
VMO-Op3(Global)	3.399	6.75E-4	0.025	Rechazada
VMO-Op3(Local)	3.004	0.002	0.05	Rechazada

La Tabla 3.21 muestra los resultados de este test para la dimensión 30 en la cual se puede apreciar el modelo VMO-PURO difiere significativamente de todas las distintas variantes algorítmicas seleccionadas como las mejores en cada una de las comparaciones realizadas anteriormente para esta dimensión.

Tabla 3.21 Tabla de Holm para $\alpha=0.05$ y dimensión 30.

<i>Algoritmo</i>	$Z = (R_i - R_j) / \sqrt{\frac{k - (k+1)}{6 - N}}$	<i>p-value</i>	<i>Holm</i>	<i>Hipótesis</i>
------------------	--	----------------	-------------	------------------

VMO-Op3(Global)	5.771	7.87E-9	0.025	Rechazada
VMO-PBX-α(Local)	3.478	5.04E-4	0.05	Rechazada

Para el tercer grupo de comparación nos podemos percatar que la variante VMO-PURO(seleccionado como muestra de control por poseer menor rankings) difiere significativamente de las distintas variantes pertenecientes a este grupo, como se puede apreciar en la Tabla 3.22.

Tabla 3.22 Tabla de Holm para $\alpha=0.05$ y dimensión 50.

<i>Algoritmo</i>	$Z = (R_i - R_j) / \sqrt{\frac{k - (k+1)}{6 - N}}$	<i>p-value</i>	<i>Holm</i>	<i>Hipótesis</i>
VMO-HíbridoGlobal7	6.008	1.87E-9	0.025	Rechazada
VMO-Op3(Local)	3.004	0.002	0.05	Rechazada

Se puede concluir que el modelo original tiene un mejor comportamiento que cada una de las variantes analizadas para cada una de las dimensiones comparadas y las diferencias entre ellas y la variante original son estadísticamente significativas.

Conclusiones

Como resultado de esta investigación se desarrollaron diversas variantes de incorporación de operadores de cruce con codificación real al modelo VMO; cumpliéndose de esta forma el objetivo general planteado, ya que:

1. Se seleccionaron e implementaron los operadores de codificación real BLX- α , BLX- α - β , PBX- α , Heurístico, LinearBGA, Geométrico y Aritmético, los cuales son los más utilizados en la literatura internacional y se incorporaron a la etapa de exploración local y global de VMO.
2. Se diseñaron las distintas variantes de hibridación tomando los operadores que tuvieron un mejor comportamiento para cada una de las dimensiones seleccionadas.
3. La incorporación del operador PBX- α al modelo tanto en la búsqueda local y como en la global para cada una de las dimensiones fue la variante de mejor comportamiento, comparando la incorporación de los operadores simples y los híbridos.
4. Se comprobó que las distintas variantes de utilizar los operadores de cruce de codificación real no mejoran el comportamiento del modelo original.

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:

2. Estudiar el comportamiento de VMO con la incorporación de otros operadores de cruzamiento con codificación real.
3. Analizar nuevas variantes de incorporación de operadores de cruce con codificación real a la etapa de búsqueda en las fronteras de la malla.
3. Estudiar operadores de mutación, con características propias de explotación, con el fin de ser incorporados en el proceso de expansión del modelo original.

