The background of the slide features a stylized illustration of business professionals in suits and dresses walking across a platform. Various icons are scattered around them, including a lightbulb, puzzle pieces, a flag, a magnifying glass, a gear, and a handshake. The text is centered within a semi-transparent white box.

**Desarrollo de metaheurísticas
poblacionales para la solución
de problemas complejos**

Amilkar Yudier Puris Cáceres
Pavel Novoa Hernández
Byron Oviedo Bayas

Desarrollo de metaheurísticas poblacionales para la solución de problemas complejos

Desarrollo de metaheurísticas poblacionales para la solución de problemas complejos

**Amilkar Yudier Puris Cáceres
Pavel Novoa Hernández
Byron Oviedo Bayas**

Título original: Desarrollo de metaheurísticas
poblacionales para la solución
de problemas complejos

© Amilkar Yudier Puris Cáceres
Pavel Novoa Hernández
Byron Oviedo Bayas
2020,

Publicado por acuerdo con los autores.
© 2020, Editorial Grupo Compás
Universidad Técnica Estatal de Quevedo
Guayaquil-Ecuador

Grupo Compás apoya la protección del copyright, cada uno de sus
textos han sido sometido a un proceso de evaluación por pares
externos con base en la normativa del editorial.

El copyright estimula la creatividad, defiende la diversidad en el
ámbito de las ideas y el conocimiento, promueve la libre expresión y
favorece una cultura viva. Quedan rigurosamente prohibidas, bajo las
sanciones en las leyes, la producción o almacenamiento total o
parcial de la presente publicación, incluyendo el diseño de la
portada, así como la transmisión de la misma por cualquiera de sus
medios, tanto si es electrónico, como químico, mecánico, óptico, de
grabación o bien de fotocopia, sin la autorización de los titulares del
copyright.

Editado en Guayaquil - Ecuador

ISBN: 978-9942-33-269-1

Cita.

Puris. A, Novoa. P, Oviedo. B. (2020) Desarrollo de metaheurísticas poblacionales para la solución de problemas complejos, Editorial Compás, Guayaquil Ecuador, 136 pag

CONTENIDO

PRÓLOGO	IV
INTRODUCCIÓN	5
CAPITULO 1: METAHEURÍSTICAS POBLACIONALES	12
ALGORITMOS EVOLUTIVOS.....	13
<i>Diversidad Poblacional</i>	<i>15</i>
<i>Recombinación</i>	<i>16</i>
<i>Criterios de Selección.....</i>	<i>17</i>
<i>Principales Algoritmos Evolutivos.....</i>	<i>18</i>
INTELIGENCIA COLECTIVA.....	19
<i>Caracterización de la Inteligencia Colectiva</i>	<i>20</i>
<i>Optimización Basada en Colonia de Hormigas.....</i>	<i>21</i>
TÉCNICAS ESTADÍSTICAS UTILIZADAS PARA LA VALIDACIÓN DE LOS RESULTADOS.....	32
CONSIDERACIONES FINALES	34
CAPÍTULO 2: OPTIMIZACIÓN BASADA EN COLONIA DE HORMIGAS EN DOS ETAPAS	36
FUNCIONAMIENTO GENERAL DE LA OPTIMIZACIÓN BASADA EN COLONIA DE HORMIGAS EN DOS ETAPAS	37
DESCRIPCIÓN DE CADA ETAPA	39
ANÁLISIS DE COMPLEJIDAD	44
ESTUDIO EXPERIMENTAL.....	46
<i>Selección del mejor algoritmo basado en colonia de hormigas</i>	<i>47</i>
<i>Estudio experimental de la exploración en dos etapas</i>	<i>56</i>
<i>Estudio comparativo entre ambas estrategias de exploración</i>	<i>62</i>
CONCLUSIONES PARCIALES.....	63
CAPÍTULO 3: OPTIMIZACIÓN BASADA EN MALLAS DINÁMICAS	65
DESCRIPCIÓN GENERAL DE LA METAHEURÍSTICA.....	65
FOMENTO DE LA DIVERSIDAD.....	69
CARACTERIZACIÓN Y NOVEDADES DEL MÉTODO	71
CASO DE ESTUDIO: APROXIMACIÓN DE FUNCIONES CONTINUAS.....	74
<i>Análisis interno del método.....</i>	<i>80</i>
<i>Estudio comparativo con otros métodos.....</i>	<i>86</i>

CONCLUSIONES PARCIALES	89
CAPÍTULO 4: SOLUCIÓN AL PROBLEMA DE SELECCIÓN Y ESCALAMIENTO DE MOVIMIENTOS SÍSMICOS FUERTES	91
DESCRIPCIÓN DEL DOMINIO DE APLICACIÓN	91
PROBLEMA DE SELECCIÓN Y ESCALAMIENTO DE MOVIMIENTO SÍSMICOS FUERTES.....	94
<i>Proceso de Selección de registros de sismos fuertes</i>	94
<i>Proceso de Escalado</i>	98
ESTUDIO EXPERIMENTAL.....	99
<i>Aplicación de la Optimización basada en Colonia de Hormigas en una y dos etapas</i>	99
<i>Aplicación de la Optimización basada en Mallas Dinámicas</i>	101
<i>Algoritmo Voraz basado en la Búsqueda de los Vecinos más Cercanos</i>	103
<i>Estructura de los datos utilizados</i>	103
<i>Análisis de los resultados obtenidos por los métodos presentados</i>	106
CONCLUSIONES PARCIALES	108
CONCLUSIONES	110
REFERENCIAS BIBLIOGRÁFICAS	112
PRODUCCIÓN CIENTÍFICA DEL AUTOR.....	128

PRÓLOGO

Las metaheurísticas poblacionales representan en el campo de la Inteligencia Artificial una poderosa herramienta para la solución de problemas complejos en distintos dominios de aplicación. Estudios revelan que no existe la mejor de las metaheurísticas para solucionar cualquier problema complejo, por lo que resulta imprescindible encontrar nuevas alternativas de búsqueda en el ámbito de las metaheurísticas poblacionales que mejoren el comportamiento actual en términos de eficiencia y eficacia. El objetivo de esta investigación es desarrollar nuevos métodos de solución de problemas de optimización; los cuales, basados en el enfoque poblacional, doten a los investigadores en esta rama de otras alternativas para enfrentar los problemas existentes. Los resultados de esta investigación son: la división en dos etapas de la estrategia de exploración de los algoritmos basados en colonia de hormigas que garantiza mayor eficiencia y eficacia de la misma; la definición de una metaheurística poblacional con características evolutivas, donde se introducen nuevas formas para aumentar la exploración y la intensificación de las soluciones en dominios tanto discretos como continuos; y la aplicación de los métodos propuestos en la solución del problema de selección y escalamiento de movimientos sísmicos fuertes para brindarle a los ingenieros estructurales una ayuda computacional en el estudio de la resistencia de las estructuras en Cuba, problema real de gran importancia en el área de la Sismología.

INTRODUCCIÓN

En el lenguaje coloquial, optimizar significa poco más que mejorar; sin embargo, en el contexto científico la optimización es el proceso de tratar de encontrar la mejor solución posible para un determinado problema. En un problema de optimización existen diferentes soluciones y un criterio para discriminar entre ellas. De forma más precisa, estos problemas se pueden expresar como: encontrar el valor de las variables de decisión para los que una determinada función objetivo alcanza su valor máximo o mínimo. El valor de las variables en ocasiones está sujeto a determinadas restricciones.

Se pueden encontrar una gran cantidad de problemas de optimización, tanto en la industria como en la ciencia. Desde los clásicos problemas de diseño de redes de telecomunicación u organización de la producción hasta los más actuales en ingeniería y reingeniería de software, existe una infinidad de problemas teóricos y prácticos que involucran a la optimización. Algunas clases de problemas de optimización son relativamente fáciles de resolver. Este es el caso, por ejemplo, de los problemas lineales, en los que tanto la función objetivo como las restricciones son expresiones lineales. Estos problemas pueden ser resueltos con el conocido método Simplex [1] introducido en su forma original en 1945 por George Dantzig. Otros métodos clásicos de Análisis Matemático y de Investigación de Operaciones permiten resolver problemas no necesariamente lineales, e incluso discretos, en forma exacta o aproximada con técnicas que determinan el óptimo barriendo un espacio de soluciones o sus propiedades. Sin embargo, existen muchos problemas que por su propia naturaleza, no pueden ser resueltos

mediante un procedimiento de búsqueda exhaustivo¹ o completo [2], ya sea porque tal método no exista o porque su complejidad crece de manera no polinomial. De hecho, la mayor parte de los que se encuentran en la práctica pertenecen a este tipo de problemas, conocidos en Investigación de Operaciones como problemas complejos.

Todo esto ha impulsado al desarrollo de procedimientos eficientes para encontrar soluciones aproximadas², donde la rapidez del proceso es tan importante como la calidad de estas. Estos procedimientos son conocidos como métodos heurísticos. El término fue introducido en el siglo pasado por dos de los precursores de la Inteligencia Artificial. Polya en 1957 [3] en su obra "How to Solve It" utiliza el término por primera vez. Años más tarde, Simon [4] define heurístico como un proceso que puede resolver un problema dado, pero no ofrece garantía de hacerlo.

Más recientemente en [5] se recogen varias definiciones diferentes de algoritmo heurístico, entre las que está:

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

En los últimos años ha habido un crecimiento espectacular en el desarrollo de procedimientos heurísticos para resolver problemas de optimización. Este hecho queda claramente reflejado en la creación de revistas especializadas para la difusión de este tipo de procedimientos;

¹ Métodos que obtienen la mejor solución, tras una búsqueda en todo el dominio de posibles valores.

² Aquí se utiliza el término, no como aproximaciones numéricas sino como el conjunto de soluciones de buena calidad pero no necesariamente las óptimas.

como es el caso de la revista Journal of Heuristics³ editada por primera vez en el año 1995.

Al abordar el estudio de los algoritmos heurísticos [6] se puede comprobar que éstos dependen en gran medida del problema en concreto para el que se han diseñado. De manera que no se puede asegurar que existe el método heurístico que sea capaz de obtener los mejores resultados para cualquier problema de optimización, resultado conocido como Teorema "No Free Lunch" [7].

En los últimos años ha aparecido un conjunto de métodos bajo el nombre de metaheurísticos [8] 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 [9]. Los autores Glover y Kochenberger introducen la siguiente definición [10]:

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.

³ Accesible desde <http://www.springer.com/math/applications/journal/10732>

Ante la proliferación de los procedimientos metaheurísticos, y con el objetivo de esclarecer el amplio rango de posibilidades para abordar un problema de optimización, se han realizado diversas propuestas de clasificación de los mismos. Estas clasificaciones están condicionadas por características que presentan los métodos, dando lugar a biparticiones del conjunto de metaheurísticas. En [11] se realiza una clasificación de las metaheurísticas en dependencia de la cantidad de soluciones que se mantienen en una misma iteración. Esta clasificación da origen a los siguientes grupos:

- *metaheurísticas de trayectoria simple*: se utiliza el término de trayectoria simple porque el proceso de búsqueda que desarrollan estos métodos se caracteriza por una trayectoria en el espacio de soluciones; es decir, que partiendo de una solución inicial, son capaces de generar un camino o trayectoria en el espacio de búsqueda a través de operaciones de movimiento. En este tipo de metaheurísticas se destacan: la Búsqueda Tabú (Taboo Search) [12], Recocido Simulado (Simulated Annealing) [13], Búsqueda de Vecindades Variables (Variable Neighborhood Search) [14], Búsqueda Local Guiada (Guided Local Search) [15], Búsqueda Local Iterativa (Iterated Local Search) [16], entre otras.
- *metaheurísticas poblacionales*: Las metaheurísticas basadas en población, o metaheurísticas poblacionales, son aquellas que emplean un conjunto de soluciones (población) en cada iteración del algoritmo, en lugar de utilizar una única solución como las metaheurísticas del grupo anterior. Estas proporcionan de forma intrínseca un mecanismo de exploración paralelo del espacio de soluciones, y su eficacia depende en gran medida de cómo se manipule dicha población. Dentro de esta clasificación se

destacan los Algoritmos Evolutivos (Evolutionary Algorithms; EA) [17, 18] y los algoritmos basados en Inteligencia Colectiva (Swarm Intelligence; SI) [19, 20].

Estas metaheurísticas poblacionales son de las más estudiadas y comparten como característica fundamental que han sido inspiradas en algún proceso natural. Los EA, fueron inspirados por la teoría de la evolución de Darwin [21]. Según su teoría, la evolución de las especies se produce por tres conceptos: replicación, variación y selección natural. Un ejemplo clásico de este tipo de algoritmos son los Algoritmos Genéticos (Genetic Algorithms; GA) [22]. Por otra parte, los SI toman su inspiración en ejemplos biológicos de comportamiento colectivo (enjambre) como es el caso de las colonias de insectos, las bandadas de aves y los cardúmenes de peces. Dentro de estos algoritmos se encuentran; la Optimización Basada en Enjambre de Partículas (Particle Swarm Optimization; PSO) [23], y Optimización Basada en Colonia de Hormigas (Ant Colony Optimization; ACO) [24].

En la actualidad existe un gran interés por el estudio de las metaheurísticas poblacionales, donde se destacan internacionalmente los laboratorios de Soft Computing and Intelligent Information Systems (SCI2S)⁴ y el Institut de Recherches Interdisciplinaires et de Développements en Intelligence Artificielle (IRIDIA)⁵, de la Universidad de Granada, en España, y la Universidad Libre de Bruselas, en Bélgica respectivamente. En Cuba también se realizan investigaciones reconocidas en este tema, como es el caso del grupo de Optimización

⁴ Accesible desde <http://sci2s.ugr.es/presentation/index.php>

⁵ Accesible desde <http://code.ulb.ac.be/iridia.home.php>

en el Instituto de Cibernética, Matemática y Física (ICIMAF) ⁶, el Centro de Estudios de Ingeniería de Sistemas del ISPJAE⁷, el grupo de Optimización de la UH⁸ y el Laboratorio de Inteligencia Artificial de la UCLV⁹. La mayoría de los estudios, están encaminados a mejorar el comportamiento de los métodos existentes, ya sea perfeccionándole alguna característica fundamental o simplemente incorporándole nuevas ideas de otras metaheurísticas, a manera de métodos híbridos [25]. Además, se evidencia un gran interés por aplicar estos métodos exitosos experimentalmente, a problemas reales con perfil económico-social dada la exigente demanda de obtener soluciones más precisas y en el menor tiempo posible (eficacia y eficiencia).

En consecuencia, con lo antes planteado, el laboratorio de Inteligencia Artificial de la UCLV recibe la solicitud por parte de un grupo de Ingenieros Estructurales de estudiar el nivel de resistencia de las estructuras (puentes, edificaciones, etc.) en nuestras condiciones de suelo activo. Para ello, se hacía necesario realizar una selección de un conjunto de terremotos ocurridos en otros escenarios, que tengan mayor relación con nuestras condiciones de suelo, representada por el espectro objetivo de la Norma Cubana NC-46-1999. Otro elemento importante es obtener los valores de escalamiento adecuados teniendo en cuenta la variabilidad de la señal sísmica y sus efectos en la sensibilidad de las respuestas de las estructuras. Este problema se nombra "Selección y Escalamiento de movimientos sísmicos fuertes" (PSE).

⁶ Accesible desde http://www.icmf.inf.cu/grupos4_matematicalInterd.htm

⁷ Instituto Superior Politécnico Jose Antonio Echeverría, accesible desde <http://www.cujae.edu.cu/centros/ceis/>

⁸ Universidad de la Habana, accesible desde <http://www.matcomp.uh.cu>

⁹ Universidad Central "Marta Abreu" de las Villas, accesible desde <http://www.cei.uclv.edu.cu/>

Al no existir un estudio general del comportamiento de distintas metaheurísticas para el problema anterior, ni la mejor de ellas para resolver ciertos problemas complejos, se deriva el surge el reto de encontrar nuevas alternativas de búsqueda en el ámbito de las metaheurísticas poblacionales que permitan obtener soluciones con mayor eficiencia y eficacia que las propuestas existentes en el área de estudio, ampliando de esta forma el rango de posibilidades en la solución de problemas complejos, entre los que se encuentra la selección y escalamiento de movimientos sísmicos fuertes.

CAPITULO 1: METAHEURÍSTICAS POBLACIONALES

En el desarrollo de los procedimientos metaheurísticos es importante tomar en consideración la definición de los operadores que se ocupen de dirigir la búsqueda a zonas promisorias del espacio de solución. Para ello, se tienen que tener en cuenta dos factores importantes [26]:

- La *exploración*, también llamada diversificación en la literatura, es el proceso de guiar la búsqueda hacia regiones no exploradas. Un algoritmo que realice una insuficiente exploración podría obviar regiones enteras, por lo que si el óptimo se encontrase en una de esas regiones no tendría ninguna posibilidad de ser encontrado.
- La *explotación*, también llamada intensificación en la literatura, es el proceso de realizar una búsqueda minuciosa e intensa de mejores soluciones en un entorno cercano a buenas soluciones ya encontradas (considerando que son zonas más prometedoras). Este proceso es fundamental para obtener una mayor precisión en las soluciones encontradas; aunque, cuando caen en un óptimo local¹⁰, suelen quedarse estancados¹¹ en este.

Obtener métodos que utilicen una continua sinergia entre ambos factores es un elemento a tener en cuenta para conseguir soluciones más precisas. En particular, las metaheurísticas poblacionales suelen apostar por obtener una mayor exploración en la búsqueda. En muchos casos, este factor es regulado por parámetros introducidos a los métodos

¹⁰ Solución de mejor calidad en una vecindad determinada. Cuando la vecindad es todo el espacio de búsqueda esta solución se convierte en óptimo global.

¹¹ Cuando el método no es capaz de generar soluciones de mejor calidad que la del óptimo local alcanzado, común en los algoritmos que utilizan mucho la explotación como técnica de búsqueda.

o decisiones tomadas en presencia de números aleatorios. Para incrementar la explotación en estos algoritmos, los mismos se combinan con métodos de trayectoria simple (utilizados frecuentemente como algoritmos de Búsqueda Local (Local Search; LS) [27], los cuales presentan un alto nivel de explotación de la zona del espacio de búsqueda que ha sido explorada.

En este capítulo se realiza una descripción detallada de los principales métodos poblacionales, específicamente en el Epígrafe 1.1 se tratarán los Algoritmos Evolutivos, mientras que en el Epígrafe 1.2 se abordará la Inteligencia Colectiva. Seguidamente, en el Epígrafe 1.3 se presenta el marco experimental estadístico utilizado para validar los resultados experimentales obtenidos al estudiar los métodos propuestos. Y por último, en el Epígrafe 1.4 se presentan las consideraciones generales del capítulo.