REFERENCIAS BIBLIOGRÁFICAS

- [1] A. Díaz, et al., " Optimización Heurística y Redes Neuronales," Madrid, Paraninfo., 1996.
- [2] F. Glover and G. Kochenberger, Eds., *Handbook of Metaheuristics.*: Kluwer Academic Publishers, 2003.
- [3] OSMAN, I. H. & KELLY, J. P. (1996) *Meta-Heuristics: Theory and Applications*, Boston, Ed. Kluwer Academic.
- [4] K. E. Parsopoulos, "Stretching techniques for obtaining global minimizers through Particle Swarm Optimization.," in *Particle Swarm Optimization* Indianapolis, USA., 2001.
- [5] Z. Michalewicz and D. Fogel, *How to solve it*: Springer, 2004.
- [6] D. E. Goldberg, Ed., *Genetic Algorithms in Search* (Optimization and Machine Learning. University of Alabama: Addison-Wesley Publishing Company, 1998.
- [7] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, Piscataway, New York, USA, 1995, pp. 1942-1948.
- [8] K. V. Price, et al. (2005). *Differential Evolution: A Practical Approach to Global Optimization*.
- [9] Larrañaga, P. y J.A. Lozano, *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation* 2002, Kluwer Academic Publishers: Boston/Dordrecht/London.
- [10] T. Munakata, ""Fundamentals of the new Artificial Intelligence: Beyond traditional paradigms,"" 1998.
- [11] M. Yagiura and T. Ibaraki, "On Metaheuristic Algorithms for Combinatorial Operator Problems," *Systems Computation*, vol. 32, pp. 33–55, 2001.
- [12] F. Herrera, et al., "Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis.," *Artificial Intelligence Review*", vol. 12, pp. 265-319, 1998.

- [13]H.-G. Beyer and K. Deb, "On Self-Adaptive Features in Real Parameter Evolutionary Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 3, 250-270, 2001.
- [14]A. M. Sánchez, *et al.*, "Algoritmos Genéticos para Codificación Real con Operador de Cruce Híbrido con Múltiples Descendientes: 2BLX0.5-2FR0.5-2PNX3- 2SBX0.01," 2009.
- [15]J. Luengo, *et al.*, "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*, vol. 36, pp. 7798-7808, 2009.
- [16]D. Sheskin. (2006). *Handbook of parametric and nonparametric statistical procedures*. 1736.
- [17]R. L. Iman and J. M. Davenport, "Approximations of the critical region of the Friedman statistic," *Commun Stat*, vol. 18, pp. 571-595, 1980.
- [18]S. Holm, "A simple sequentially rejective multiple test procedure," *Scand Journal Stat*, vol. 6, pp. 65-70, 1979.
- [19]F. Wilcoxon, "Ajusted p-values for simultaneous inference," *Biometrics*, vol. 48, pp. 1005-1013, 1945.
- [20]A. Y. Puris, "Desarrollo de meta-heurísticas poblacionales para la solución de problemas complejo.," *Ciencia de la computación*, Marta Abreu,, Villa Clara,Cuba, 2009.
- [21] P. Suganthan, *et al.*, "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization," Nanyang Technological University.2005.

ANEXOS

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

DataSet	VMO-Op1	VMO-Op2	VMO-Op3	VMO-Op4	VMO-Op5	VMO-Op6	VMO-Op7
F6	7.18E+03	3.47E+03	5.59E+02	8.62E+02	1.46E+04	1.90E+04	1.21E+04
F7	6.37E+02	6.49E+02	6.38E+02	6.37E+02	6.54E+02	6.56E+02	6.52E+02
F8	2.05E+01	2.03E+01	2.04E+01	2.04E+01	2.04E+01	2.05E01	2.05E+01
F9	5.42E+01	3.18E+01	1.16E+01	1.95E+01	3.57E+01	4.49E+01	4.19E+01
F10	1.90E+01	2.74E+01	1.55E+01	1.83E+01	2.66E+01	3.15E+01	3.72E+01
F11	5.02E+00	6.08E+00	4.39E+00	4.37E+00	5.27E+00	6.03E+00	5.68E+00
F12	8.53E+02	3.35E+03	5.05E+02	1.67E+02	1.89E+03	3.83E+03	5.33E+03
F13	2.73E+00	1.20E+00	1.21E+00	1.85E+00	2.32E+00	2.69E+00	3.03E+00
F14	3.05E+00	3.09E+00	3.30E+00	2.76E+00	3.25E+00	3.16E+00	3.13E+00
F15	3.19E+02	2.31E+02	2.15E+02	3.29E+02	3.67E+02	3.28E+02	1.77E+02
F16	1.44E+02	1.51E+02	1.29E+02	1.44E+02	1.52E+02	1.60E+02	1.60E+02
F17	1.64E+02	6.08E+00	3.57E+02	1.74E+02	1.83E+02	5.25E+02	1.61E+02
F18	9.05E+02	9.36E+02	8.97E+02	9.16E+02	9.59E+02	8.87E+02	9.59E+02
F19	8.59E+02	9.33E+02	8.96E+02	9.37E+02	9.87E+02	9.31E+02	9.77E+02
F20	8.70E+02	9.58E+02	8.77E+02	9.52E+02	9.35E+02	9.63E+02	9.48E+02
F21	8.10E+02	9.91E+02	7.52E+02	9.73E+02	9.89E+02	1.11E+03	1.06E+03
F22	8.21E+02	8.10E+02	8.12E+02	8.07E+02	8.30E+02	8.34E+02	7.98E+02
F23	8.75E+02	1.13E+03	6.16E+02	7.97E+02	8.69E+02	8.77E+02	6.61E+02
F24	2.37E+02	3.85E+02	2.00E+02	2.00E+02	8.53E+02	9.40E+02	6.85E+02
F25	1.80E+03	1.83E+03	1.79E+03	1.76E+03	1.88E+03	1.88E+03	1.87E+03

Tabla 1. Valores obtenidos para cada una de las funciones con los modelos VMO-Operadores de Cruce con dimensión 10 en la etapa de búsqueda local.

DataSet	VMO-Op1	VMO-Op2	VMO-Op3	VMO-Op4	VMO-Op5	VMO-Op6	VMO-Op7
F6	7.71E+08	5.97E+09	8.63E+06	2.61E+06	3.14E+08	3.74E+09	3.06E+09
F7	2.71E+03	3.03E+03	2.72E+03	2.71E+03	3.17E+03	3.21E+03	3.22E+03
F8	2.09E+01	2.09E+01	2.10E+01	2.10E+01	2.10E+01	2.10E+01	2.09E+01
F9	5.21E+03	2.65E+02	1.20E+02	3.19E+02	1.17E+03	1.01E+03	1.01E+03
F10	2.65E+02	1.20E+02	3.40E+02	3.81E+02	3.81E+02	3.19E+02	3.19E+02
F11	2.73E+01	3.04E+01	1.92E+01	2.69E+01	3.24E+01	3.08E+01	2.94E+01
F12	2.25E+05	1.82E+05	3.45E+04	6.64E+04	3.12E+05	2.09E+05	2.79E+05
F13	1.18E+02	1.04E+01	8.69E+00	1.86E+01	2.06E+02	1.40E+02	1.59E+02
F14	1.27E+01	1.27E+01	1.23E+01	1.25E+01	1.27E+01	1.26E+01	1.27E+01
F15	6.23E+02	6.38E+02	3.99E+02	4.95E+02	7.61E+02	7.38E+02	7.23E+02
F16	3.93E+02	5.30E+02	2.65E+02	4.72E+02	4.78E+02	4.83E+02	4.54E+02
F17	3.93E+02	5.30E+02	3.57E+02	4.21E+02	6.24E+02	5.25E+02	5.25E+02
F18	1.01E+03	1.04E+03	9.18E+02	9.90E+02	1.13E+03	1.09E+03	9.24E+02
F19	1.01E+03	1.04E+03	9.34E+02	9.80E+02	1.11E+03	1.11E+03	1.11E+03
F20	1.06E+03	1.08E+03	9.24E+02	9.55E+02	1.09E+03	1.09E+03	1.10E+03
F21	1.21E+03	1.25E+03	6.17E+02	1.09E+03	1.24E+03	1.24E+03	1.26E+03
F22	1.09E+03	1.14E+03	1.07E+03	1.13E+03	1.15E+03	1.20E+03	1.22E+03
F23	1.18E+03	1.27E+03	6.15E+02	1.05E+03	1.25E+03	1.24E+03	1.23E+03
F24	1.26E+03	1.34E+03	8.65E+02	1.13E+03	1.29E+03	1.28E+03	1.28E+03
F25	1.65E+03	1.65E+03	1.68E+03	1.65E+03	1.72E+03	1.72E+03	1.73E+03

Tabla 2. Valores obtenidos para cada una de las funciones con los modelos VMO-Operadores de cruce con dimensión 30 en la etapa de búsqueda local.