Algoritmos Evolutivos

Los Algoritmos Evolutivos [17] fueron inspirados por la Teoría de la Evolución de Darwin [21]. Según su teoría, la evolución de las especies se produce por tres conceptos: replicación, variación y selección natural. La población en este tipo de algoritmos representa un conjunto de individuos¹², los cuales se mantienen mediante replicación. La variación es la encargada de introducir diversidad en la población, mediante una serie de diferencias entre los descendientes y sus progenitores¹³.

¹² En los Algoritmos Evolutivos esta estructura representa una solución al problema, también conocido como cromosoma en Algoritmos Genéticos.

¹³ Nuevos individuos generados mediante un mecanismo de recombinación sexual de dos individuos, en el que el nuevo individuo hereda características de ambos.

Existen diferentes tipos de EA: Los Algoritmos Genéticos [22], Programación Genética (Genetic Programming) [28], Estrategias de Evolución (Evolution Strategies) [29] y Programación Evolutiva (Evolutive Programming) [30]. Todos los EA se caracterizan por estar formados de una población inicial (usualmente generada aleatoriamente) que sufre un proceso iterativo que conduce a la evolución de dicha población (a semejanza de la natural), mediante los siguientes procesos:

1. Se aplica un operador de selección que determina la probabilidad de cada individuo de perdurar en la generación siguiente, construyéndose así una población temporal.
2. Se aplican operadores evolutivos (recombinación, mutación) a parte, o la totalidad, de la población temporal, con lo que se produce un conjunto de nuevas soluciones.
3. Se calcula el valor de la función a optimizar en las nuevas soluciones generadas.
4. Se obtiene una nueva población a partir de la población temporal y los nuevos individuos generados.

Dichos pasos son realizados hasta que se cumple un criterio de terminación (se alcanza un límite máximo temporal o un número máximo de evaluaciones de la función objetivo).

Este procedimiento de ir creando nuevos individuos a partir de los existentes en la población, y en el que los mejores sustituyen a los anteriores, es lo que se denomina evolución de la población. Construir los nuevos individuos combinando características de buenas soluciones encontradas hasta el momento (almacenadas en la población) es lo que permite orientar la búsqueda hacia las regiones más prometedoras.

Todos los EA comparten los mismos conceptos básicos, aunque difieren en el mecanismo empleado para codificar las soluciones y los operadores que emplean para producir la siguiente generación.

Diversidad Poblacional

En los Algoritmos Evolutivos un elemento importante para conseguir una profunda exploración del espacio de búsqueda es el uso de mecanismos que garanticen encontrar y mantener soluciones con diversas características, tanto para problemas discretos como continuos. A este elemento se le conoce como diversidad de una población y se manifiesta de forma directa o indirecta [25, 31, 32]. A continuación, se examinan algunos enfoques:

- En [33] se presentan los primeros esfuerzos por introducir diversidad, proponiéndose un mecanismo que mantiene en paralelo varias sub-poblaciones, procesadas cada una por un algoritmo genético.
- En [34, 35] se presentan Algoritmos Meméticos que abordan de forma muy eficiente problemas clásicos de optimización discreta (Viajante de Comercio, Asignación Cuadrática, entre otros). Sus buenos resultados son un modelo a seguir, y se apoyan en:
 - ✓ El uso de operadores de cruce y mutación específicos del problema.
 - ✓ Adaptar el esquema del EA empleado al problema considerado, incorporando mecanismos de reinicio en caso de estancamiento (perturbar la población).

-
- Seront y Bersini [36] presentan un EA con un método de agrupamiento que puede reducir el costo total de la búsqueda local, evitando redescubrir varias veces los mismos óptimos locales.
 - Mecanismo de prevención de incesto, introducido en el algoritmo generacional con selección elitista y recombinación heterogénea (Cross generational elitist selection, Heterogeneous recombination, and Cataclysmic mutation; CHC) [37], donde se cruzan solo aquellos individuos que presenten una determinada diversidad.

Recombinación

El operador de recombinación es un método para compartir información entre individuos, el cual, juega un papel central en los EA [38, 39] ya que explota la información disponible en ambas muestras para influir en futuras búsquedas. En [40] se presenta una taxonomía detallada para la clasificación de los tipos de operadores de combinación existentes en dependencia de cómo generan los descendientes. Específicamente se tienen cuatro grupos diferentes de operadores:

- *Operadores discretos*. Esta categoría reúne a todos los operadores de recombinación presentes para codificación binaria. Aunque fue desarrollada principalmente para problemas discretos, se pueden aplicar a representaciones en codificación real. Esto incluye por ejemplo a los cruzamientos en los Algoritmos Genéticos por un punto (simple) y dos puntos (doble), donde el valor de un gen en los descendientes coincide exactamente con el valor de este gen en uno de los padres.
- *Operadores basados en agregación*. Incluyen los operadores que usan funciones de agregación, las que combinan numéricamente

los valores de los genes de los padres para generar los genes de los descendientes. Como ejemplo de estos aparecen los operadores Aritmético (AX), Geométrico (GX) y Lineal (LX). Estos operadores son totalmente determinísticos.

- *Operadores basados en entornos.* Este grupo incluye los cruzamientos que determinan los genes en los descendientes extrayendo valores desde intervalos definidos por vecindades asociadas con los genes de los padres, a través de distribuciones probabilísticas. Ejemplos de estos operadores son: BLX-a [41], SBX- η [42], FR-d [43] y PNX- η [44], basados en distribuciones de probabilidad uniforme, polinomial, triangular y normal, respectivamente.
- *Operadores híbridos.* Estos operadores generan los descendientes mediante operadores de distintos grupos. Es el caso del operador Mínimo-Máximo (MMAX), el cual crea cuatro descendientes, dos mediante un Operador Discreto (máximo y mínimo) y dos mediante Agregación (aritmético).

Criterios de Selección

En los EA existen diversas formas de realizar la selección de la población temporal que servirá de base de la generación en la próxima iteración. Entre las más utilizadas están:

- *Selección elitista:* la población temporal está compuesta por los mejores¹⁴ individuos de la población actual. Este tipo de selección

¹⁴ En este contexto significa los individuos que tengan mejor evaluación de la función objetivo a optimizar.

puede llevar al método a un estancamiento de la población en óptimos locales.

- *Selección aleatoria*: es la encargada de obtener una población temporal de forma totalmente aleatoria. Este esquema reduce la explotación del espacio de búsqueda, ya que buenas soluciones pueden no formar parte de la población temporal que será la encargada de generar la nueva población.
- *Selección probabilística*: donde a cada solución se le asigna una probabilidad en dependencia de su calidad y mediante las generaciones aleatorias se puede determinar qué solución formará parte de la población temporal.

Principales Algoritmos Evolutivos

Dentro de los EA se destacan los Algoritmos Genéticos [22], los cuales se caracterizan porque el mecanismo empleado para generar nuevos individuos es mediante un operador de recombinación (denominado cruce), el cual combina información de diferentes individuos para producir los nuevos individuos. De manera general, los EA presentan la ventaja de ser algoritmos de búsqueda capaces de identificar regiones con buen rendimiento en dominios de búsqueda amplios y complejos. Debido al alto grado de exploración que realizan para la búsqueda, presentan problemas para obtener soluciones precisas [45].

Con el paso del tiempo otros métodos de EA han sido propuestos. A continuación, se describen algunos de los nuevos métodos surgidos y que se consideran de especial interés:

- *La búsqueda dispersa (Scatter Search)* [46]. En este método se mantienen dos sub-poblaciones, una élite, de las mejores

soluciones encontradas (para explotar las soluciones encontradas), y otra formada por individuos alejados de los individuos de la población anterior (para mantener cierta diversidad en el proceso de combinación). Además, es un método que considera el uso de un proceso de mejora local, como una búsqueda local, para explotar más las soluciones encontradas.

- *La evolución diferencial (Differential Evolution)* [47]. Este método se caracteriza por el hecho de que, dado que las soluciones se representan mediante vectores de valores reales, las poblaciones de soluciones son modificadas de forma sucesiva mediante operadores de suma y resta vectorial, así como operaciones de intercambios de valores, hasta que la población converge en el espacio de búsqueda.
- *Algoritmos basados en estimación de distribuciones (Estimation of Distribution Algorithms)* [48]. Se caracterizan por no utilizar ni los operadores de cruce ni los de mutación para generar nuevas soluciones. Se genera la nueva población mediante muestreo de una distribución de probabilidad, la cual es estimada a partir de los mejores individuos de las generaciones anteriores.
- *Algoritmos meméticos (Memetic Algorithms)* [49, 50]. Representan un método clásico híbrido, donde al poder de exploración de un EA se le incluye una o más fases de búsqueda local en el mismo ciclo evolutivo. De este modo se logra explorar y explotar el espacio de búsqueda al mismo tiempo.

Inteligencia Colectiva

Las metaheurísticas basadas en Inteligencia Colectiva son técnicas inspiradas en el estudio de comportamientos colectivos [19] presentes en sistemas de la naturaleza, generalmente de carácter descentralizado y auto organizativo. Dicho comportamiento social define los movimientos de las variables de decisión en el espacio de búsqueda y las orienta hacia soluciones óptimas. La expresión "Inteligencia Colectiva" fue introducida por Gerardo Beni, Suzanne Hackwood y Jing Wang en 1989, en el contexto de sistemas robóticos celulares.

Caracterización de la Inteligencia Colectiva

La principal característica de los algoritmos basados en la SI viene determinada por la estrecha colaboración social que presentan a través del sistema de comunicación que surge entre los individuos del colectivo [20]. Esta comunicación, a su vez, puede aparecer de forma directa o indirecta. La comunicación indirecta ocurre cuando un individuo altera el medio en que se desarrollan y los otros son capaces de captar estos cambios. La comunicación directa es aquella que ocurre a través de la obtención de la ubicación de otros individuos mediante sonido, visibilidad u otra forma directa de interacción. Entre los algoritmos más significativos y estudiados de SI están:

- *Optimización basada en Enjambre de Partículas* [23]. Se caracteriza por ser un método inspirado en el movimiento natural desarrollado por las comunidades de animales, tales como las emigraciones de las bandadas de pájaros. En estos, la población representa un enjambre y está compuesta por una serie de partículas (pájaros) que representan soluciones a un problema determinado, donde cada una de ellas realiza un desplazamiento (cambia de posición) en el espacio de búsqueda. Cada partícula

se desplaza influenciada por dos direcciones: una hacia la mejor partícula encontrada hasta el momento, y por la mejor posición por la que ha pasado. Este comportamiento es un ejemplo de comunicación directa.

- *Optimización basada en colonia de hormigas* (Ant Colony Optimization; ACO) [51]. Son métodos poblacionales inspirados en el comportamiento de las hormigas naturales. Los mismos realizan un proceso constructivo y estocástico guiado por unos rastros de feromona¹⁵ que van depositando cada hormiga, dando una medida de cuán deseado ha sido un determinado camino, y a través de una función de visibilidad que evalúa la calidad del desplazamiento. Es un ejemplo clásico de comunicación indirecta (conocido como Stigmergy [52]).

A continuación se presenta en detalles la metaheurística ACO, debido a la importancia que tienen para el presente trabajo.

Optimización Basada en Colonia de Hormigas

Los algoritmos de ACO [51, 53-55] se inspiran directamente en el comportamiento de las colonias reales de hormigas para solucionar problemas de optimización combinatoria. Se basan en una colonia de hormigas artificiales; esto es, unos agentes computacionales simples que trabajan de manera cooperativa y se comunican mediante rastros artificiales de feromona. Los algoritmos de ACO son esencialmente métodos constructivos: en cada iteración del algoritmo, cada hormiga construye una solución al problema recorriendo un grafo de construcción

¹⁵ Sustancia química olorosa que depositan las hormigas en su recorrido. La intensidad de esta sustancia disminuye a través de un proceso de evaporación que ocurre en el tiempo de manera constante.

G^{16} . Cada arista del grafo, que representa los posibles pasos que la hormiga puede dar, tiene asociados dos tipos de información que guían el movimiento de la hormiga:

- *Información heurística*, mide la preferencia heurística de moverse desde el nodo r hasta el nodo s ; es decir, la propensión a recorrer la arista a_{rs} . Se denota por η_{rs} . Las hormigas no modifican esta información durante la ejecución del algoritmo.
- *Información de los rastros artificiales de feromona*, mide la “deseabilidad aprendida”¹⁷ del movimiento de r a s . Imita de forma numérica a la feromona real que depositan las hormigas naturales. Esta información se modifica durante la ejecución del algoritmo dependiendo de las soluciones encontradas por las hormigas. Se denota por τ_{rs} .

Modo de funcionamiento general

El modo de operación básico de un algoritmo de ACO [51, 55, 56] es como sigue: las m hormigas (artificiales) de la colonia se mueven, concurrentemente y de manera asíncrona, a través de los estados adyacentes del problema (que puede representarse en forma de grafo con ponderaciones o sin ellas). Este movimiento se realiza siguiendo una regla de transición que está basada en la información local disponible en las componentes (nodos). Esta información local incluye la información heurística y memorística (rastros de feromona) para guiar la búsqueda.

¹⁶ Como se dice en [54], el conjunto de aristas puede conectar completamente los componentes. En este caso, la implementación de las restricciones está completamente integrada en la política de construcción de las hormigas.

¹⁷ Nótese que las hormigas solo se comunican de manera indirecta, a través de modificaciones del espacio físico que perciben. Esta forma de comunicación se denomina Estimergia Artificial.

Las hormigas construyen incrementalmente soluciones al moverse por el grafo de construcción. Opcionalmente, las hormigas pueden depositar feromona cada vez que crucen un arco (conexión) mientras que construyen la solución (*actualización en línea paso a paso de los rastros de feromona*). Una vez que cada hormiga ha generado una solución, ésta se evalúa y el agente puede depositar una cantidad de feromona en dependencia de la calidad de su solución (*actualización en línea de los rastros de feromona*). Esta información guiará la búsqueda de las otras hormigas de la colonia en el futuro. Además, el modo de operación genérico de un algoritmo de ACO incluye dos procedimientos adicionales, la evaporación de los rastros de feromona y las acciones del demonio. La evaporación de feromona la lleva a cabo el entorno y se usa como un mecanismo que evita el estancamiento en la búsqueda y permite que las hormigas busquen y exploren nuevas regiones del espacio. Las acciones del demonio son acciones opcionales (que no tienen un contrapunto natural) para implementar tareas desde una perspectiva global que no pueden llevar a cabo las hormigas por la perspectiva local que ofrecen. Ejemplos son: observar la calidad de todas las soluciones generadas y depositar una nueva cantidad de feromona adicional sólo en las transiciones/componentes asociadas a algunas soluciones, o aplicar un procedimiento de búsqueda local a las soluciones generadas por las hormigas antes de actualizar los rastros de feromona. En ambos casos el demonio reemplaza la actualización en línea a posteriori de feromona y el proceso pasa a llamarse *actualización fuera de línea de rastros de feromona*.

En ACO el significado de los rastros de feromona y la función heurística o de visibilidad dependen totalmente del problema a resolver. En el caso específico de los rastros de feromona, cuando se está en presencia de

un problema de secuenciación (Viajante de Comercio [57], Asignación Cuadrática [58], entre otros), donde el orden en que aparecen las componentes en una solución influye en la calidad de ésta, los rastros de feromona son asociados a los arcos del grafo, con el objetivo de premiar las buenas secuencias de componentes¹⁸. Por otra parte, en problemas de asignación (Selección de Rasgos [59], Partición de Conjuntos [60], entre otros) donde los cambios de posición entre componentes de una solución no influye en la calidad de la misma, los rastros de feromona son asociados a los nodos del grafo¹⁹.

La estructura general de ACO es como sigue [51]:

Procedimiento metaheurística ACO;

Actividades Programadas

 Construir Soluciones de las Hormigas

 Actualizar Feromonas

 Evaporación de la Feromona

 Acciones del Demonio (opcional)

Fin de las Actividades Programadas

Fin del Procedimiento

Ilustración 1: Procedimiento general de ACO

¹⁸ Mide la deseabilidad de la colonia por una determinada secuencia de nodos.

¹⁹ Mide la deseabilidad de la colonia por un estado en específico, no interesa de donde fue alcanzado.

Este procedimiento se anida en el siguiente procedimiento iterativo:

Paso1: Inicializar los valores de feromona

iteracionActual=1

Paso2: Repetir

Procedimiento metaheurística ACO

iteracionActual = iteracionActual +1

Hasta que: criterio de parada

Ilustración 2: Estructura genérica de ACO.

Para los métodos de ACO existen distintos criterios de parada [55], entre los que se encuentran:

- Se alcanza un número máximo de iteraciones o ciclos.
- Se obtiene una solución con una calidad deseada.
- Se alcanza un tiempo límite o predeterminado de procesamiento.
- Se obtiene un número máximo de evaluaciones de la función objetivo.

Tareas de diseño

Observando las aplicaciones actuales de ACO, se pueden identificar algunas directivas sobre cómo atacar problemas utilizando esta metaheurística. Estas directivas se pueden resumir en las seis tareas de diseño que se enumeran a continuación [56]:

-
1. Representar el problema como un conjunto de componentes (nodos) y transiciones (aristas) a través de un grafo que será recorrido por las hormigas para construir soluciones.
 2. Definir de manera apropiada en base a las características del problema, el significado de los rastros de feromona τ .
 3. Definir de manera apropiada la preferencia heurística o función de visibilidad η asociada a cada componente o transición.
 4. Si es posible, implementar una búsqueda local eficiente para mejorar las soluciones obtenidas por ACO.
 5. Escoger un algoritmo de ACO específico y aplicarlo al problema que hay que solucionar teniendo en cuenta las características propias de cada uno de estos algoritmos.
 6. Refinar los parámetros del algoritmo de ACO seleccionado.

Dentro de los algoritmos de ACO las diferencias fundamentales radican en la regla de transición de estados que utilizan para la construcción de las soluciones y en el tratamiento que les dan a los rastros de feromona. Debido a esto, aparecen en la literatura distintos algoritmos ACO.

Principales algoritmos basados en colonia de hormigas

Entre los algoritmos de ACO disponibles para problemas de optimización combinatoria [61] se encuentran: el Sistema de Hormigas (Ant System; AS) [62], el Sistema de Colonia de Hormigas (Ant Colony System; ACS) [63], el Sistema de Hormigas Máximo-Mínimo (Max-Min Ant System; MMAS) [64], el Sistema de Hormigas con Ordenación Jerárquica (Rank-Based Ant System) [65], el Sistema de Hormigas Mejor-Peor (Best-Worst Ant System) [66], entre otros. Esta técnica comienza a tener la madurez tecnológica

adecuada para su utilización en problemas reales, como puede verse en la publicación del libro [61].