DataSet	VMO-Op1	VMO-Op2	VMO-Op3	VMO-Op4	VMO-Op5	VMO-Op6	VMO-Op7
F6	7.93E+09	3.27E+10	8.63E+06	2.57E+06	1.82E+08	1.60E+10	2.10E+10
F7	3.71E+03	4.38E+03	3.71E+03	3.71E+03	4.69E+03	4.66E+03	4.66E+03
F8	2.11E+01	2.11E+01	2.11E+01	2.11E+01	2.11E+01	2.11E+01	2.11E+01
F9	1.86E+04	5.40E+02	2.12E+02	7.55E+02	1.82E+04	1.36E+04	1.44E+04
F10	2.65E+02	1.20E+02	3.40E+02	3.81E+02	3.81E+02	3.19E+02	3.19E+02
F11	5.49E+01	6.09E+01	3.98E+01	5.59E+01	4.22E+01	5.99E+01	6.05E+01
F12	1.43E+06	1.32E+06	9.35E+04	2.53E+05	1.94E+06	1.59E+07	1.88E+06
F13	4.02E+02	2.69E+01	2.11E+01	6.37E+01	8.42E+02	9.24E+02	5.86E+02
F14	2.21E+01	2.26E+01	2.21E+01	2.22E+01	2.22E+01	2.23E+01	2.25E+01
F15	7.35E+02	7.93E+02	4.06E+02	5.68E+02	9.38E+02	8.81E+02	8.97E+02
F16	5.85E+02	6.16E+02	3.75E+02	5.49E+02	6.28E+02	6.06E+02	6.32E+02
F17	6.98E+02	6.43E+02	3.57E+02	5.77E+02	7.85E+02	7.19E+02	7.17E+02
F18	1.18E+03	9.00E+02	9.61E+02	1.11E+03	1.22E+03	1.21E+03	1.20E+03
F19	1.17E+03	9.00E+02	9.60E+02	1.11E+03	1.24E+03	1.23E+03	1.22E+03
F20	1.16E+03	9.00E+02	9.50E+02	1.20E+03	1.21E+03	1.20E+03	9.50E+02
F21	1.33E+03	1.36E+03	7.80E+01	1.18E+03	1.35E+03	1.32E+03	1.34E+03
F22	1.22E+03	1.35E+03	1.21E+03	1.32E+03	1.37E+03	1.33E+03	1.36E+03
F23	1.27E+03	1.36E+03	9.46E+02	1.27E+03	1.34E+03	1.32E+03	1.34E+03
F24	1.39E+03	1.39E+03	1.30E+03	1.40E+03	1.40E+03	1.40E+03	1.39E+03
F25	1.70E+03	1.72E+03	1.73E+03	1.68E+03	1.80E+03	1.80E+03	1.80E+03

Tabla 3. Valores obtenidos para cada una de las funciones con los modelos VMO-Operadores de cruce con dimensión 50 en la etapa de búsqueda local

DataSet	VMO-Op1	VMO-Op2	VMO-Op3	VMO-Op4	VMO-Op5	VMO-Op6	VMO-Op7
F6	3.93E+09	4.66E+09	4.26E+09	4.70E+09	2.28E+08	3.51E+09	1.08E+09
F7	3.22E+03	3.20E+03	3.13E+03	3.27E+03	3.21E+03	3.15E+03	3.12E+03
F8	2.03E+01	2.03+01	2.03E+01	2.10E+01	2.02E+01	2.04E+01	2.05E+01
F9	7.86E+02	1.35E+03	7.81E+02	1.15E+03	8.65E+02	1.09E+03	8.03E+02
F10	3.22E+02	3.33E+02	2.98E+02	3.64E+02	3.57E+02	3.01E+02	3.16E+02
F11	2.73E+01	3.20E+01	2.72E+01	3.32E+01	3.22E+01	3.13E+01	2.53E+01
F12	3.19E+05	2.59E+05	2.00E+05	3.37E+05	2.29E+05	2.05E+05	3.65E+05
F13	1.54E+02	1.88E+02	9.24E+01	1.58E+02	1.24E+02	2.25E+02	1.44E+02
F14	1.23E+01	1.29E+01	1.23E+01	1.21E+01	1.23E+01	1.30E+01	1.29E+01
F15	7.08E+02	6.86E+02	6.20E+02	7.34E+02	7.10E+02	7.75E+02	5.57E+02
F16	4.52E+02	5.84E+02	4.95E+02	3.33E+02	4.10E+02	4.30E+02	5.30E+02
F17	4.28E+02	4.50E+02	4.14E+02	5.01E+02	5.23E+02	5.37E+02	5.70E+02
F18	1.06E+03	1.09E+03	1.04E+03	1.10E+03	1.02E+03	1.15E+03	1.04E+03
F19	1.09E+03	1.10E+03	1.07E+03	1.11E+03	1.03E+03	1.09E+03	1.08E+03
F20	1.10E+03	1.09E+03	1.12E+03	1.07E+03	1.09E+03	1.06E+03	1.09E+03
F21	1.25E+03	1.20E+03	1.22E+03	1.25E+03	1.20E+03	1.25E+03	1.23E+03
F22	1.13E+03	1.27E+03	1.13E+03	1.15E+03	1.13E+03	1.22E+03	1.09E+03
F23	1.16E+03	1.19E+03	1.13E+03	1.21E+03	1.15E+03	1.25E+03	1.21E+03
F24	1.24E+03	1.31E+03	1.23E+03	1.33E+03	1.11E+03	1.33E+03	1.30E+03
F25	1.72E+03	1.73E+03	1.73E+03	1.71E+03	1.73E+03	1.71E+03	1.72E+03

Tabla 4. Valores obtenidos para cada una de las funciones con los modelos VMO-Operadores de cruce con dimensión 10 en la etapa de búsqueda global.

DataSet	VMO-Op1	VMO-Op2	VMO-Op3	VMO-Op4	VMO-Op5	VMO-Op6	VMO-Op7
F6	5.09E+03	4.52E+02	2.32E+02	1.29E+04	2.69E+03	3.00E+02	7.05E+02
F7	6.62E+02	6.56E+02	6.45E+02	6.58E+02	6.54E+02	6.48E+02	6.46E+02
F8	2.04E+01	2.03E+01	2.03E+01	.04E+01	2.01E+01	2.01E+01	2.03E+01
F9	1.87E+01	4.16E+01	3.58E+01	5.82E+01	3.02E+01	2.18E+01	3.53E+01
F10	2.41E+01	1.20E+01	6.58E+00	3.23E+01	2.11E+01	2.75E+01	2.21E+01
F11	5.36E+00	5.13E+00	4.39E+00	5.32E+00	5.14E+00	4.89E+00	5.42E+00
F12	2.93E+03	1.47E+02	2.93E+03	8.59E+03	2.18E+03	1.19E+03	8.39E+03
F13	1.62E+00	2.39E+00	1.29E+00	1.76E+00	1.38E+00	4.81E+00	3.41E+00
F14	2.93E+00	2.65E+00	2.26E+00	3.03E+00	3.10E+00	2.23E+00	3.25E+00
F15	2.15E+02	2.46E+02	1.07E+02	3.08E+02	3.43E+02	2.55E+02	2.39E+02
F16	1.48E+02	1.28E+02	1.43E+02	1.55E+02	1.23E+02	1.36E+02	1.44E+02
F17	1.57E+02	1.80E+02	1.76E+02	1.78E+02	1.41E+02	1.75E+02	1.65E+02
F18	9.36E+02	8.02E+02	9.06E+02	1.00E+03	9.43E+02	9.76E+02	1.00E+03
F19	9.88E+02	9.99E+02	9.75E+02	1.02E+03	9.94E+02	9.16E+02	9.79E+02
F20	9.61E+02	9.19E+02	9.92E+02	1.03E+03	9.93E+02	9.74E+02	1.01E+03
F21	8.95E+02	8.62E+02	7.89E+02	1.11E+03	1.19E+03	8.82E+02	9.08E+02
F22	8.60E+02	7.59E+02	7.58E+02	8.33E+02	7.90E+02	8.39E+02	8.98E+02
F23	7.56E+02	7.56E+02	6.54E+02	1.04E+03	9.57E+02	8.34E+02	7.56E+02
F24	4.65E+02	6.98E+02	8.48E+02	8.71E+02	6.86E+02	5.22E+02	4.65E+02
F25	1.87E+03	1.88E+03	1.88E+03	1.86E+03	1.88E+03	1.87E+03	1.87E+03