A continuación se presenta una pequeña descripción de los algoritmos Sistema de Hormigas, Sistema de Colonia de Hormigas y el Sistema de Hormigas Máximo-Mínimo, debido a que fueron seleccionados para llevar a cabo este trabajo.

El **Sistema de Hormigas** [62, 67], desarrollado por Dorigo en su tesis doctoral en 1992 [24], fue el primer algoritmo de ACO. Su versión actual (Ant Cycle) apareció conjuntamente con otras variantes de éste, como el Sistema de Hormigas Densidad (Ant Density) y Sistema de Hormigas Calidad (Ant Quality). El AS se caracteriza por el hecho de que, la actualización de feromona se realiza una vez que todas las hormigas han completado sus soluciones, y se lleva a cabo como sigue: primero, todos los rastros de feromona se reducen en un factor constante, implementándose de esta manera la evaporación de feromona según la ecuación (0.1),

$$t_{ij} \leftarrow (1 - \rho) \cdot t_{ij}, \rho \in (0,1] \quad (0.1)$$

donde ρ se conoce como constante de evaporación y es la encargada de reducir los rastros de feromona para evitar el estancamiento de las soluciones y t_{ij} la cantidad de feromona asociada al arco α_{ij} .

A continuación, cada hormiga de la colonia deposita una cantidad de feromona en función de la calidad de su solución, según la ecuación (0.2),

$$t_{ij} \leftarrow t_{ij} + \Delta t^k \quad \forall a_{ij} \in S^k \quad (0.2)$$

donde $\Delta t^k = f(C(S^k))$, representa la cantidad de feromona a depositar por la hormiga k en cada arco a_{ij} de su solución encontrada (S^k). Este valor depende de la calidad de dicha solución ($C(S^k)$).

Las soluciones en el AS se construyen como sigue. En cada paso de construcción, una hormiga k escoge ir al siguiente nodo con una probabilidad que se calcula como:

$$p_{ij}^k = \frac{(\tau_{ij})^\alpha \cdot (\eta_{ij})^\beta}{\sum_{j \in N_i^k} (\tau_{ij})^\alpha \cdot (\eta_{ij})^\beta} \quad \text{si } j \in N_i^k \quad (0.3)$$

donde N_i^k es el vecindario alcanzable por la hormiga k cuando se encuentra en el nodo i . Los parámetros alfa (α) y beta (β) controlan el proceso de búsqueda. Para $\alpha=0$ se tiene una búsqueda heurística estocástica clásica, mientras que para $\beta=0$ sólo el valor de la feromona tiene efecto. Un valor de $\alpha < 1$ lleva a una rápida situación de convergencia (stagnation) [54]. El vector P_{ij}^k contiene las probabilidades de movimiento calculadas para los nodos de la vecindad (N_i^k) de la hormiga k . El valor τ_{ij} representa el elemento (i, j) en la matriz de feromona y η_{ij} se denomina función de visibilidad o función heurística y mide la calidad de un estado j a partir del i .

El **Sistema de Colonia de Hormigas** [63, 68] es uno de los primeros sucesores del AS que introduce tres modificaciones importantes con respecto a dicho algoritmo de ACO:

-
1. Utiliza una regla de transición distinta y más agresiva, denominada regla proporcional pseudo-aleatoria. Sea k una hormiga situada en el nodo r , $q_0 \in [0,1]$ un parámetro y q un valor aleatorio en el mismo intervalo, el siguiente nodo j se elige como:

$$j = \max_{j \in N_r^k} \{ \tau_{ij} \cdot \eta_{ij}^\beta \} \quad \text{si } q \leq q_0 \quad (0.4)$$

En caso contrario se utiliza la regla probabilística del AS (ecuación (0.3)).

Como puede observarse, la regla tiene una doble intención: cuando $q \leq q_0$, utiliza en gran medida el conocimiento disponible (explotar), eligiendo la mejor opción con respecto a la información heurística y los rastros de feromona. Sin embargo, si $q > q_0$ se aplica una exploración controlada, tal como se hacía en el AS.

2. Las hormigas aplican una actualización en *línea paso a paso* de los rastros de feromona que favorece la generación de soluciones distintas a las encontradas.

Cada vez que una hormiga viaja por una arista a_{ij} , aplica la regla:

$$t_{ij} \leftarrow (1 - \varphi) \cdot t_{ij} + t(0) \quad (0.5)$$

donde $\varphi \in (0,1]$ es un segundo parámetro de decremento de feromona. Como puede verse, la regla de actualización en *línea paso a paso* incluye tanto la evaporación de feromona como la actualización de la misma.

-
3. Se realiza una actualización *fuera de línea* de los rastros de feromona (acción del demonio), donde el ACS sólo considera una hormiga concreta, la que generó la mejor solución global, $S_{mejor-global}^{20}$.

La actualización de la feromona se lleva a cabo evaporando primero estos rastros en todas las conexiones utilizadas por la mejor hormiga global (es importante recalcar que, en el ACS, la evaporación de la feromona sólo se aplica a las conexiones de la solución, que es también la usada para depositar feromona) tal como sigue:

$$t_{ij} \leftarrow (1 - \rho) \cdot t_{ij} \quad \forall a_{ij} \in S_{mejor-global} \quad (0.6)$$

A continuación se deposita feromona a los arcos que pertenecen a la mejor solución encontrada hasta el momento usando la regla:

$$t_{ij} \leftarrow t_{ij} + \Delta t \quad \forall a_{ij} \in S_{mejor-global} \quad (0.7)$$

donde $\Delta t = f(C(S_{mejor-global}))$, es decir la cantidad de feromona está en dependencia de la calidad de la mejor solución encontrada hasta el momento $C(S_{mejor-global})$.

El **Sistema de Hormigas Max-Min** [64, 70] ha sido específicamente desarrollado para obtener una fuerte exploración de las soluciones y para mejorar los estancamientos prematuros de las soluciones presentados por los anteriores algoritmos de ACO en la solución de algunos problemas discretos. Para obtener esta mejora el MMAS presenta los siguientes aspectos:

²⁰Aunque en algunos trabajos iniciales se consideraba también una actualización basada en la mejor hormiga de la iteración [69], en ACS casi siempre se aplica la actualización por medio de la mejor global

-
1. Para explotar las mejores soluciones encontradas durante la ejecución del algoritmo, después de cada iteración sólo una hormiga adiciona feromona a su camino encontrado. Esta hormiga puede ser la que obtuvo la mejor solución en la iteración actual (mejor solución en la iteración) o la hormiga que ha obtenido la mejor solución desde el inicio de la ejecución hasta el momento (mejor solución global). Estudios desarrollados en [64] muestran que para obtener un mejor rendimiento del MMAS es preciso realizar una actualización en la que se utilicen ambos elementos.
 2. Utilizar rangos para los rastros de feromona, de forma tal que ningún rastro sea menor que una cota mínima ni mayor que otra máxima $[T_{min}, T_{max}]$. Estos valores son calculados dinámicamente por el algoritmo [64]; de esta forma se logra no tener mucha diferencia entre valores en la matriz de feromona. Para esto, si algún rastro de feromona es menor que el permitido $T_{ij} < T_{min}$ será inicializado con el valor mínimo permitido $T_{ij} = T_{min}$. Lo mismo sucede para todos los valores de feromona que superan al valor máximo permitido $T_{ij} > T_{max}$, $T_{ij} = T_{max}$.
 3. Adicionalmente, algunos trabajos [64, 71, 72] proponen que los valores iniciales de feromona al inicio del algoritmo sean T_{max} . De esta forma se obtiene una alta exploración del espacio de búsqueda al inicio del algoritmo.

En el MMAS las hormigas se mueven del estado i al j de igual manera que se realiza en el AS (ecuación (0.3)).

En general, las soluciones obtenidas con la metaheurística ACO suelen ser de una calidad moderada. Esto se debe a que estos algoritmos se inclinan hacia una mayor exploración del espacio de búsqueda, por lo

que es razonable aplicarle diversos algoritmos de búsqueda local [56, 60, 67, 69, 73] o alguna estrategia para agregarle un mayor grado de explotación de las soluciones encontradas [70, 74-76].

Análisis de la complejidad temporal

La complejidad temporal de los algoritmos ACO está determinada fundamentalmente por la estrategia de exploración que utilizan, donde un conjunto de m hormigas construyen soluciones de magnitud nn , hasta que se cumpla un número máximo de iteraciones (determinado por la condición de parada cs). Un elemento que influye en la complejidad de esta estrategia es la forma de actualizar los rastros de feromona, pero por ser particular de cada algoritmo ACO, no se tendrá en cuenta para el análisis siguiente.

De manera general la función que describe dicho comportamiento es:

$$f_{ACO}(m, cs, nn) = m \cdot cs \cdot nn \cdot (nn - 1) \quad (0.8)$$

donde $(nn-1)$ es resultado de la naturaleza constructiva de los métodos ACO, ya que una hormiga para moverse del nodo i al j , necesita en el peor de los casos, evaluar todos los nodos adyacentes a i (ecuación (0.3)). Por tanto la complejidad de la estrategia de exploración de la metaheurística ACO es:

$$O(m \cdot cs \cdot nn^2) \quad (0.9)$$

Técnicas estadísticas utilizadas para la validación de los resultados

No existe un procedimiento establecido y aceptado para comparar algoritmos sobre múltiples conjuntos de datos. Un motivo importante es el

comportamiento no determinístico de éstos, por lo que la diferencia detectada entre los resultados de dos algoritmos podría deberse a factores aleatorios, y no a una mejora real [77, 78].

Para intentar determinar si las diferencias encontradas entre dos algoritmos son significativas, los investigadores pueden aplicar distintas técnicas estadísticas. A lo largo de esta memoria se aplican normalmente las pruebas no paramétricas [79], debido a que los resultados obtenidos por las metaheurísticas no cumplen las condiciones requeridas para poder usar de forma correcta comparaciones paramétricas [80-82].

A continuación se resume el proceso seguido para identificar si hay uno o varios algoritmos significativamente mejores que otros; en todos los casos se trabaja con un margen de error del 5% (valor de significación de 0.05); en caso de utilizar otro será especificado.

- 1- Aplicar la prueba de Iman-Davenport [83] para detectar diferencias entre un conjunto de algoritmos.
- 2- Si no se detectan diferencias, se puede concluir que los algoritmos involucrados obtienen resultados que no difieren significativamente unos de otros.
- 3- En otro caso, se pueden utilizar las pruebas de comparaciones múltiples (post-hoc). En este trabajo se aplica específicamente el test Bonferroni-Dunn [84]. En caso de que este no detecte diferencias significativas para un conjunto de más de dos algoritmos, se utiliza una prueba de Holm [85] con el conjunto de algoritmos parecidos, para corroborar los resultados de la prueba anterior. Esto se realiza porque en [81] se presenta Holm como una prueba más potente que Bonferroni-Dunn, pero más costosa.

4- En caso de que sea necesario establecer una comparación entre dos algoritmos, se realiza un prueba de Wilcoxon [86].

Para más información remitirse a [81] donde se presenta con total profundidad el marco experimental utilizado en este trabajo.

Consideraciones finales

Las metaheurísticas poblacionales se caracterizan por mantener en una misma iteración más de una solución. Este elemento provoca que de manera natural sean métodos con un alto nivel de exploración del espacio de solución. Unido a esto, en la revisión bibliográfica desarrollada, se pudo apreciar cómo los investigadores en esta línea se inclinan por introducir más exploración a estos métodos, a través de operadores, para aumentar la diversificación de las soluciones. La explotación de las soluciones se les incorpora, a través de algoritmos de trayectoria simple (formando las metaheurísticas híbridas) en forma de búsqueda local, donde en muchos casos son causantes de cambiar la dirección de la exploración de los algoritmos, limitando su capacidad de diversificación.

Por ello, se hace necesario encontrar mecanismos para aumentar la intensificación de las soluciones en un mismo proceso de búsqueda que provoque la obtención de soluciones más precisas. Todo esto sin tener que aumentar el costo computacional que significa la incorporación de los métodos de trayectoria simple para incrementar la explotación. En términos de complejidad computacional, se conoce que las metaheurísticas poblacionales son métodos que necesitan mucha memoria para mantener el estado de la población. Al mismo tiempo,

invierten mucho tiempo para obtener una buena convergencia de las soluciones.

En el caso específico de la metaheurística ACO, su naturaleza constructiva provoca que se invierta un tiempo adicional en la evaluación de todas las alternativas para la incorporación de un componente a la solución. Esto provoca un aumento considerable en el tiempo de cómputo de estos algoritmos. Por esta razón, se hace necesaria la búsqueda de alternativas para mejorar la estrategia de exploración de los algoritmos ACO, de manera tal que su tiempo de ejecución disminuya considerablemente.

CAPÍTULO 2: OPTIMIZACIÓN BASADA EN COLONIA DE HORMIGAS EN DOS

ETAPAS

En los procesos de búsqueda heurística la determinación del estado inicial desde donde comenzar la búsqueda, ha sido una problemática de interés, pues se ha mostrado que tiene un efecto importante en la solución final. El propósito es poder acercar lo más posible el estado inicial al estado final u objetivo. Por supuesto, es necesario considerar un balance adecuado entre el costo computacional de lograr ese acercamiento y el costo total. No sea que la suma del costo de aproximar el estado inicial al final, más el costo de encontrar la solución desde ese estado inicial "mejorado", sea mayor que el costo de buscar la solución desde un estado inicial original. Además, es necesario que en el proceso de acercamiento no se pierda la posibilidad de explorar buenas soluciones. De modo que el propósito es el siguiente. Sea E_i un estado inicial generado aleatoriamente, u obtenido por cualquier otro método sin un costo computacional significativo; E_i^* un estado inicial generado por algún método M que lo acerca al estado final con un costo $C_M(E_i^*)$; y sea $CC_{ABH}(x)$ el costo computacional de encontrar una solución desde el estado x usando el algoritmo ABH (Algoritmo de Búsqueda Heurística). Entonces, el objetivo es que $C_M(E_i^*) + CC_{ABH}(E_i^*) < CC_{ABH}(E_i)$.

En este capítulo se estudia la problemática anterior para el caso de la búsqueda que se desarrolla en la Optimización Basada en Colonia de Hormigas. El mismo se distribuye de la siguiente manera; seguidamente se presenta una propuesta para la exploración en dos etapas de los algoritmos de ACO. Luego, en el Epígrafe 2.2, se realiza un estudio experimental para seleccionar entre los tres algoritmos básicos AS, ACS y

MMAS, cuál es el que obtiene los mejores resultados en cada uno de los problemas que se utilizan para evaluar la propuesta. En el Epígrafe 2.3 se aplica la estrategia en dos etapas a cada uno de los algoritmos seleccionados en el paso anterior, con el objetivo de estudiar distintas variantes de esta estrategia y seleccionar las que obtienen los mejores resultados. Para finalizar, en el Epígrafe 2.4 se realiza una comparación entre estrategias de exploración, que concluye escogiendo la que mejores resultados obtiene. En todos los casos se utilizan pruebas estadísticas para validar los resultados.

Funcionamiento general de la Optimización Basada en Colonia de Hormigas en Dos Etapas

Para encontrar un estado inicial mejorado que garantice la obtención de buenas soluciones, en este trabajo se propone una forma de exploración para los algoritmos ACO llamada Optimización Basada en Colonia de Hormigas en Dos Etapas (Two Stage Ant Colony Optimization; TS-ACO); en la cual, la estrategia de exploración se divide en dos etapas. En la primera se obtienen resultados preliminares que son utilizados como punto de partida en la segunda etapa (estado inicial mejorado) para obtener soluciones reales al problema en cuestión.

Para realizar la división en dos de la estrategia de exploración, se le introduce al método un factor de proporcionalidad r , el cual se encarga de regular el tamaño que ocupa cada una de las etapas. Este parámetro toma valores en el intervalo $(0, 1)$. Si $r=0$ ó $r=1$ se realiza la exploración en una sola etapa, por lo que no se incluyen entre los valores permitidos para el nuevo método.

Los parámetros que son influenciados por el factor r son los siguientes:

-
- *Tamaño de la colonia (m)*: la cantidad de hormigas designadas para ejecutar el algoritmo se divide en dos conjuntos o sub-colonias, a través de las ecuaciones:

$$\begin{aligned} m_1 &= \text{round}(r \cdot m) \quad (\text{valor del entero más próximo}) \\ m_2 &= m - m_1 \end{aligned} \tag{0.1}$$

donde m_1 representa la cantidad de hormigas que serán utilizadas en la primera etapa, y la cantidad restante m_2 , en la etapa posterior.

- *Profundidad del algoritmo o condición de parada (cs)*: este parámetro pone fin a la ejecución del algoritmo, y al ponerlo en función del parámetro r , se logra obtener una condición de parada para la primera etapa, cs_1 , y otra para la segunda, cs_2 . Por ejemplo, cs puede denotar la cantidad máxima de iteraciones o un tiempo de ejecución total. Estos valores se calculan de la siguiente manera:

$$\begin{aligned} cs_1 &= \text{round}(r \cdot cs) \quad (\text{valor del entero más próximo}) \\ cs_2 &= cs - cs_1 \end{aligned} \tag{0.2}$$

- *Magnitud de la solución (nn)*: este parámetro es totalmente dependiente del problema en cuestión y denota una característica en común de todas las soluciones encontradas. Por ejemplo, en algunos casos este valor puede representar la cantidad de variables a valorizar para construir una solución (cantidad de nodos del grafo) y, en otros, interpretarse en función de una calidad esperada de las soluciones encontradas. Estos

valores son computados como sigue:

$$\begin{aligned}nn_1 &= \text{round}(r \cdot nn) \quad (\text{valor del entero más próximo}) \\nn_2 &= nn - nn_1\end{aligned}\tag{0.3}$$

Como se puede apreciar, la elección del factor r debe tener una alta incidencia en el desempeño del algoritmo. Un valor alto de r , es decir, cercano al valor 1, hace que el resultado de la primera etapa se acerque más al estado objetivo, de modo que se realice muy poca exploración en la segunda y decisiva etapa del método. Además, existen algunos elementos en contra de que la búsqueda en la primera etapa sea mayor que en la segunda y que recomiendan lo contrario:

- i. Existen menos hormigas trabajando, por lo que disminuye la calidad de la búsqueda debido a la reducida cantidad de soluciones exploradas.
- ii. Se realizan pocas iteraciones, esto hace que exista muy poca comunicación en el método.

De modo que resulta clave estudiar cuál debe ser el valor del factor de proporcionalidad r que logre el mejor balance entre las búsquedas en ambas etapas.