Tabla 5. Valores obtenidos para cada una de las funciones con los modelos VMO-Operadores de cruce con dimensión 30 en la etapa de búsqueda global.

DataSet	VMO-Op1	VMO-Op2	VMO-Op3	VMO-Op4	VMO-Op5	VMO-Op6	VMO-Op6
F6	1.85E+08	2.25E+09	2.02E+91	1.83E+10	3.22E+09	1.85E+09	1.98E+08
F7	4.75E+03	4.65E+03	4.75E+03	4.68E+03	4.66E+03	4.69E+03	4.58E+03
F8	2.13E+01	2.14E+01	2.14E+01	2.11E+01	2.13E+01	2.10E+01	2.18E+01
F9	1.12E+04	2.59E+04	1.16E+04	1.62E+04	2.44E+04	1.16E+04	1.86E+04
F10	8.89E+02	7.64E+02	8.89E+02	9.39E+02	8.37E+02	8.77E+02	8.76E+02
F11	6.43E+01	6.14E+01	6.47E+01	6.18E+01	6.32E+01	6.67E+01	6.35E+01
F12	1.58E+06	1.95E+06	1.75E+06	1.68E+06	1.55E+06	2.05E+06	2.38E+06
F13	8.94E+02	8.10E+02	5.76E+02	9.63E+02	9.04E+02	8.70E+02	8.44E+02
F14	2.21E+01	2.27E+01	2.24E+01	2.24E+01	2.17E+01	2.24E+01	2.20E+01
F15	9.87E+02	8.11E+02	9.67E+02	8.58E+02	8.49E+02	1.02E+03	9.75E+02
F16	7.90E+02	5.80E+02	7.54E+02	6.99E+02	6.70E+02	7.37E+02	5.21E+02
F17	7.36E+02	7.95E+02	7.22E+02	1.24E+03	6.56E+02	6.96E+02	6.54E+02
F18	1.17E+03	1.24E+03	1.19E+03	1.20E+03	1.23E+03	1.20E+03	1.22E+03
F19	1.20E+03	1.20E+03	1.11E+03	1.20E+03	1.16E+03	1.20E+03	1.20E+03
F20	1.20E+03	1.24E+03	1.22E+03	1.22E+03	1.23E+03	1.23E+03	1.22E+03
F21	1.35E+03	1.30E+03	1.33E+03	1.33E+03	1.39E+03	1.33E+03	1.33E+03
F22	1.33E+03	1.31E+03	1.41E+03	1.31E+03	1.30E+03	1.30E+03	1.36E+03
F23	1.29E+03	1.35E+03	1.35E+03	1.35E+03	1.37E+03	1.34E+03	1.35E+03
F24	1.40E+03	1.39E+03	1.41E+03	1.39E+03	1.40E+03	1.42E+03	1.41E+03
F25	1.81E+03	1.80E+03	1.79E+03	1.80E+03	1.80E+03	1.81E+03	1.79E+03

Tabla 6. Valores obtenidos para cada una de las funciones con los modelos VMO-Operadores Genéticos con dimensión 50 en la etapa de búsqueda global.