Descripción de cada etapa

En el proceso de búsqueda desarrollado por TS-ACO luego de calcular la dimensión de cada una de las etapas se realiza el proceso de exploración, el cual ocurre de la siguiente manera:

-
- *Primera etapa:* las (m_1) hormigas asignadas a esta etapa, construyen una solución de magnitud (nn_1) en cada iteración que se ejecute. En este proceso continuo, los rastros de feromona son actualizados en dependencia del algoritmo ACO utilizado. De esta manera se mantiene la comunicación indirecta presentada por las hormigas. El proceso de exploración termina cuando se cumple la condición de terminación definida para esta etapa (cs_1) . Las mejores soluciones parciales y el estado de los rastros de feromona son almacenados (resultados parciales). La cantidad de soluciones parciales almacenadas está definida por una variable ns que se le introduce al método.
 - *Segunda etapa:* en cada iteración de esta etapa, cada una de las (m_2) hormigas determina su estado inicial, seleccionado de manera aleatoria entre las mejores soluciones parciales de la etapa anterior (ns) . Luego, cada hormiga completa su estado inicial hasta alcanzar una solución de magnitud nn_2 ; el proceso termina cuando se ha cumplido con la condición de parada calculada para esta etapa (cs_2) . En este proceso de exploración se utilizan los rastros de feromona resultantes de la etapa anterior para guiar la búsqueda.

Por ejemplo, considere el problema del Viajante de Comercio, en el cual hay que encontrar el camino mínimo entre n ciudades (en el Epígrafe 2.4.1.1 se describe formalmente). En este problema siempre las soluciones incluyen las n ciudades; de modo que esta característica puede tomarse como magnitud de la solución. Si se definen los parámetros siguientes: 10 hormigas, 100 iteraciones a ejecutar por el algoritmo ACO seleccionado para una instancia de 18 ciudades y considerando un factor de

proporcionalidad $r=0,3$ entonces, 3 hormigas ($0,3 \cdot 10$) construirán caminos de 6 ciudades ($0,3 \cdot 18$) durante 30 iteraciones ($0,3 \cdot 100$) en la primera etapa. Luego se seleccionan los mejores caminos parciales, y estas sub-soluciones se toman como estados iniciales en la segunda etapa. Para esta última etapa, las restantes 7 hormigas completaran los caminos (agregando 12 ciudades no incluidas en la solución parcial seleccionada como estado inicial) durante 70 ciclos.

El estudio del comportamiento de la exploración en dos etapas, tal y como se describió anteriormente, fue presentado en [87-93] para la solución de distintos problemas discretos (Viajante de Comercio, Asignación Cuadrática, Selección de Rasgos, Secuenciación de Tareas y Planificación de la Fuerza de Trabajo); donde en comparaciones realizadas con la propuesta de exploración de ACO, se logró reducir el tiempo de ejecución entre un 60% y un 40 %, sin alterar la calidad de los resultados.

Además, en este trabajo se presentan dos alternativas para la búsqueda en dos etapas, con el objetivo de aumentar la exploración en la segunda fase de la propuesta y, de esta forma probar el efecto que provoca para la calidad de las soluciones computadas:

1. *Basada en feromona* (Pheromone Two Stage Ant Colony Optimization; P-TS-ACO): Solo los rastros de feromona resultante de la etapa anterior son utilizados en la segunda etapa. En ellos se encuentra toda la información memorística de la exploración realizada. En la segunda etapa, las hormigas comienzan su exploración desde una única componente (de la misma forma que en la primera etapa) y se valorizan todas las variables del

problema, por lo que $nm_2 = nm$ (ecuación (0.3)). De esta forma, la exploración sólo queda limitada por la preferencia de los rastros de feromona que vienen de la etapa anterior.

2. *Basada en soluciones* (Solutions Two Stage Ant Colony Optimization; S-TS-ACO): Las hormigas de la segunda etapa parten de una de las mejores soluciones parciales encontradas en la primera etapa ns . Los rastros de feromona utilizados en esta etapa son reinicializados. En esta alternativa la exploración sólo queda limitada por las magnitudes de las soluciones que se deben completar con el proceso constructivo de la segunda etapa.

Estas dos alternativas realizan una mayor exploración del espacio de búsqueda en la segunda etapa que la propuesta original TS-ACO. Esto se debe a que sólo reutilizan una información de las resultantes en la primera etapa (mejores soluciones parciales o rastros de feromona).

Tanto para TS-ACO como para la variante S-TS-ACO, se incorpora un parámetro ns que regula la cantidad de soluciones parciales que pasan de un estado a otro. Este parámetro, influye en el nivel de exploración que se desee introducir al método. Un valor alto hace que se complete una mayor cantidad de soluciones parciales; por lo que el nivel de exploración global en la etapa aumenta en gran medida, lo que no es muy recomendable en las etapas finales de los algoritmos ACO. Mientras que, si se almacenan pocas soluciones, se pasaría a completar siempre las mismas, lo que provocaría que el método se estanque en óptimos locales rápidamente. En consecuencia con lo anteriormente explicado, en esta investigación se utiliza un valor de ns igual a 5% de las soluciones computadas en la primera etapa. Valor estudiado anteriormente [90].

La Ilustración 3 representa la forma general de exploración en dos etapas de los algoritmos de ACO, donde se puede apreciar que difiere de la metaheurística original en los siguientes aspectos:

- i. El proceso de búsqueda se divide en dos etapas, donde en cada una de ellas se aplica un algoritmo de ACO para dar solución a una parte del problema.
- ii. Se incorporan dos variables nuevas; r define la dimensión de cada una de las etapas y ns indica la cantidad de soluciones parciales que se almacenan en el proceso de búsqueda de la primera etapa.

Se presentan alternativas para aumentar la exploración en la segunda etapa del método, en las cuales se reutilizan indistintamente tanto los rastros de feromona como las mejores soluciones parciales.

Dados los parámetros del método (*factor (r), cantidad de soluciones parciales almacenadas (ns)*)

Paso1: Etapa 1.

Paso1.1: Calcular los parámetros que definen la dimensión de la primera etapa.

Paso1.2: Aplicar cualquier algoritmo de la metaheurística ACO.

Paso1.3: $El \leftarrow$ las ns mejores soluciones parciales encontradas en la primera etapa (en dependencia de la alternativa utilizada).

Paso2: Etapa 2.

Paso2.1: Calcular los parámetros para el proceso de búsqueda en la segunda etapa.

Paso2.2: Aplicar algoritmo de ACO utilizando como estados iniciales,

las soluciones parciales seleccionadas aleatoriamente del conjunto EI y los rastros de feromona resultantes de la primera etapa (en dependencia de la alternativa utilizada).

Paso3: Fin del proceso.

Ilustración 3: Modelo general de TS-ACO.

El proceso en dos etapas no altera para nada los principales elementos de los algoritmos de ACO, desde la regla de transición utilizada hasta la forma de actualizar los rastros de feromona. Como se puede apreciar, cada paso es totalmente independiente del otro. Esta característica hace que sea posible aplicar un algoritmo de ACO diferente en cada una de las etapas. Aunque en los estudios realizados en esta investigación se utilizó siempre el mismo algoritmo de ACO, esta independencia de los pasos en el proceso en dos etapas es un elemento que se puede tener en cuenta al aplicar la estrategia de exploración.

Análisis de complejidad

Como se explicó anteriormente, la búsqueda en dos etapas no modifica la forma interna de ejecución de los algoritmos ACO, sino solamente la manera en que estos realizan la exploración. Las variables que definen el tamaño de la exploración (cantidad de hormigas m , cantidad de iteraciones cs y el tamaño de las soluciones construidas nn) están influenciadas directamente por el parámetro r , el cual divide en dos etapas el proceso de búsqueda de ACO. La función que describe dicho comportamiento es:

$$f_{TS-ACO}(m, cs, nn) = f_{etapa1}(m, cs, nn) + f_{etapa2}(m, cs, nn) \quad (0.4)$$

donde

$$f_{etapa1}(m, cs, nn) = rm \cdot rcs \cdot rnn \cdot (n - 1) = r^3(m \cdot cs \cdot nn) \cdot (nn - 1) \quad (0.5)$$

además

$$\begin{aligned} f_{etapa2}(m, cs, nn) &= (1 - r)m \cdot (1 - r)cs \cdot (1 - r)nn \cdot (nn - 1) \\ &= (1 - r)^3(m \cdot cs \cdot nn) \cdot (nn - 1) \end{aligned} \quad (0.6)$$

Sumando las funciones que describen cada etapa, el resultado es:

$$\begin{aligned} f_{TS-ACO}(m, cs, nn) &= f_{etapa1}(m, cs, nn) + f_{etapa2}(m, cs, nn) \\ &= (3r^2 - 3r + 1) \cdot (m \cdot cs \cdot nn) \cdot (nn - 1) \end{aligned} \quad (0.7)$$

Este comportamiento está acotado superiormente por la expresión:

$$O(m \cdot cs \cdot nn^2) \quad (0.8)$$

En términos de complejidad presenta la misma que ACO pero si se analiza el coeficiente $(3r^2 - 3r + 1)$ de la función f_{TS-ACO} , se tiene que el valor mínimo se alcanza para $r=0,5$. La Ilustración 4 muestra el comportamiento general del coeficiente para distintos valores de r , donde para $r=1$ ó $r=0$ el valor de éste, hace que la estrategia sea igual a la de ACO.

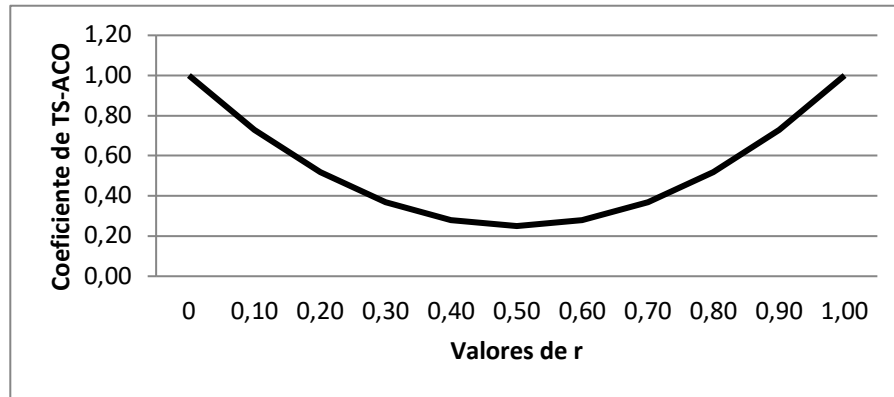


Ilustración 4: Efecto del parámetro r para TS-ACO

Este resultado provoca que el tiempo de ejecución de TS-ACO disminuya considerablemente en dependencia del valor de r , aunque los dos métodos tengan la misma complejidad computacional.

Estudio experimental

Los resultados experimentales encontrados al estudiar el desempeño de TS-ACO en diferentes problemas discretos [87-94], son coherentes con el análisis desarrollado en el epígrafe anterior, donde el tiempo de ejecución se redujo considerablemente entre un 40% y un 60%. De modo que el nuevo método es capaz de realizar una exploración más eficiente²¹ que ACO para cualquier valor de r , obteniéndose la máxima eficiencia para $r = 0,5$.

Esta conclusión no garantiza la calidad de las soluciones computadas por TS-ACO. Por ello resulta de gran importancia encontrar para cuáles valores de r se obtiene el mejor balance entre eficacia²² y eficiencia de la propuesta. Esta interrogante fue estudiada en los trabajos publicados

²¹ Referido al tiempo de ejecución de los método para obtener las soluciones

²² Referido a la calidad de las soluciones encontradas

[89, 92], donde se probaron distintos valores de $r=\{0,2; 0,25; 0,3; 0,35; 0,4; 0,5; 0,7, 0,8\}$ y se concluyó que para $r=\{0,2; 0,25; 0,3\}$ el método obtiene un mejor balance entre eficacia y eficiencia; obteniéndose resultados que no difieren significativamente en cuanto a calidad con los computados por ACO.

A partir de estas conclusiones, se decidió en esta investigación probar cuál de las dos estrategias de exploración (TS-ACO vs ACO) logra encontrar las mejores soluciones en un mismo tiempo de exploración. Para ello, se seleccionó como condición de parada un tiempo máximo de ejecución. Para seleccionar el algoritmo ACO al cual se le aplicará la exploración en dos etapas, se presenta a continuación una experimentación de los tres algoritmos clásicos: AS, ACS y MMAS.

Todos los algoritmos fueron implementados en el lenguaje java JDK 1.5, como entorno de desarrollo (IDE) se utilizó NetBeans 5.1 y fueron ejecutados en un procesador Intel Pentium IV a 3.4 GHz, con 1 GB de RAM.

Selección del mejor algoritmo basado en colonia de hormigas

Los algoritmos de ACO, y especialmente AS, ACS y MMAS, han sido aplicados con muy buenos resultados a los dos problemas que se consideran seguidamente [51, 55, 56, 61]. En todos los casos se hace necesario aplicar algoritmos de búsqueda local para mejorar las soluciones encontradas por las hormigas. Específicamente en este trabajo se utiliza el algoritmo 2-opt [64, 95].

Caso de estudio: Problema del Viajante de Comercio

El Viajante de Comercio (Travelling Salesman Problem; TSP) [96], fue uno de los primeros problemas en estudiarse, probando Karp en [97], que pertenece a la clase de los problemas difíciles (NP-hard). Debido a su alta complejidad, se ha puesto a la disposición de la comunidad científica la librería TSPLIB²³ [98] de dominio público, la cual contiene un conjunto de ejemplos del TSP con la mejor solución obtenida hasta la fecha y, en algunos casos, con la solución óptima.

El Problema del Viajante puede enunciarse del modo siguiente:

Un viajante de comercio ha de visitar n ciudades, comenzando y finalizando en su propia ciudad y no visitando más de una vez ninguna ciudad intermedia. Conociendo el costo de ir de cada ciudad a otra, determinar el recorrido de costo mínimo.

Para enunciar el problema formalmente se introduce la siguiente terminología: Sea $G=(V; A; D)$ un grafo, donde $V=(v_i)$ es el conjunto de vértices²⁴, $A=(a_{ij})$ es el conjunto de aristas, y $D=(d_{ij})$ es la matriz de costos. Cada elemento (i, j) de la matriz C representa un costo o distancia de la arista a_{ij} . De modo que el objetivo es encontrar una permutación en el conjunto V , $P=\{v_0, v_1, \dots, v_{n-1}\}$ tal que la distancia sea mínima:

$$d_p = \sum_{i=0}^{n-1} c(v_i, v_{i+1 \bmod n}) \quad (0.9)$$

Desde su origen, la metaheurística ACO ha sido probada en la solución del TSP [57, 99] en su versión simétrica. Por su naturaleza, los rastros de

²³ Accesible desde <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>

²⁴ Indistintamente se utiliza en esta investigación los términos nodos y vértices.

feromona son asociados a los arcos del grafo. En el proceso de solución, las hormigas de la colonia son posicionadas de forma aleatoria en uno de los nodos del grafo (una ciudad). Luego cada una de ellas se desplaza por el grafo utilizando la regla de transición de estados correspondiente con el algoritmo utilizado (AS, ACS, MMAS). Para este problema la función heurística o de visibilidad es $\eta_{ij}=1/d_{ij}$, donde d_{ij} representa la distancia de la ciudad i a la j . Después que cada hormiga ha seleccionado la siguiente ciudad en la construcción de su recorrido, ésta se incorpora en su memoria tabú y se procede a la actualización de los rastros de feromona en línea (sólo el ACS). Este proceso se ejecuta hasta que todas las hormigas hayan completado su recorrido. En este momento, se efectúa una búsqueda local (a todas las soluciones encontradas o a la mejor de la iteración). Por último, se realiza la evaporación y actualización fuera de línea de los rastros de feromona (en dependencia del algoritmo utilizado). Este proceso se repite hasta que se cumpla con una condición de parada o de terminación del algoritmo.

Los valores de cada uno de los parámetros utilizados en este primer estudio son mostrados en la

Tabla 1.

Tabla 1: Configuración de parámetros utilizados.

Algoritmo	m	$\tau(0)$	ρ	Φ	α	β	q_0	LS (2-opt)	Condición de parada
AS	10	0,2	0,1	-	2	3	-	Literación	t_{max}
ACS	10	0,2	0,1	0,1	1	5	0,8	Literación	t_{max}
MMAS	10	T_{max}	0,1	-	2	3	-	Literación	t_{max}

La columna *LS (2-opt)* hace referencia al momento en que se aplica el algoritmo de búsqueda local 2-opt por cada algoritmo, donde *Literación* representa la mejor solución de cada iteración. El término t_{max} en la columna “Condición de parada” se refiere a un tiempo máximo de ejecución y, fue seleccionado luego de ejecutar la propuesta de exploración en dos etapas del algoritmo ACS presentado en [90] (10 hormigas y 1000 iteraciones para $r=0,25$) para cada una de las instancias involucradas en este estudio. Esta configuración tiene como objetivo probar cuán eficaz se comporta ACO en el tiempo de exploración requerido por TS-ACO.

Para el caso del algoritmo MMAS los valores máximo y mínimo de feromona son calculados de la siguiente forma [64]:

$$\tau_{max} = (1/(1-\rho)) \cdot (1/f(C(S_{mejor-global}))) \quad (0.10)$$

donde $f(C(S_{mejor-global}))$ representa la calidad de la mejor solución encontrada por la colonia en todo el proceso de búsqueda. El valor mínimo depende totalmente del valor máximo y se calcula como sigue:

$$\tau_{min} = \tau_{max} / 10 \cdot n \quad (0.11)$$

donde n representa la dimensión de la instancia seleccionada.

A continuación, se presenta el estudio comparativo entre los resultados alcanzados por los algoritmos AS, ACS y MMAS. Para validar las comparaciones realizadas, se hace uso de las técnicas estadísticas presentadas en el Epígrafe 1.3²⁵.

En la **¡Error! No se encuentra el origen de la referencia.** los resultados del test de Iman-Davenport muestran que existen diferencias significativas entre los resultados de los algoritmos que conforman el grupo de control; esto se debe a que el valor p es menor que el valor de significación seleccionado (0,05). Este resultado provoca que la hipótesis sea rechazada (R). De manera que se hace necesario ejecutar los tests de comparaciones múltiples para focalizar dichas diferencias.

Tabla 2: Resultados del test de Iman-Davenport para el grupo de control en la solución del TSP.

Método	Valor del Test	Valor de distribución	Valor p	Hipótesis
Iman-Davenport	25,000	3,4434	1,54E-6	R

En los resultados de la Ilustración 5 se muestra cómo el algoritmo ACS (muestra de control) es significativamente mejor que al AS, cuyo orden medio es mayor que el patrón de comparación (2,123) definido por valor crítico (CD) computado para este caso. En cuanto al MMAS no se encontraron diferencias significativas en comparación con la muestra de control.

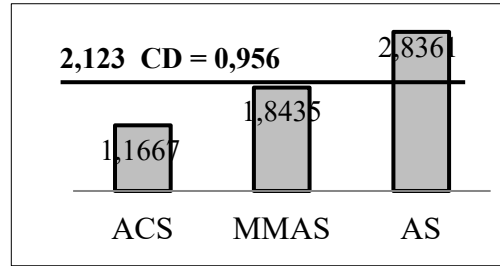


Ilustración 5: Resultados del test de Bonferroni-Dunn para el TSP.

Seguidamente se presenta en la Tabla 3 **Error! No se encuentra el origen de la referencia.** un test de Wilcoxon con el objetivo de validar las conclusiones anteriores entre el ACS y MMAS. Para este caso, la hipótesis de igualdad es aceptada; por lo que se deduce que no son significativas las diferencias entre ambos algoritmos, aunque sí hay diferencia a favor del ACS, debido a que los rangos negativos (R^-) son superiores a los rangos positivos (R^+).

Debido a estos resultados el algoritmo ACS fue seleccionado entre el conjunto de algoritmos de ACO en la solución del TSP, para aplicarle la exploración en dos etapas.

Tabla 3: Resultados del test de Wilcoxon para el TSP.

Algoritmo	R^+	R^-	Valor p	Hipótesis
ACS vs MMAS	20,00	58,00	0,136	A

Caso de estudio: Problema de Asignación Cuadrática

El Problema de Asignación Cuadrática (Quadratic Assignment Problem; QAP) [100] es un problema clásico en optimización combinatoria y, básicamente, consiste en encontrar una asignación óptima de n recursos a n localidades con el propósito de minimizar el costo de transporte,

dados una matriz de requerimientos de unidades a transportar y el respectivo costo de transporte por unidad entre las localidades. El QAP fue propuesto por Koopmans y Beckmann en 1957; y en 1976 Shani y González probaron que QAP es un problema NP-completo [101].

Su planteamiento como problema de optimización combinatoria es el siguiente:

Sean $N=\{1,2,\dots, n\}$, $F=(f_{ij})$ y $D=(d_{kl})$ dos matrices cuadradas de $(n \cdot n)$; se trata de encontrar la asignación de n plantas a n localidades; es decir, una permutación $\pi \in S_{\mathfrak{N}}$ que minimice (ecuación (0.12)).

$$\pi_N = \sum_{i=1}^n \sum_{j=1}^n d_{ij} f_{\pi(i)\pi(j)} \quad (0.12)$$

donde $S_{\mathfrak{N}}$ es el conjunto de todas las permutaciones del conjunto N . Además, f_{ij} (facilidades) representa el flujo de materiales de la planta i a la planta j , y d_{kl} es la distancia de la localidad k a la localidad l .

La manera en que se aplica ACO al TSP puede ser extendida al QAP de forma directa [58, 64, 102]. La principal diferencia consiste en la definición de las componentes de una solución, las cuales para el QAP representan una asignación de facilidades a localizaciones. En la representación del grafo cada uno de los nodos coincide con una determina asignación y los arcos determinan la transición de una asignación a otra. Para este problema, como para el TSP, los rastros de feromona son asociados a los arcos del grafo [58], determinando cuán deseada ha sido una cadena de asignaciones. Para la construcción de la solución puede ser conveniente utilizar un pre-ordenamiento de las facilidades (o

equivalente de las localizaciones) y asignar localizaciones en el orden tomado (o facilidades). Cada hormiga, para seleccionar qué localización poner en la próxima facilidad, utiliza la regla de transición de estado correspondiente al algoritmo utilizado (AS, ACS, MMAS). En [64] se propone no hacer uso de la función heurística si se utiliza algún algoritmo de búsqueda local para mejorar las soluciones. El proceso continúa de la misma forma que para el TSP, incorporando la localización seleccionada en la lista tabú de cada hormiga, realizando la actualización de los rastros de feromona en línea, y cuando cada hormiga haya realizado su asignación completa, se efectúa la evaporación y actualización de la feromona fuera de línea. Todas estas acciones toman lugar en dependencia del algoritmo utilizado.

Las instancias del problema con las que se pondrán a prueba los algoritmos propuestos en este documento se han tomado de la biblioteca de referencia QAPLIB²⁶. Los valores de los parámetros seleccionados para la comparación son mostrados en la Tabla 4 con la misma característica que la Tabla 1 **¡Error! No se encuentra el origen de la referencia..**

Tabla 4: Configuración de los parámetros utilizados.

Algoritmo	m	$\tau(0)$	ρ	Φ	α	β	q_0	LS (2-opt)	Condición de parada
AS	10	0,2	0,8	-	1	-	-	L_k	t_{max}
ACS	10	0,2	0,8	0,8	1	-	0,6	L_k	t_{max}
MMAS	10	T_{max}	0,8	-	1	-	-	L_k	t_{max}

Para este caso en particular, la búsqueda local (LS (2-opt)) es aplicada a todas las soluciones encontradas por las hormigas (L_k) en cada iteración

²⁶ accesible en http://serv1.imm.dtu.dk/_sk/qaplib/

de los algoritmos [58]. El tiempo máximo de ejecución ($t_{i_{max}}$) es computado en dependencia de la dimensión de cada instancia; para las que su dimensión (n) es menor que 50, el tiempo máximo de ejecución es $(n \cdot 10)$ segundos, y en el caso de las instancias cuya dimensión excede o es igual que 50, el tiempo máximo es $(n \cdot 20)$ segundos. Los valores máximos y mínimos de los rastros de feromona para el algoritmo MMAS son computados según las ecuaciones (0.10) y (0.11).

Tabla 5: Resultados del test de Iman-Davenport para los algoritmos de ACO en el QAP.

Método	Valor del Test	Valor de distribución	Valor p	Hipótesis
Iman-Davenport	27,580	3,1751	6,82E-9	R

La Tabla 5; **Error! No se encuentra el origen de la referencia.** muestra que la hipótesis de igualdad es rechazada según el test de Iman-Davenport para un nivel de confianza de 0,05; lo que garantiza que existen diferencias significativas entre el conjunto de resultados de los tres algoritmos en cuestión.

Para el caso del test de comparaciones múltiples presentado en la Ilustración 6 se muestra que el algoritmo ACS obtiene resultados significativamente peores que el algoritmo de control (menor orden medio, MMAS), ya que su orden medio está por encima del patrón de comparación (2,008) definido por el valor crítico computado para esta comparación. En el caso de la comparación entre el MMAS y AS no se detectan diferencias significativas.

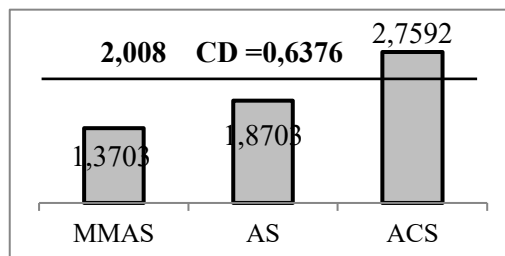


Ilustración 6: Resultados del test de Bonferroni-Dunn para el QAP con MMAS como algoritmo de control.

En la Tabla 6 **¡Error! No se encuentra el origen de la referencia.** se presentan los resultados del estadístico de Wilcoxon para corroborar el análisis de Bonferroni-Dunn, donde observa que la hipótesis de igualdad es rechazada a favor del algoritmo MMAS, debido a que el valor p es menor que el valor de significación 0,05, por lo que es estadísticamente mejor que el AS. De esta manera se concluye que los mejores resultados para el QAP son obtenidos por el MMAS.

Tabla 6: Resultados del test de Wilcoxon.

<i>Algoritmo</i>	R^+	R^-	Valor p	Hipótesis
MMAS vs AS	78,00	273,00	0,013	R

Por los análisis experimentales desarrollados anteriormente se puede concluir que el algoritmo ACS es el que mejores resultados brinda en la solución del TSP y el MMAS para el QAP. Estos algoritmos son seleccionados para realizar el análisis experimental de aplicarles la exploración en dos etapas.

Estudio experimental de la exploración en dos etapas

Para esta experimentación, se aplicó la estrategia en dos etapas y sus alternativas a los algoritmos ACO elegidos en el estudio anterior. Para lo cual, los valores de r seleccionados son $r=\{0,2; 0,25; 0,3\}$, de acuerdo a los resultados publicados [88, 90]. Las alternativas que utilizan como estados

iniciales en la segunda etapa, las mejores soluciones parciales encontradas en la etapa anterior, la cantidad almacenada representa el 5% de la cantidad de soluciones generadas en la primera etapa $ns = 5\%$ ($m_1 \cdot cs_1$).

Para mantener una alta fiabilidad en las pruebas estadísticas utilizadas, las cuales necesitan que la cantidad de ejemplos resueltos sea mucho mayor que la cantidad de algoritmos involucrados en la comparación, se estructuró la experimentación de la siguiente forma:

- 1- Se selecciona cuál de las alternativas (TS-ACO, S-TS-ACO, P-TS-ACO) obtiene los mejores resultados para cada valor de r .
- 2- Luego se compara cada una de las alternativas ganadoras para cada valor de r con la propuesta original de ACO seleccionada en el estudio anterior.

Caso de estudio: Problema del Viajante de Comercio

Para el caso del TSP la estrategia en dos etapas aplicada al ACS [90-92] obtiene en la primera etapa recorridos parciales, los cuales contienen un subconjunto del total de ciudades. Para medir la calidad de estas soluciones parciales no se tiene en cuenta el regresar a la ciudad de inicio. En dependencia de la alternativa utilizada los mejores recorridos parciales son almacenados o no. En cuanto al efecto de aplicar la búsqueda local en cada etapa, en [90] se presenta un estudio de por qué es aconsejable introducirla al finalizar la segunda etapa del método.

Caso de estudio: Problema de Asignación Cuadrática

Al igual que para el TSP, la aplicación del MMAS en dos etapas al QAP [88], se encarga de encontrar en la primera etapa asignaciones parciales, en las cuales sólo una parte de las localizaciones son asignadas a una misma cantidad de facilidades. Al finalizar esta fase las mejores cadenas de asignaciones parciales son almacenadas. El proceso en la segunda etapa comienza utilizando como punto de partida de las hormigas el conjunto de asignaciones parciales o desde un nodo del grafo, en dependencia de la alternativa utilizada (en los dos casos el punto de partida se selecciona de manera aleatoria). Las hormigas se mueven por el grafo hasta obtener asignaciones completas. De la misma forma que para el TSP, el operador de búsqueda local se aplica a las soluciones encontradas en la segunda etapa.

Análisis de los resultados

A continuación, se presenta el análisis estadístico de los resultados para seleccionar la mejor alternativa en cada problema de estudio.

La Tabla 7 muestra los resultados del test de Iman-Davenport para los conjuntos de algoritmos agrupados por el factor r en cada problema. Atendiendo al valor de p ($> 0,05$), en cada caso, se detectan diferencias significativas en los grupos de control, tanto en el TSP como en el QAP.

Tabla 7: Resultados del test de Iman-Davenport para cada grupo de algoritmos en la solución del TSP y el QAP.

Método	Valor del Test	Valor de distribución	Valor p	Hipótesis
TSP				
Iman-	18,886	$r=0,2$ 3,4434	1,67E-5	R

Davenport				
		$r=0,25$		
Iman-Davenport	34,257	3,4434	1,74E-7	R
		$r=0,3$		
Iman-Davenport	47,666	3,4434	1,00E-8	R
QAP				
		$r=0,2$		
Iman-Davenport	11,073	3,1751	9,85E-5	R
		$r=0,25$		
Iman-Davenport	16,713	3,1751	2,48E-6	R
		$r=0,3$		
Iman-Davenport	18,181	3,1751	1,03E-6	R

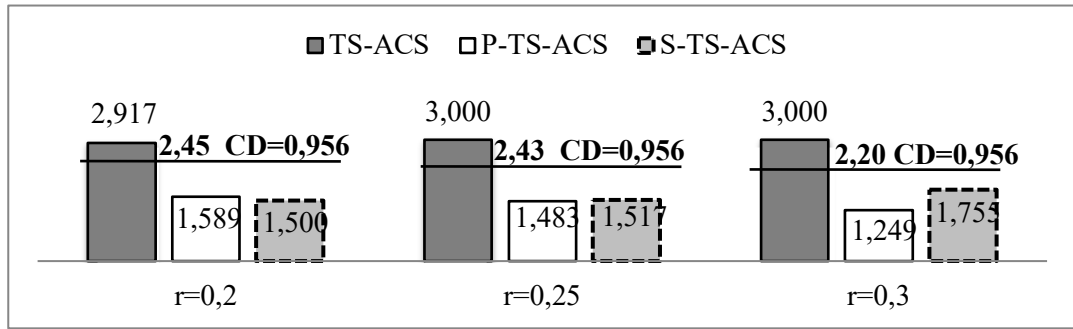


Ilustración 8: Resultados del test de Bonferroni-Dunn para cada valor de r en la solución de TSP.

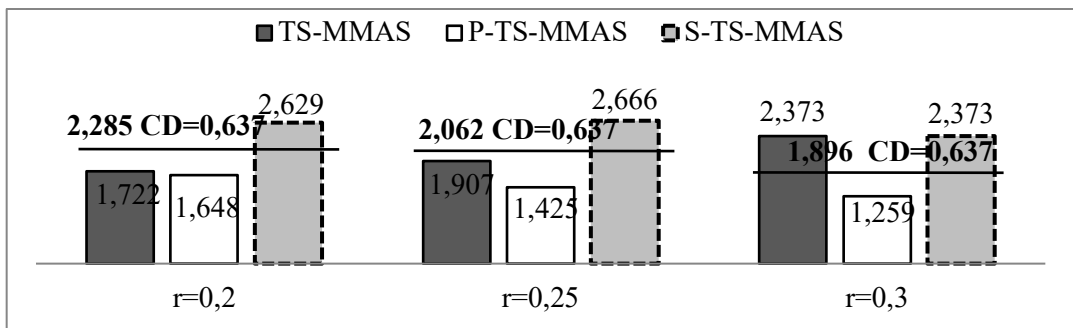


Ilustración 7: Resultados del test de Bonferroni-Dunn para cada valor de r en la solución del QAP.

En la Ilustración 7 e Ilustración 8 se presentan los resultados del test de Bonferroni-Dunn para cada valor de r en la solución de los problemas de estudio, concluyéndose que:

1. Para el TSP
 - a. $r=0,2$, el algoritmo de control S-TS-ACS (menor orden medio) obtuvo resultados significativamente mejores que el TS-ACS y no significativos con respecto al P-TS-ACS.
 - b. $r=0,25$, el algoritmo de control P-TS-ACS obtuvo resultados significativamente mejores que el TS-ACS y no significativos con respecto al S-TS-ACS.

c. $r=0,3$, presenta el mismo comportamiento que el caso anterior.

2. Para el QAP

a. $r=0,2$, el algoritmo de control P-TS-MMAS (menor orden medio) obtuvo resultados significativamente mejores que el TS-MMAS y no significativos con respecto al S-TS-MMAS.

b. $r=0,25$, presenta el mismo comportamiento que el caso anterior.

c. $r=0,3$, el algoritmo de control P-TS-MMAS obtiene resultados significativamente superiores que los demás algoritmos.

En las Tabla 8 y

Tabla 9 se muestran los resultados del test de Wilcoxon con el objetivo de comprobar los resultados del test de Bonferroni-Dunn para los casos semejantes de cada comparación.

Tabla 8: Resultados del test de Wilcoxon para cada valor de r en el TSP.

Algoritmo	R^+	R^-	Valor p	Hipótesis
S-TS-ACS vs P-TS-ACS	20,00	58,00	$r=0,2$ 0,136	A
S-TS-ACS vs P-TS-ACS	49,00	29,00	$r=0,25$ 0,433	A
S-TS-ACS vs P-TS-ACS	56,00	22,00	$r=0,3$ 0,183	A

Tabla 9: Resultados del test de Wilcoxon para cada valor de r en el QAP.

Algoritmo	R^+	R^-	Valor p	Hipótesis
P-TS-MMAS vs TS-MMAS	140,5	168,5	$r=0,2$ 0,689	A
P-TS-MMAS vs TS-MMAS	74,0	257,0	$r=0,25$ 0,014	R

Como conclusión de estas tablas se tiene que:

1. Resultados para el TSP: entre las alternativas S-TS-ACS y P-TS-ACS no se detectaron diferencias significativas para todos los valores de r ; aunque según los rangos positivos y negativos P-TS-ACS obtiene mejores resultados que S-TS-ACS para $r=0,25$ y $0,3$; no siendo así para $r=0,2$, donde la alternativa S-TS-ACS alcanza mejores resultados.
2. Resultados para el QAP: para $r=0,2$ no se detectan diferencias significativas entre P-TS-MMAS y TS-MMAS, aunque los valores de rangos se inclinan a favor del primero. En el caso de $r=0,25$ la hipótesis es rechazada a favor de P-TS-MMAS, lo que garantiza que sus resultados son significativamente mejores que las demás variantes.

En cada caso, la alternativa seleccionada fue aquella que obtuvo resultados significativamente superiores a las demás o, en caso de semejanza, la que según los rangos fue superior. A continuación, se muestra la

Tabla 10 con los algoritmos seleccionado para cada problema de estudio.

Tabla 10: Algoritmos seleccionados para cada problema.

	$r=0,2$	$r=0,25$	$r=0,3$
TSP	S-TS-ACS	P-TS-ACS	P-TS-ACS
QAP	P-TS-MMAS	P-TS-MMAS	P-TS-MMAS

Estudio comparativo entre ambas estrategias de exploración

En este epígrafe se presenta un estudio comparativo entre cada una de las alternativas de control seleccionadas en el estudio anterior, con cada uno de los algoritmos ACO obtenidos como conclusión en el Epígrafe 2.4.1. Para esta comparación se utiliza el test de Wilcoxon con un valor de significación de 0,1.

Como resultado de la Tabla 11 **¡Error! No se encuentra el origen de la referencia.** se tienen las siguientes conclusiones:

1. Para el TSP, los mejores resultados son alcanzados por la exploración en dos etapas con la alternativa S-TS-ACS para $r=0,2$ y P-TS-ACS para $r=0,3$. En el caso de $r=0,25$ no se detectan diferencias significativas entre los algoritmos P-TS-ACS y el ACS; aunque según los valores de los rangos positivos y negativos, la alternativa de la estrategia en dos etapas es superior al método original.
2. En el caso de QAP, para todos los valores de r experimentados, la segunda alternativa de la exploración en dos etapas (P-TS-MMAS) obtiene resultados significativamente mejores que el algoritmo MMAS.