DataSet VMO-HíbridoLocal1	VMO-HíbridoLocal2	VMO-HíbridoLocal3
F6	6.69E+02	1.00E+03
F7	6.51E+02	6.46E+02
F8	2.02E+01	2.04E+01
F9	4.78E+01	4.77E+01
F10	2.32E+01	3.04E+01
F11	5.28E+00	6.63E+00
F12	6.06E+02	3.23E+03
F13	3.86E+00	3.08E+00
F14	4.01E+00	3.55E+00
F15	4.92E+02	3.61E+02
F16	3.72E+02	1.92E+02
F17	1.55E+02	1.96E+02
F18	8.93E+02	1.03E+03
F19	8.06E+02	1.03E+03
F20	9.65E+02	1.04E+03
F21	1.21E+03	1.04E+03
F22	8.71E+02	8.31E+02
F23	1.15E+03	7.93E+02
F24	1.15E+03	8.86E+02
F25	1.90E+03	1.88E+03

Tabla 7. Valores obtenidos para cada una de las funciones con los modelos VMO-Híbridos con dimensión 10 en la búsqueda local.

DataSet	VMO-HíbridoLocal4	VMO-HíbridoLocal5	VMO-HíbridoLocal6
F6	2.59E+09	2.21E+09	2.28E+08
F7	3.19E+03	3.29E+03	3.20E+03
F8	2.10E+01	2.10E+01	2.02E+01
F9	1.13E+03	1.24E+03	1.01E+03
F10	3.92E+02	3.81E+02	3.40E+02
F11	3.92E+02	3.13E+01	3.46E+02
F12	1.94E+05	1.57E+05	2.57E+05
F13	1.74E+02	1.79E+02	1.04E+02
F14	1.23E+01	1.26E+01	1.29E+01
F15	7.71E+02	7.13E+02	6.40E+02
F16	4.64E+02	5.76E+02	4.04E+02
F17	5.10E+02	5.44E+02	4.39E+02
F18	1.08E+03	1.10E+03	1.12E+03
F19	1.07E+03	1.10E+03	1.04E+03
F20	1.09E+03	1.06E+03	1.12E+03
F21	1.18E+03	1.23E+03	1.25E+03
F22	1.16E+03	1.16E+03	1.19E+03
F23	1.25E+03	1.22E+03	1.16E+03
F24	1.30E+03	1.26E+03	1.31E+03
F25	1.73E+03	1.71E+03	1.72E+03

Tabla 8. Valores obtenidos para cada una de las funciones con los modelos VMO-Híbridos con dimensión 30 en la búsqueda local.

DataSet	VMO-HíbridoLocal7	VMO-HíbridoLocal8	VMO-HíbridoLocal9
F6	2.85E+10	2.35E+10	1.30E+10
F7	4.64E+03	4.63E+03	4.62E+03
F8	2.11E+01	2.11E+01	2.12E+01
F9	1.73E+04	1.41E+04	1.71E+04
F10	8.64E+02	8.94E+02	8.96E+02
F11	5.82E+01	6.14E+01	5.81E+01
F12	1.82E+06	1.85E+06	2.12E+06
F13	7.03E+02	9.80E+02	9.83E+02
F14	2.24E+01	2.26E+01	2.24E+01
F15	8.50E+02	8.18E+02	8.94E+02
F16	5.49E+02	6.39E+02	7.39E+02
F17	6.45E+02	7.94E+02	8.01E+02
F18	1.21E+03	1.22E+03	1.19E+03
F19	1.20E+03	1.21E+03	1.18E+03
F20	1.21E+03	1.26E+03	1.22E+03
F21	1.38E+03	1.37E+03	1.32E+03
F22	1.36E+03	1.32E+03	1.30E+03
F23	1.31E+03	1.34E+03	1.26E+03
F24	1.38E+03	1.39E+03	1.40E+03
F25	2.85E+10	2.35E+10	1.80E+03

Tabla 9. Valores obtenidos para cada una de las funciones con los modelos VMO-Híbridos con dimensión 50 en la búsqueda local.