Tabla 11: Resultados del test de Wilcoxon entre estrategias de exploración para cada problema.

Algoritmo	R^+	R^-	Valor p	Hipótesis
TSP				
S-TS-ACS($r=0,2$) vs ACS	10,00	68,00	0,023	R
P-TS-ACS($r=0,25$) vs ACS	28,00	50,00	0,388	A
P-TS-ACS($r=0,3$) vs ACS	15,00	63,00	0,058	R
QAP				
P-TS-MMAS($r=0,2$) vs MMAS	94,50	214,50	0,086	R
P-TS-MMAS($r=0,25$) vs MMAS	90,00	218,00	0,070	R
P-TS-MMAS($r=0,3$) vs MMAS	89,50	219,50	0,060	R

Conclusiones Parciales

Se ha presentado una nueva estrategia de búsqueda para los algoritmos de ACO, donde la exploración es dividida en dos etapas. La primera fase del método es la encargada de obtener los estados iniciales en el proceso de búsqueda desarrollado en la segunda etapa. Para definir el tamaño que ocupa cada etapa, se le introduce al método un factor de proporcionalidad r que afecta a la fuerza de exploración y a la dimensión del problema.

Esta forma de exploración del espacio de solución en dos etapas (TS-ACO) logra reducir el tiempo de ejecución de los algoritmos ACO entre el 40% y el 60%, en dependencia del valor de r ; obteniendo resultados de buena calidad. Los valores de r para los que el método obtiene el mejor equilibrio entre eficiencia y eficacia son $r=\{0,2; 0,25; 0,3\}$ (conclusiones reportadas en [89, 90, 93]).

Para mejorar la calidad de los resultados obtenidos por TS-ACO se presentan dos alternativas de búsqueda P-TS-ACO y S-TS-ACO, las cuales aumentan el nivel de exploración en la segunda etapa del método. Esto

se lleva a cabo utilizando indistintamente los rastros de feromona y las mejores soluciones parciales, obtenidos ambos en el proceso desarrollado en la primera etapa.

Según los estudios experimentales desarrollados para el TSP y el QAP se concluye que la estrategia basada en dos etapas es capaz de obtener soluciones con mejor calidad que las encontradas por la metaheurística original (ACO) en el mismo tiempo de ejecución. Esto sucede específicamente en las dos variantes que utilizan los resultados parciales (mejores soluciones parciales y rastros de feromona) de forma independiente (P-TS-ACO y S-TS-ACO). De esta manera, se logra mantener un nivel de exploración en la etapa decisiva del método; lo cual es fundamental para obtener buenas soluciones.

CAPÍTULO 3: OPTIMIZACIÓN BASADA EN MALLAS DINÁMICAS

La Optimización Basada en Mallas Dinámicas (Dynamic Mesh Optimization; DMO) es una metaheurística poblacional con características evolutivas donde un conjunto de nodos que representan soluciones potenciales a un problema de optimización forma una malla (población) que dinámicamente crece y se desplaza por el espacio de búsqueda (evoluciona). Para ello, se realiza un proceso de expansión en cada ciclo, donde se generan nuevos nodos en dirección a los extremos locales (nodos de la malla con mejor calidad²⁷ en distintas vecindades) y el extremo global (nodo obtenido de mejor calidad en todo el proceso desarrollado); así como a partir de los nodos fronteras de la malla. Luego se realiza un proceso de contracción de la malla, donde los mejores nodos resultantes en cada iteración son seleccionados como malla inicial para la iteración siguiente. La formulación general de la metaheurística abarca tanto los problemas de optimización continuos como los discretos.

Descripción general de la metaheurística

La esencia del método DMO es crear una malla de puntos en el espacio m dimensional, donde se realiza el proceso de optimización de una función $FO(x_1, x_2, \dots, x_m)$; la cual se mueve mediante un proceso de expansión hacia otras regiones del espacio de búsqueda. Dicha malla se hace más "fina" en aquellas zonas que parecen ser más promisorias. Es dinámica en el sentido que la malla cambia su tamaño (cantidad de

²⁷ Resultado de evaluar la función objetivo del problema en cuestión.

nodos) y configuración durante el proceso de búsqueda. Los nodos se representan como vectores de la forma $n(x_1, x_2, \dots, x_m)$.

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

1. Generación de la malla inicial.
2. Generación de nodos en dirección a los extremos locales (nl).
3. Generación de nodos en dirección al extremo global (ng).
4. Generación de nodos a partir de las fronteras de la malla (nf).

El método incluye los parámetros:

- Cantidad de nodos de la malla inicial (N_i).
- Cantidad máxima de nodos de la malla en cada ciclo (N), donde $3 \cdot N_i \leq N$.
- Tamaño de la vecindad (k).
- Condición de parada (M).

A continuación, se presenta una descripción más detallada de cada uno de los pasos de DMO en el proceso de generación de nuevos nodos (expansión):

Generación de la malla inicial en cada iteración: la malla inicial consta de N_i nodos, los cuales en la primera iteración son generados de forma aleatoria o por otro método que garantice obtener soluciones diversas. En las restantes iteraciones del método se realiza un proceso de

contracción de la malla, que se basa en una selección de los nodos con mejor calidad entre los nodos (N nodos) existentes al final de cada iteración.

Generación de nodos en dirección a los extremos locales: el primer tipo de exploración que se realiza en DMO se lleva a cabo en las vecindades de cada uno de los nodos de la malla inicial. Para lo cual, se buscan los vecinos más cercanos de cada nodo n a través de una función de distancia o semejanza, en dependencia del dominio de cada dimensión²⁸. Luego, se selecciona cuál de los vecinos tiene mejor calidad (evaluación de la FO) que el nodo actual (denotándose ese mejor nodo por nl). Si ninguno de los vecinos es mejor, entonces este se considera un extremo local y no se generan nodos a partir de él en este paso. En otro caso, se genera un nodo (n^*) que estará situado entre el nodo (n) y el extremo local (nl).

La cercanía del nuevo nodo al actual o al extremo local depende de un factor (r), calculado en base a los valores que alcanza la FO en cada uno de los nodos involucrados.

Luego se calculan los valores de las componentes del nuevo nodo usando la ecuación:

$$n^*(i) = f(n(i), nl(i), r) \quad (0.1)$$

Donde i representa el i -ésimo componente de cada nodo. La función f depende totalmente del dominio de cada una de las dimensiones del problema, por lo que se puede presentar como una función de selección

²⁸ Representa la cantidad de variables a valorizar en el proceso de optimización.

de valores para casos discretos o como una aproximación para casos continuos. Mientras mayor sea la diferencia entre los valores de FO en los nodos involucrados, mayor será la cercanía o semejanza de n^* a nl , esto lo garantiza el factor r^{29} . Este paso se encarga de realizar una intensificación de la búsqueda en la vecindad de cada nodo.

Generación de nodos en dirección hacia el extremo global: Este paso tiene como propósito realizar una exploración global hacia el nodo que mejor calidad ha tenido hasta el momento (extremo global, ng); para lo cual, se generan nuevos nodos a partir de cada nodo de la malla inicial en dirección a este utilizando la ecuación (0.2):

$$n^*(i) = g(n(i), ng(i), r) \quad (0.2)$$

Al igual que en el paso anterior, g es una función que está totalmente relacionada con el dominio de cada dimensión y garantiza que mientras mayor sea la diferencia entre la calidad de cada nodo involucrado (determinado por r) mayor será la cercanía del nuevo nodo al extremo global. Este paso es el encargado de acelerar la convergencia del método.

Generación de nodos a partir de los nodos más externos de la malla: este proceso de generación de nuevos nodos tiene lugar con el objetivo de explorar el espacio de búsqueda en dirección a las fronteras de cada dimensión. Para ello, se seleccionan los nodos cuyas posiciones se encuentran en los extremos de la malla inicial (nodos fronteros). El

²⁹ No representa el mismo parámetro introducido en TS-ACO. En este contexto se utiliza para obtener la relación entre dos zonas del espacio de búsqueda, definidas por las soluciones involucradas.

proceso de detección de este tipo de nodo (nf) se realiza siguiendo algún criterio en dependencia del espacio solución. La generación de los nuevos nodos se obtiene a través de la ecuación:

$$n^*(i) = h(nf(i), w) \quad (0.3)$$

donde w se conoce como desplazamiento y tiene como objetivo desplazar los nodos fronteras en dirección a los puntos más y menos externos del espacio de soluciones (depende de la definición de las fronteras del problema).

Nótese que en este paso se seleccionan tantos nodos externos como sean necesarios para completar el tamaño de la malla en el ciclo; en principio, se debe tener ($N > 3 * Ni$) para garantizar que se generen algunos nodos en este paso.

Como se puede apreciar, este paso explora en los entornos de los nodos que definen las fronteras de la malla en cada iteración y es utilizado para ampliar la búsqueda hacia las fronteras del espacio de búsqueda.

Fomento de la diversidad

Se conoce que la diversidad en los métodos poblacionales es un elemento importante para realizar una buena exploración del espacio de búsqueda. Los investigadores han planteado distintas estrategias en tal sentido (Epígrafe 1.1). En este trabajo se presenta un mecanismo para fomentar la diversidad, que se basa en mantener una cierta separabilidad entre cada nodo de la malla inicial. Para ello, se propone un operador de limpieza (clearing) adaptativo, que funciona de la siguiente manera:

-
1. Se ordenan todos los nodos seleccionados como malla inicial en función de su calidad.
 2. De forma secuencial, se compara cada nodo de la malla con sus sucesores, eliminando aquellos cuya distancia espacial sea menor que una cota calculada dinámicamente. Este valor de la distancia debe permitir que el proceso sea decreciente; de manera que se obtenga mayor separabilidad entre los nodos al inicio que al final de la ejecución del método.
 3. Luego, se completa la malla inicial (de ser necesario) con nodos generados de forma aleatoria.

Este mecanismo mezcla dos elementos importantes, tales como: la calidad de los nodos con la posición que ocupan en el espacio de solución. Esto provoca que los nodos con mejor calidad tengan mayor probabilidad de formar parte de la próxima malla inicial.

El valor adaptativo de la cota de distancia (cd) le permite al método comenzar con exploraciones más generales y luego ir disminuyendo su influencia hasta centrarse en una zona más pequeña del espacio de búsqueda. Este elemento aumenta el nivel de explotación del método y lo hace más robusto.

Un mecanismo parecido fue propuesto en [37], donde el algoritmo evolutivo CHC cruza sólo aquellos padres seleccionados que respeten una determinada separabilidad entre ellos; esta estrategia es conocida como *prevención de incestos*. Se evidencia claramente que en ningún momento se garantiza que todos los individuos seleccionados están a una determinada distancia en el espacio solución, y mucho menos que

se realice un proceso adaptativo de la mínima distancia permitida entre padres.

Caracterización y novedades del método

Luego de tener bien definido cada uno de los procesos del método DMO, se presenta en la Ilustración 9 la estructura general del mismo, donde la condición de parada está descrita por un número máximo de evaluaciones de la función objetivo.

Generación de la malla inicial (N_i) de forma aleatoria.
Evaluar los nodos de la malla inicial, seleccionar el mejor ng .

Repetir:

Para cada nodo en la malla inicial del ciclo **hacer**

Encontrar sus k nodos más cercanos.
Determinar el mejor de los vecinos ne .

Si ne es mejor que el nodo actual, **entonces**

Generar nuevo nodo usando ecuación (0.1).

Fin Si

Fin Para

Para cada nodo en la malla inicial del ciclo **hacer**

Generar un nuevo nodo usando la ecuación (0.2).

Fin Para

Seleccionar los nodos más externos de la malla.
Generar nuevo nodo usando la ecuación (0.3).

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

Generar nuevo nodo aleatoriamente.
Construir la malla inicial actual de manera elitista.

Ordenar los nodos de la malla inicial según su calidad.

Aplicar operador de limpieza adaptativo.

Hasta M evaluaciones

Ilustración 9: Funcionamiento general de DMO.

Esta nueva metaheurística poblacional tiene características similares con otros métodos existentes como, por ejemplo:

1. El considerar solamente los vecinos más cercanos en la valoración de los extremos locales fue introducido en el método PSO, como se reporta en [103]; donde al determinar la mejor partícula global se toman en cuenta dos enfoques: considerar el mejor de toda la población o la mejor partícula entre los vecinos.
2. La atracción a zonas más promisorias del espacio de búsqueda con el acercamiento al extremo global fue introducido también en PSO; donde en su versión original [23], cada partícula es atraída por la mejor posición global del enjambre.
3. La concepción de expansión y contracción de la malla inicial es una forma de aumentar la población inicial con la incorporación de nuevas soluciones y luego reducirla a través de un proceso de selección. Este elemento ha sido bien estudiado en los Algoritmos Evolutivos Generacionales [22].
4. La selección de la malla inicial de manera elitista también fue introducida en los Algoritmos Genéticos como estrategia para acelerar la convergencia.

A pesar de compartir algunas características con otros métodos existentes, DMO incorpora otros elementos propios, como son:

1. Utilizar en una misma iteración la atracción hacia los extremos locales de cada vecindad y el extremo global de la población. Esto representa una nueva forma de realizar intensificación y diversificación manteniendo la dirección de la búsqueda.
2. Se realiza una selección elitista con diversidad, en la que se tiene en cuenta tanto la calidad como la separabilidad entre soluciones al mismo tiempo. Este elemento garantiza que el método realice una profunda exploración del espacio solución, disminuyendo en gran medida el estancamiento de soluciones.
3. Se introduce un proceso de limpieza adaptativo, donde la distancia que garantiza la separabilidad de las soluciones es un valor que decrece en función del estado del método. Permitiendo al inicio del algoritmo soluciones más distantes que al final de la ejecución.
4. Se incorpora una forma para guiar la exploración hacia los entornos de espacio de búsqueda a través de la generación de nuevos nodos a partir de los nodos fronteras de la malla. De esta manera se aprovecha la posición que ocupan estos nodos, explorando fuera de los entornos donde se está realizando la búsqueda.

Caso de estudio: Aproximación de funciones continuas

En este epígrafe se definen con mayor precisión cada uno de los operadores de DMO para dominios continuos [104, 105]; la modelación del caso discreto fue presentada en [106] para el problema de Selección de Rasgos. Como problema de prueba para estudiar el caso continuo, se presenta la aproximación (minimización) de un conjunto de 20 funciones multimodales, publicadas como marco experimental obligatorio de una sección especial organizada por la IEEE en el Congress on Evolutionary Computation (CEC 2005) celebrado en Londres y, desde entonces, de obligada utilización para el estudio actual del comportamiento de las metaheurísticas para dominios continuos. Una amplia descripción de todas las funciones³⁰ se puede encontrar en [107], así como el código fuente para los investigadores de acceso libre³¹.

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, $x \in [-32,32]^D$

³⁰ La enumeración de cada función coincide exactamente con la reportada en la referencia correspondiente.

³¹ Accesible desde <http://www.ntu.edu.sg/home/EPNSugan/>

-
- ✓ 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.

Con esta experimentación se estudia el comportamiento de algunos elementos importantes del algoritmo. A continuación, se describen cada uno de los operadores para el caso del estudio continuo.

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

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

$$D_{euclidiana}(n_1, n_2) = \sqrt{\sum_{i=1}^n (n_1(i) - n_2(i))^2} \quad (0.4)$$

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

$$r = \frac{1}{1 + |FO(n) - FO(ne)|} \quad (0.5)$$

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

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

$$n^*(i) = \begin{cases} vm(i), & \text{si } |vm(i) - ne(i)| > cd \text{ y } va[0, 1] \leq r \\ ne(i) + va[-cd, cd], & \text{si } |vm(i) - ne(i)| \leq cd \\ va[vm(i), ne(i)], & \text{en otro caso} \end{cases} \quad (0.6)$$

donde $vm(i)$ representa el valor medio entre el nodo actual y el extremo local para la i -ésima dimensión y se calcula como:

$$vm(i) = \frac{n(i) + ne(i)}{2} \quad (0.7)$$

Además, $va[a, b]$ denota un valor aleatorio en el intervalo $[a, b]$ y cd es una cota de distancia adaptativa y se calcula según la ecuación (0.8)

$$cd = \begin{cases} \frac{|inf + sup|}{4}, & \text{si } j < 15\%M \\ \frac{|inf + sup|}{8}, & \text{si } 15\%M \leq j < 30\%M \\ \frac{|inf + sup|}{16}, & \text{si } 30\%M \leq j < 60\%M \\ \frac{|inf + sup|}{50}, & \text{si } 60\%M \leq j < 80\%M \\ \frac{|inf + sup|}{100}, & \text{si } j \geq 80\%M \end{cases} \quad (0.8)$$

donde M denota el valor máximo de evaluaciones de la función objetivo y j la evaluación actual. En dependencia del por ciento del total que represente la evaluación actual, se definen valores de distancia que representan partes del intervalo permitido.

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

permitida y, en el último caso, se genera un número aleatorio entre el valor medio y el extremo local.

Generación de nodos en dirección al extremo global: en este paso de la generación se crean nuevos nodos a partir de cada nodo de la malla inicial en dirección al extremo global. Para esto, se utiliza el valor de r calculado por la ecuación (0.5) sustituyendo n_e por n_g . Luego se define la función de generación g :

$$n^*(i) = \begin{cases} vm(i), & \text{si } va[0,1] \leq r \\ va[vm(i), ng(i)] & \text{en otro caso} \end{cases} \quad (0.9)$$

donde el valor medio (vm) entre el nodo actual y el extremo global se calcula utilizando la ecuación (0.7).

Según la función g , si existe una gran diferencia entre la calidad del extremo global y el nodo actual, hay mayor probabilidad de que la i -ésima componente tome valores más cercanos a ng , en caso contrario se valoriza con el valor medio.

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

$$\|n\| = \sqrt{\sum_{i=1}^n (n(i))^2} \quad (0.10)$$

Los nodos de mayor norma son los que están situados en el contorno (puntos más externos) de la malla inicial y los de menor norma se consideran los nodos que más cerca se encuentran del origen 0^m (puntos más internos)³². La función h permite generar nuevos nodos en dirección de las fronteras definidas para este caso de estudio, mediante las expresiones:

Para los nodos más externos:

$$n^*(i) = \begin{cases} nl(i) + w, & \text{si } nl(i) > 0 \\ nl(i) - w, & \text{si } nl(i) < 0 \end{cases} \quad (0.11)$$

Para los nodos más internos:

$$n^*(i) = \begin{cases} |nl(i) - w|, & \text{si } nl(i) > 0 \\ |nl(i) + w|, & \text{si } nl(i) \leq 0 \end{cases} \quad (0.12)$$

donde el desplazamiento w se calcula como:

$$w = (w_0 - w_f) \cdot \frac{M - j}{M} + w_f \quad (0.13)$$

donde el parámetro M y la variable j , están estrechamente relacionados y provocan las variaciones en el valor de w ; el primero representa el número total de iteraciones del algoritmo o un número máximo de evaluaciones de la función objetivo (criterio más utilizado). Por su parte, j

³² La utilización del origen como referencia para determinar los puntos más internos fue elegida debido a un conocimiento previo del problema, donde en 18 funciones de las 20, este punto representa el centro del espacio de búsqueda.

denota el valor actual de uno de los criterios mencionados anteriormente. La variable w_0 representa el desplazamiento inicial y w_f el valor final de este (si $w_0 > w_f$ efecto decreciente). Para obtener desplazamientos decrecientes relacionados con las amplitudes de cada función, se desarrolló una propuesta adaptativa, donde $w_0 = am/10$ y $w_f = am/100$. Por su parte am denota la amplitud media del intervalo permitido para la i -ésima componente y se calcula como:

$$am = \frac{|inf| + |sup|}{2} \quad (0.14)$$

Según las ecuaciones utilizadas para la función h se puede dar el caso que el valor de alguna de las componentes no esté dentro del intervalo permitido $[inf, sup]$. Entonces se corrige utilizando la siguiente regla:

$$n^*(i) = \begin{cases} va[inf, vm], & si \ n^*(i) < inf \\ va[vm, sup], & si \ n^*(i) > sup \end{cases} \quad (0.15)$$

donde vm representa el valor medio del intervalo permitido para la componente i y se calcula utilizando la ecuación (0.7).

Análisis interno del método

En este epígrafe se realiza un estudio del comportamiento de algunos de los elementos involucrados en el proceso de exploración de DMO en la solución de las 19 funciones multimodales presentadas anteriormente.

Los experimentos serán descritos de la siguiente manera:

-
1. Tamaño de la población: se experimenta con distintos tamaños de malla para probar si tiene influencia en la calidad de los resultados.
 2. Operador de limpieza: se estudia el comportamiento del método utilizando o no la limpieza adaptativa y algunas variantes con valores constantes.
 3. Expansión hacia las fronteras: se comparan los resultados del método utilizando el operador de exploración a través de los nodos fronteras y sin utilizarlo; es decir, usando la función h y sin usarla.

Todos los experimentos se realizaron para dimensión 10 de las funciones involucradas, con un número máximo de evaluaciones de la FO de 100 000. Todos los resultados fueron computados como promedio sobre 25 corridas independientes.

Tamaño de la malla

En las metaheurísticas poblacionales es muy importante la definición del tamaño de las poblaciones para obtener un buen desempeño de estos métodos. Por ejemplo, en los GA las poblaciones deben de ser grandes [108]; debido a que no todos los individuos influyen en la evolución de la población, en cada iteración, se realiza un proceso de selección que obtiene solo un subconjunto de esta para generar nuevos individuos. En el caso de los algoritmos PSO, ocurre lo contrario; en ellos se recomienda utilizar poblaciones pequeñas [20] debido a que todos los agentes cambian de posición, influyendo directamente en el desplazamiento de la bandada.

El proceso presentado por DMO es muy parecido al de PSO, donde todos los nodos de la malla inicial intervienen en la exploración. Debido a esta similitud, se estudian distintos tamaños de malla inicial (8, 12, 24, 50). Para todos los casos, el tamaño total de la expansión es $N = 3Ni + Ni/2$, con el objetivo de generar con el operador de las fronteras una cantidad de nodos, al menos, igual a la mitad de la malla inicial. En este estudio se utilizan todos los operadores de DMO tal y como se formularon anteriormente.

A continuación, se muestran los resultados del test de Iman-Davenport con 2 y 38 grados de libertad de la distribución F , donde se puede apreciar que existe alguna diferencia entre los algoritmos del grupo de control; debido a esto la hipótesis de igualdad es rechazada.

Tabla 12: Resultados del test de Iman-Davenport para los distintos tamaños de malla.

<i>Método</i>	<i>Valor del Test</i>	<i>Valor de distribución F</i>	<i>Valor p</i>	<i>Hipótesis</i>
Iman-Davenport	27,580	3,2448	6,82E-9	R

Seguidamente se aplica una prueba de comparaciones múltiples para detectar donde están las diferencias encontradas en la tabla anterior.

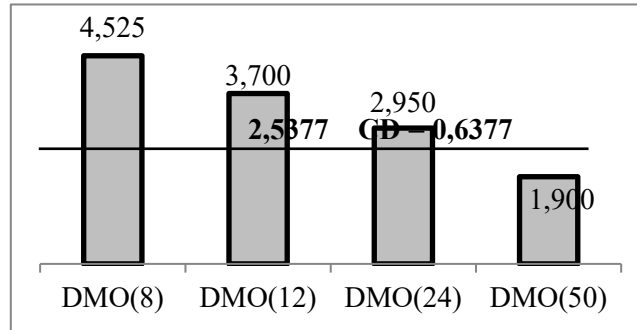


Ilustración 10: Resultados del test de Bonferroni-Dunn para el tamaño inicial de la malla en DMO.

Según los resultados del test de Bonferroni-Dunn, mostrados en la Ilustración 10, los resultados para un tamaño de malla inicial igual a cincuenta (DMO(50)) son significativamente superiores que para las otras configuraciones realizadas. Esta conclusión se debe a que el orden medio de todas las variantes computadas están por encima del patrón de comparación (2,5377) definido para el valor crítico (CD) reportado por la configuración de control (DMO(50)). En lo adelante se utiliza este tamaño de malla inicial para los estudios siguientes.

Operador de limpieza adaptativo

Las columnas DMO-SC representan las soluciones del algoritmo sin el operador de limpieza y DMO-CA utilizando la limpieza adaptativa. Por otra parte, DMO-C1, DMO-C2, DMO-C3, DMO-C4 y DMO-C5 muestran las soluciones de DMO con el operador de limpieza para distintos valores constantes de la distancia, que representan una parte (1/4, 1/8, 1/16, 1/50 y 1/100, respectivamente) de la amplitud del dominio en cada dimensión (ecuación (0.8)).

Tabla 13: Resultados del test de Iman-Davenport para las distintas alternativas de limpieza.

Método	Valor del Test	Valor de distribución	Valor p	Hipótesis
Iman-Davenport	26,315	3,2448	1,79E-19	R

Como se muestra en la **¡Error! No se encuentra el origen de la referencia.**, el estadístico de Iman-Davenport rechaza la hipótesis de igualdad, debido a que existen diferencias significativas en el grupo de algoritmos involucrados. Por este motivo se aplica el test de comparaciones múltiples Bonferoni-Dunn.

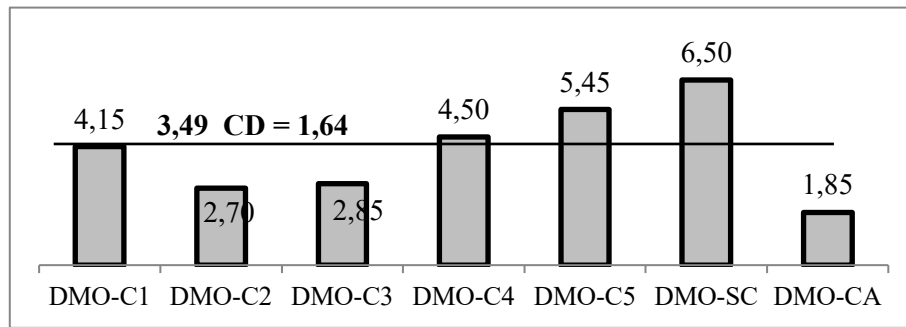


Ilustración 11: Resultado del test de Bonferoni-Dunn para las distintas alternativas de limpieza.

La figura anterior muestra que el algoritmo de control (DMO-CA) obtiene resultados significativamente superiores a todos los algoritmos que se encuentran por encima de la recta que representa el patrón de comparación (3,49) para el valor crítico (CD=1,64). Con respecto a los algoritmos DMO-C2 y DMO-C3 no se encuentran diferencias significativas. Para verificar este resultado se aplica un test de Holm entre los tres algoritmos estadísticamente similares, con DMO-CA como algoritmo de control (menor orden medio).

En la Tabla 14 se evidencia claramente que, en todos los casos de comparaciones múltiples, el algoritmo de control es superior estadísticamente para un valor de significación de 0,05. Esto se debe a que el *valor p* de cada comparación es menor que el valor *a/i* correspondiente, por este motivo, la hipótesis de igualdad es rechazada en cada comparación.

Tabla 14: Resultados del test de Holm, DMO-CA como algoritmo de control.

Algoritmo	Z	Valor p	a/i	Hipótesis
DMO-C3	2,2135	0,0248	0,025	R
DMO-C2	2,0554	0,0398	0,05	R

De esta manera se puede resumir que la utilización del operador de limpieza adaptativo le brinda ventajas al método DMO. En lo adelante, se utiliza para realizar los estudios restantes.

Generación mediante los nodos fronteras

Este proceso de generación se encarga de explorar el espacio de solución en dirección a las fronteras definidas para el problema en cuestión. Por ello, es necesario realizar una experimentación para determinar si es aconsejable aplicarlo al conjunto de funciones de prueba.

Seguidamente se realiza el estudio estadístico de los resultados experimentales, donde la columna DMO-SF representa el algoritmo DMO sin utilizar el operador de fronteras y DMO-F para el caso en que sí lo utiliza.

Tabla 15: Resultados del test de Wilcoxon para el operador de las fronteras.

Algoritmo	R ⁺	R ⁻	Valor p	Hipótesis
DMO-F vs DMO-SF	46,00	164,00	0,028	R

Como se puede apreciar en la Tabla 15 la hipótesis de igualdad se rechaza a favor del algoritmo DMO-F, por lo que los resultados de DMO, utilizando la generación de nodos hacia la frontera, son significativamente mejores que la variante que no la utiliza (DMO-SF).

En lo adelante se utiliza el algoritmo DMO con el operador adaptativo de limpieza y el operador de generación de nodos a partir de las fronteras.

Estudio comparativo con otros métodos

Se realiza un estudio comparativo entre la metaheurística DMO y otras que han mostrado un alto desempeño en la actualidad. Estos métodos son buenos representantes de Algoritmos Evolutivos e Inteligencia Colectiva. En ningún caso se hace uso de algoritmos de búsqueda local para mejorar las soluciones. A continuación, se presenta una breve descripción de estos métodos:

- Algoritmo Cross generational elitist selection, Heterogeneous recombination, and Cataclysmic mutation (CHC) con codificación real [41]. Es un algoritmo muy competitivo tanto en codificación binaria como real. Todos los parámetros son los propuestos por los propios autores, con cruce BLX-0,5 [40].
- Algoritmo Steady-State Genetic Algorithm (SSGA) [109], con selección de padres Negative Assortative Mating [110] [111], con $N_{nam}=3$, y estrategia de reemplazo del peor (Replace Worst). Este

algoritmo presenta un adecuado comportamiento de exploración y explotación.

- Algoritmo Linearly Decreasing Inertia Weigth in Particle Swarm Optimization (LDWPSO) [112]. Se considera el algoritmo PSO propuesto por Shi y Ebenhart aplicando la configuración de los autores: la inercia varía desde $w_{max}=0,9$ hasta $w_{min}=0,4$; y los parámetros c_1 y c_2 valen 2,8 y 1,3 respectivamente.
- Algoritmo Opposite Differential Evolution (ODE) [110]. Este algoritmo es un algoritmo del estado del arte en codificación continua en el que se combina un algoritmo de Evolución Diferencial [47] con una técnica de opuestos para fomentar la diversidad.

En todos los casos se utilizaron los parámetros definidos por los autores.

Una propiedad muy estudiada en las metaheurísticas es la escalabilidad. Esta propiedad garantiza que si se aumenta la dimensión de las instancias el método sigue siendo competitivo. Para probar el comportamiento de DMO en problemas escalables se utilizan las mismas funciones de test pero con dimensiones 30 y 50, con 300 000 y 500 000 evaluaciones de la función objetivo respectivamente como condición de parada.

A continuación, se presenta el análisis estadístico realizado sobre las Tablas de resultados.

Tabla 16: Test de Wilcoxon para comparaciones de DMO en dimensión 10.

Algoritmo	R ⁺	R ⁻	Valor p	Hipótesis
SSGA vs DMO	210	0	0,000	R
CHC vs DMO	181	29	0,004	R
ODE vs DMO	89	121	0,55	A
LDWPSO vs DMO	210	0	0,000	R

Tabla 17: Test de Wilcoxon para comparaciones de DMO en dimensión 30.

Algoritmo	R ⁺	R ⁻	Valor p	Hipótesis
SSGA vs DMO	210	0	0,000	R
CHC vs DMO	143	67	0,159	A
ODE vs DMO	135	75	0,263	A
LDWPSO vs DMO	154	56	0,000	R

Tabla 18: Test de Wilcoxon para comparaciones de DMO en dimensión 50.

Algoritmo	R ⁺	R ⁻	Valor p	Hipótesis
SSGA vs DMO	210	0	0,000	R
CHC vs DMO	157	53	0,042	R
ODE vs DMO	200	10	0,000	R
LDWPSO vs DMO	194	16	0,001	R

Según los resultados del test de Wilcoxon (Tabla 16, Tabla 17 y Tabla 18) computado para cada caso, se puede observar que:

- Para dimensión 10, DMO es significativamente mejor que SSGA, CHC y LDWPSO. Sólo es peor en términos absolutos respecto a ODE, pero no de una forma significativa.
- Para dimensión 30, DMO es mejor que todas las demás alternativas consideradas. En el caso específico de SSGA y LDWPSO de forma significativa, la diferencia con respecto a CHC disminuyó con relación a la dimensión anterior.

-
- Por último, para la dimensión 50 los resultados obtenidos por DMO son significativamente mejores que los de todas las propuestas consideradas. El algoritmo que más se acerca a los resultados de DMO es el CHC.

En términos de escalabilidad se puede observar que el método DMO presenta un alto nivel de aplicabilidad. Esto se evidencia claramente en las conclusiones anteriores; donde para dimensión 10 obtiene peores resultados que ODE; ya para dimensión 30 supera a todas las demás, aunque en algunos casos no de forma significativa. Y por último, para la dimensión 50 sus resultados superan significativamente a todas las propuestas implementadas.

Conclusiones parciales

En este capítulo se introdujo una metaheurística poblacional llamada Optimización Basada en Mallas Dinámicas (denotada por DMO), la cual incorpora nuevas formas para explorar el espacio de búsqueda. Esto lo diferencia de otros métodos poblacionales en que:

- Utiliza en una misma iteración del algoritmo la atracción hacia los extremos locales de cada vecindad y hacia el extremo global obtenido hasta el momento. Esto provoca un equilibrio entre explotación y exploración del espacio de búsqueda.
- Se realiza una selección de la malla inicial en cada iteración de manera elitista en cuanto a calidad, teniendo en cuenta la separabilidad entre los nodos a través de un operador de limpieza. De esta manera, se garantiza mantener una diversidad

controlada en la población que facilite una mayor exploración del espacio de solución.

- El operador de limpieza funciona de manera adaptativa; ya que decrece el valor de la distancia permitida entre nodos a medida que se ejecuta el algoritmo, en dependencia de la amplitud de cada intervalo. Este funcionamiento provoca que el método comience con un nivel alto de exploración, el cual decrece a medida que disminuye la distancia permitida con la ejecución del algoritmo. Lo contrario sucede con el nivel de explotación del método.
- Se introduce una forma para explorar en dirección a las fronteras del espacio de solución. Este proceso ocurre mediante el operador de generación a través de los nodos más externos y menos externos de la malla. Los resultados experimentales muestran que esta alternativa de generación de nodos favorece la exploración para las funciones estudiadas, por lo que resulta necesario definir de forma apropiada el significado de fronteras para un problema determinado.

Se probó que el algoritmo DMO presenta un alto nivel de escalabilidad, obteniéndose con el mismo resultados superiores de manera general a los demás algoritmos del estado del arte involucrados en la comparación.

CAPÍTULO 4: SOLUCIÓN AL PROBLEMA DE SELECCIÓN Y ESCALAMIENTO DE MOVIMIENTOS SÍSMICOS FUERTES

En la actualidad se realizan grandes esfuerzos por aplicar métodos, cuya aplicabilidad ha sido probada experimentalmente, en la solución de problemas de la vida real. Así, es posible alcanzar soluciones más eficaces y eficientes en un entorno cada vez más exigente y competitivo.

Otro elemento que motiva el creciente interés de los investigadores en resolver problemas reales de optimización está dado en que la modelación y solución de estos usualmente plantea retos nuevos, no considerados antes en los estudios teóricos; de modo que se hace necesario realizar modificaciones o adaptaciones a los métodos existentes o desarrollar otras alternativas que se adecuen mejor a las características de los problemas a resolver.

Este es el caso del estudio que se presenta en este capítulo, donde se desarrolla la aplicación de los métodos propuestos a un problema real en el área de la Sismología.

Descripción del dominio de aplicación

La Ingeniería Sísmica basada en niveles de desempeño exige la utilización cada vez más frecuente de los procedimientos avanzados de análisis de estructuras con historias de tiempo del movimiento del suelo (acelerogramas) para la estimación estadística de la respuesta sísmica de estructuras [113, 114]. En el transcurso de la humanidad han ocurrido grandes desastres causados por fuertes terremotos; la principal causa de pérdidas humanas está relacionada con el nivel de resistencia de las construcciones en zonas activas. En la actualidad existen diferentes

herramientas computacionales que simulan el desempeño de distintas edificaciones frente a terremotos de laboratorio [115]. Para garantizar la confiabilidad de la predicción de la respuesta y del diseño estructural, se hace necesario utilizar un conjunto de historias de tiempo del movimiento del suelo reales que sean compatibles con el nivel de amenaza sísmica de la región de estudio [116]. Esta compatibilidad puede ser lograda utilizando ya sea registros sísmicos sintéticos o reales, cuyas ordenadas espectrales de respuesta “iguales” a las del espectro de diseño del suelo, amortiguado en un 5%, del suelo dentro de un margen de error aceptable para el diseñador o estipulado por el código de diseño, en el rango de períodos estructurales considerado.