DataSet	VMO-HíbridoGlobal1	VMO-Híbrido Global2	VMO-HíbridoGlobal3
F6	1.99E+04	1.00E+04	1.09E+04
F7	6.54E+02	6.55E+02	6.49E+02
F8	2.03E+01	2.04E+01	2.05E+01
F9	5.98E+01	4.36E+01	2.78E+01
F10	2.04E+01	2.52E+01	2.98E+01
F11	5.73E+00	6.33E+00	5.82E+00
F12	2.94E+03	9.86E+03	3.77E+03
F13	2.75E+00	3.88E+00	2.84E+00
F14	2.80E+00	3.04E+00	3.40E+00
F15	4.95E+02	3.71E+02	2.71E+02
F16	1.80E+02	1.57E+02	1.76E+02
F17	1.88E+02	1.46E+02	1.74E+02
F18	9.88E+02	9.77E+02	9.68E+02
F19	9.33E+02	1.05E+03	9.82E+02
F20	9.63E+02	9.81E+02	9.69E+02
F21	1.05E+03	1.06E+03	1.15E+03
F22	8.20E+02	7.94E+02	8.46E+02
F23	1.09E+03	1.18E+03	1.12E+03
F24	7.09E+02	6.45E+02	9.10E+02
F25	1.88E+03	1.86E+03	1.86E+03

Tabla 10. Valores obtenidos para cada una de las funciones con los modelos VMO-Híbridos con dimensión 10 en la búsqueda global.

DataSet	VMO-HíbridoGlobal4	VMO-HíbridoGlobal5	VMO-HíbridoGlobal6
F6	4.29E+09	3.01E+09	4.06E+09
F7	3.20E+03	3.17E+03	3.15E+03
F8	2.09E+01	2.08E+01	2.10E+01
F9	2.53E+03	1.02E+03	1.06E+03
F10	3.49E+02	3.76E+02	3.74E+02
F11	2.98E+01	3.21E+01	2.98E+01
F12	2.76E+05	2.54E+05	2.43E+05
F13	1.56E+02	1.06E+02	2.12E+02
F14	1.27E+01	1.24E+01	1.30E+01
F15	7.49E+02	7.87E+02	7.46E+02
F16	4.18E+02	4.50E+02	4.74E+02
F17	4.19E+02	5.47E+02	5.00E+02
F18	1.10E+03	1.11E+03	1.10E+03
F19	1.11E+03	1.07E+03	1.08E+03
F20	1.02E+03	1.08E+03	1.10E+03
F21	1.26E+03	1.25E+03	1.25E+03
F22	1.24E+03	1.14E+03	1.14E+03
F23	1.28E+03	1.16E+03	1.23E+03
F24	1.30E+03	1.31E+03	1.31E+03
F25	1.72E+03	1.73E+03	1.73E+03

Tabla 11. Valores obtenidos para cada una de las funciones con los modelos VMO-Híbridos con dimensión 30 en la búsqueda global.

DataSet	VMO-HíbridoGlobal7	VMO-HíbridoGlobal8	VMO-HíbridoGlobal9
F6	2.64E+10	1.66E+10	2.22E+10
F7	4.65E+03	4.67E+03	4.42E+03
F8	2.11E+01	2.12E+01	2.11E+01
F9	1.49E+04	1.41E+03	2.09E+04
F10	8.76E+02	8.64E+02	8.12E+02
F11	6.05E+01	5.69E+01	5.92E+01
F12	1.53E+06	1.97E+06	1.62E+06
F13	7.08E+02	8.96E+02	8.58E+02
F14	2.21E+01	2.25E+01	2.20E+01
F15	8.54E+02	8.89E+02	9.01E+02
F16	5.46E+02	6.89E+02	7.16E+02
F17	7.14E+02	5.37E+02	6.50E+02
F18	1.21E+03	1.28E+03	1.21E+03
F19	1.22E+03	1.20E+03	1.24E+03
F20	1.26E+03	1.26E+03	1.21E+03
F21	1.32E+03	1.33E+03	1.31E+03
F22	1.34E+03	1.34E+03	1.35E+03
F23	1.29E+03	1.27E+03	1.29E+03
F24	1.39E+03	1.38E+03	1.39E+03
F25	1.80E+03	1.80E+03	1.80E+03

Tabla 12. Valores obtenidos para cada una de las funciones con los modelos VMO-Híbridos con dimensión 50 en la búsqueda global.

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