El **espectro de respuesta** no es más que la gráfica construida a partir del máximo valor de la respuesta de un conjunto de osciladores (suelos, edificios, puentes, etc.) que vibran con un período determinado, al ser sometidos a una excitación sísmica. La unión de los n puntos con sus respectivos períodos forma una gráfica. Estas respuestas se pueden expresar en términos de desplazamiento relativo, velocidad relativa y aceleración absoluta.

El **amortiguamiento** expresa la atenuación de la amplitud de la onda sísmica en el tiempo. Esta pérdida o transformación de energía cinética está asociada a diferentes mecanismos que se oponen al movimiento; entre ellos la viscosidad, la fricción ó la histéresis. Un ejemplo más concreto es el de una onda sísmica atravesando por diferentes medios (estratos de suelo), que varían de acuerdo con sus propiedades mecánicas. El amortiguamiento se suele expresar como un porcentaje (%) del amortiguamiento crítico bajo suposición elástica del medio. En el

estudio que se presenta aquí se considera como valor de amortiguamiento el 5%.

Espectro de diseño: Debido al carácter aleatorio de los terremotos y de sus espectros de respuesta, los espectros de diseño se determinan a partir de estudios estadísticos de familias de movimientos sísmicos. Mediante los espectros suavizados se puede obviar la necesidad de hacer cálculos de edificios para cada uno de los movimientos sísmicos que pudieran ocurrir en una región de interés

Para realizar el estudio de resistencias de las estructuras se pueden emplear lo mismo historias de tiempo generadas artificialmente, que acelerogramas de sismos reales ocurridos en otros escenarios siempre y cuando se puedan aproximar, lo más posible, al acelerograma que se espera en el lugar en términos de magnitud, distancia epicentral, la profundidad, el mecanismo focal y el perfil de suelo. Sin embargo, en una región dada, no siempre es posible identificar todos estos parámetros al definir un sismo de diseño y tampoco es posible encontrar acelerogramas que cumplan todas las condiciones deseadas, especialmente en aquellas zonas donde la recurrencia de sismos fuertes es muy baja como es el caso de Cuba. Lo anterior justifica la necesidad de realizar ciertas manipulaciones de registros sísmicos originales, con el objetivo de ajustarlos para reproducir las características que se desean en nuestro país. Esta es la base sobre la que descansa el surgimiento de diferentes métodos para la generación de terremotos compatibles con la amenaza del sitio y del escalamiento de historias de tiempo de terremotos fuertes reales para hacerlos compatibles con los espectros de diseño de un área determinada [117].

Problema de Selección y Escalamiento de movimiento sísmicos fuertes

De forma general el Problema de Selección y Escalamiento de registros de movimientos sísmicos fuertes (PSE) se perfila como un problema sobre un dominio discreto, que consiste en seleccionar dentro de un conjunto de terremotos aquellos cuyo espectro respuesta promedio se aproxime lo más posible a un espectro objetivo (espectro de diseño para diferentes perfiles de suelo) dado un nivel de amortiguamiento. Conjuntamente con la selección se realiza un proceso de escalamiento, el cual garantiza la obtención de amplitudes de ordenadas espectrales compatibles con las que producirían sismos probables de registrar en Cuba.

Proceso de Selección de registros de sismos fuertes

En Cuba se realizan actualmente análisis de peligro sísmico probabilísticos convencionales con base en una escala de intensidad macrosísmica; debido a la falta de registros de terremotos fuertes y también de recursos económicos y tecnológicos para la ejecución de estudios detallados de las características del movimiento proveniente de cada fuente sísmica por separado. Por estas razones el espectro de diseño elástico de la actual norma cubana de cálculo sismo resistente, NC 46: 1999, se ha estimado como la envolvente suavizada de todos los posibles espectros de respuesta de sismos de varias fuentes sismogénicas internacionales (Peligro Uniforme) que ejercen influencia sobre un sitio o región considerando, ya sea movimientos sintéticos o registros de movimientos de señales débiles escaladas. La dificultad de esta aproximación radica en que se pueden perder aspectos esenciales en la estimación de daño potencial a la resistencia y la estabilidad de las edificaciones. No obstante, existe una tendencia a nivel internacional de realizar análisis de

peligro sísmico probabilístico más robusto como base para el diseño por confiabilidad de las edificaciones.

Internacionalmente existen distintas fuentes de datos de registros sísmicos fuertes confiables, las cuales contienen gran cantidad de terremotos ocurridos a lo largo de la historia. Estas bases de casos son una fuente muy rica de recursos para la búsqueda de fenómenos que pudieran ocurrir según nuestras características de fuentes, trayectorias y suelos.

En otros trabajos desarrollados previamente empleando Algoritmos Genéticos [118], se han desarrollado técnicas para reducir el espacio de búsqueda de este problema como por ejemplo: realizar un agrupamiento previo de las formas espectrales de los tipos de movimientos sísmicos, utilizando técnicas de agrupamiento. De manera que todos los terremotos dentro de cada clúster cumplen con las condiciones para formar parte de una solución. Esto está dado por la ausencia de un criterio físico específico en la determinación del factor de escalado (SI), ya que este factor es un simple valor escalar que se aplica a las ordenadas espectrales aumentando o disminuyendo sus amplitudes, dentro de un rango especificado por el ingeniero. En este caso, el SI, no es un valor escogido al azar sino un parámetro calculado a partir del propio espectro de respuesta, que expresa el contenido energético dentro de un rango de frecuencias, por lo que todos los terremotos de la base de casos pueden ser candidatos para formar parte de una solución. Este valor se puede utilizar para obtener un efecto parecido al agrupamiento, pero con un sentido físico, para que la búsqueda se centre solo en los terremotos cuyos factores de escalado de los espectros respuestas, estén en un rango definido por el ingeniero estructural (ecuación (0.1)).

$$S_{\min} \leq SI \leq S_{\max}, \quad S_{\min}, S_{\max} > 0 \quad (0.1)$$

Donde S_{\min} y S_{\max} representan el rango permitido para los factores de escalado.

Cálculo del error específico

El error calcula cuánto se acerca el promedio que forma una solución al espectro objetivo para un tipo de suelo (sitio) específico. Una solución será mejor que otra si tiene un menor error específico (EE) con respecto a un espectro objetivo, siendo el error la función a minimizar en el problema y definido de la siguiente manera:

$$EE = \sum_{t=t_0}^{t_m} \left(\sqrt{\frac{\sum_{i=1}^k [SI_i \cdot SA_i(t)]^2}{\sum_{i=1}^k SI_i^2}} - F(t) \right) \quad (0.2)$$

donde t es el período de vibración fundamental de la estructura, SI_i el factor de escalado genérico correspondiente al registro número i , $SA_i(t)$ el valor de la aceleración espectral del registro sísmico número i en el período t , $F(t)$ el valor del espectro de diseño objetivo (para un sitio dado) en el período t , t_0 el período inicial a considerar (Ejemplo: $t_0 = 0,2$) y t_m el período final a considerar (Ejemplo: $t_m = 1,5$).

La definición formal del problema es por tanto: la búsqueda del error mínimo específico entre el espectro medio escalado del conjunto de k terremotos y el espectro objetivo, formulado a través de la función de minimización, en un período determinado.

Proceso de ajuste de los espectros de respuesta

Los factores de escalado obtenidos deben garantizar que la solución encontrada quede, respecto al espectro objetivo, dentro de ciertos límites especificados por el diseñador. Es por ello que el proceso de ajuste se realiza recorriendo la curva del espectro promedio, punto a punto, desde t_0 hasta t_m ; para detectar las áreas negativas.

El proceso de ajuste se realiza recorriendo la curva del espectro promedio, punto a punto, desde t_0 hasta t_m ; para detectar si en algunas zonas el espectro promedio quedó por debajo del objetivo; esto se logra, chequeando los valores del coeficiente c y se define mediante los pasos siguientes:

Paso 1: Calcular la relación de ajuste (c) punto a punto entre las ordenadas espectrales objetivo (OEO) y promedio (OEP).

$$c(t) = \frac{OEO(t)}{OEP(t)}, \quad t_0 \leq t \leq t_m \quad (0.3)$$

Paso 2: Calcular la constante de ajuste (C), como el valor máximo entre todas las relaciones $c(t) > 1$.

$$C = \max_{t \leftarrow t_0, t_m} c(t) \quad (0.4)$$

En términos sísmicos lo que se persigue con el proceso de ajuste es reconsiderar todos aquellos puntos donde las ordenadas espectrales que conforman una solución generen un espectro respuesta de menor intensidad que las estimadas por los estudios de peligro sísmico en nuestro país, por seguridad de las construcciones.

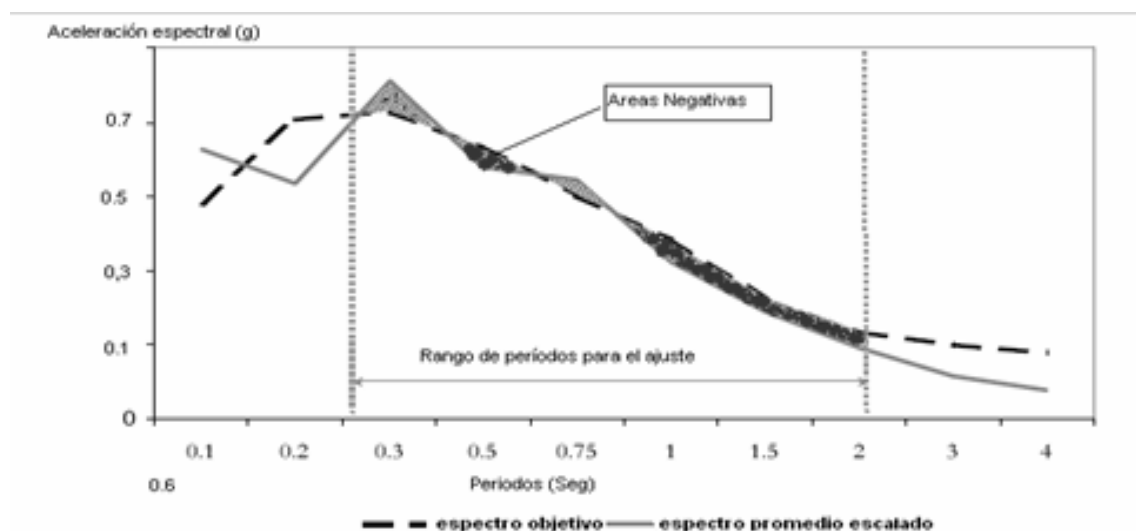


Ilustración 12: Representación de áreas negativas entre el espectro objetivo y el espectro promedio.

Proceso de Escalado

El objetivo de este proceso es modificar los movimientos sísmicos ocurridos en otros escenarios internacionales de manera que produzcan espectros respuestas compatibles con el peligro sísmico de nuestras zonas sísmicas y que puedan ser utilizados en el estudio de resistencia de las estructuras de acuerdo con la Norma Cubana.

El cálculo del factor de escalado está definido por el criterio de escalamiento que adopte el ingeniero estructural. Existen varios criterios de escalamiento dados por diferentes investigadores, cuyos estudios han demostrado el grado de correlación existente entre el índice del movimiento sísmico utilizado como parámetro escalable y la respuesta estructural (daño). En el software se implementan todos los tipos de escalados para una mayor aplicabilidad del mismo, pero según los expertos el escalado en términos de intensidad espectral [117] es el más

utilizado, debido a que solo se manipula la amplitud de la onda sísmica y no la frecuencia como la mayoría de los criterios existentes [119-122].

Criterio de escalamiento según la intensidad de la aceleración de respuesta: (A_{SI})

$$SI = \int_{t_0}^{t_m} SA(t, \vartheta) dt \quad (0.5)$$

donde SA representa el espectro de aceleración, para un determinado nivel de amortiguamiento 5 % de un espectro objetivo.

Estudio experimental

Por la necesidad en Cuba de realizar estudios de resistencia de estructuras en zonas activas y no contar con una aplicación que realice la selección de los terremotos utilizados para el análisis, se decidió crear una implementación computacional montada sobre software libre (JAVA). Para la elección del método heurístico que funciona como núcleo principal del software, se realizó un estudio experimental (el cual se muestra a continuación) que incluye los algoritmos propuestos en esta investigación (TS-ACO y DMO), así como otras metaheurísticas tales como la Optimización Basada en Colonia de Hormigas (ACO) y un método Voraz basado en la Búsqueda de los Vecinos más Cercanos (Greedy-Neighbor Near Search; G-NNS) [16].

Aplicación de la Optimización basada en Colonia de Hormigas en una y dos etapas

Para solucionar el problema PSE con las técnicas basadas en colonia de hormigas (ACO y TS-ACO), se realizó una modelación, en la cual cada

nodo del grafo representa un terremoto de los que componen la base de caso seleccionada, y los arcos, transiciones entre estos. De manera específica, todos los nodos están conectados con todos y sin dirección alguna.

Otros elementos para tener en cuenta para este problema son: el significado de los rastros de feromona y la función heurística, ambos elementos responsables de que los métodos realicen una buena exploración del espacio de solución. Los rastros de feromona son asociados a los nodos del grafo, ya que no interesa el orden en que fueron seleccionados los terremotos en el proceso de búsqueda, solo cuál fue seleccionado. En el software fueron implementadas varias funciones heurísticas propuestas por los especialistas con un sentido totalmente físico. A continuación, se describe la función utilizada para la validación de los resultados y la que aparece por defecto en la aplicación:

$$\eta_{ij} = \begin{cases} \infty & , \text{ si } SI_j = 1 \\ \frac{1}{|1 - SI_j|} & , \text{ si } SI_j \neq 1 \end{cases} \quad (0.6)$$

donde SI_j , representa el factor de escalado del terremoto j . Según los especialistas en sentido físico, si el factor de escalado es igual a uno, significa que el movimiento del terreno resultó ser totalmente compatible con el espectro respuesta definido para ese sitio, hecho que señala una condición ideal; ya que el espectro de un movimiento real nunca resultaría plenamente coincidente, en todas sus ordenadas espectrales, con el espectro medio suavizado de muchos terremotos diferentes. En otros casos, a medida que este valor se acerca a uno, más

características en común con el espectro de sitio estudiado tiene el terremoto que lo posee.

Como única restricción a este problema se tiene que un mismo terremoto no puede ser seleccionado más de una vez para formar parte de una misma solución.

La Tabla 19 muestra los parámetros utilizados para la ejecución de las estrategias de exploración en una y dos etapas del algoritmo MMAS (primera columna). La condición de parada definida para esta experimentación representa una cantidad máxima de iteraciones (Ci_{max}). Por solicitud de los especialistas se realizaron las experimentaciones para encontrar soluciones de longitud siete (cantidad de terremotos). Debido al tamaño de las soluciones y la gran cantidad de terremotos que presentan las bases de casos seleccionadas, se decidió utilizar en el caso de la búsqueda en dos etapas un radio igual a $r=0,5$ (valor donde se alcanza la mayor eficiencia del método).

Tabla 19: Parámetros utilizados para la ejecución de los métodos basados en hormigas.

Algoritmo	m	$\tau(0)$	ρ	α	β	r	Condición de parada
MMAS	15	0,2	0,8	2	3	-	$Ci_{max}=1000$
TS-MMAS	15	0,2	0,8	2	3	0,5	$Ci_{max}=1000$

Para este algoritmo los valores máximos y mínimos de feromona fueron computados según las ecuaciones (0.10) y (0.11).

Aplicación de la Optimización basada en Mallas Dinámicas

Al resolver este problema usando DMO cada nodo de la malla inicial representa una lista con k terremotos (para las experimentaciones $k=7$),

donde cada elemento es identificado por el índice que tiene el terremoto en la base de casos. Los operadores de DMO para este problema de optimización discreta son las siguientes:

Función f :

$$n^*(i) = \begin{cases} ne(i) & \text{si } va[0,1] < r \\ n(i) & \text{en otros casos} \end{cases} \quad (0.7)$$

Función g :

$$n^*(i) = \begin{cases} ng(i) & \text{si } va[0,1] < r \\ n(i) & \text{en otros casos} \end{cases} \quad (0.8)$$

Donde r se calcula como sigue:

$$r = 1 - 0,5 \cdot \frac{FO(ne) \text{ ó } FO(ng)}{FO(n)} \quad (0.9)$$

Función h :

$$n^*(i) = \begin{cases} nl(i) & \text{si } va[0,1] > w \\ Sa(i) & \text{en otros casos} \end{cases} \quad (0.10)$$

Para la función h el valor de w es calculado según la ecuación (0.13) (con $w_0=1$ y $w_f=0$) y Sa representa una selección aleatoria dentro del conjunto de terremotos que no estén en la solución actual. Para identificar los nodos de las fronteras se utilizó el valor del factor de escalado (SI) del espectro promedio de cada solución (ecuación (0.5)), donde los nodos cuyo factor SI estén más distantes de uno, representan la frontera superior (más externos) y los más cercanos a uno, la frontera

inferior (más internos). Esta forma de modelar las fronteras se llevó a cabo porque según los expertos el valor uno representa, en términos físicos, total coincidencia entre un espectro respuesta y un objetivo.

Para la experimentación se utilizó como tamaño de la malla inicial 12 nodos, tamaño máximo de la expansión 42 y como condición de parada se fijaron 1000 iteraciones.

Algoritmo Voraz basado en la Búsqueda de los Vecinos más Cercanos

El algoritmo G-NNS busca para cada posición de la solución, el espectro respuesta (terremoto) en la base de casos, que más se ajusta al espectro objetivo en un perfil de suelo determinado, según su factor de escalado (utilizando la ecuación (0.6)). De esta manera, se obtienen los k terremotos que más se ajustan al espectro objetivo definido. La implementación de este algoritmo se realizó por una propuesta de la contraparte (los expertos de dominio), pensando que sería la variante más aceptada para la solución de este problema.

Estructura de los datos utilizados

La base de casos de movimientos fuertes PEER³³ representa la información recolectada por más de 143 estaciones en las regiones activas del pacífico, procesada por el Centro de Ingeniería del Pacífico y presentada en el 2001 para la comunidad científica.

³³Toda la información académica se puede encontrar en <http://peer.berkeley.edu/smcat/data.html>, procesada por "The Pacific Earthquake Engineering Research Center through the Earthquake Engineering Research Centers Program of the National Science Foundation under Award number EEC-9701568".

Record Sequence Number	EQID	Earthquake Name	YEAR	MODY	HRMN	Station Name	T0.950S	T1.000S	T1.100S	T1.200S
1	0001	Helena, Montana-01	1935	1031	1838	Carroll College	0.082330	0.078407	0.078401	0.073646
2	0002	Helena, Montana-02	1935	1031	1918	Helena Fed Bldg	0.001437	0.001288	0.001157	0.001060
3	0003	Humbolt Bay	1937	0207	0442	Ferndale City Hall	0.035617	0.032419	0.029937	0.028956
4	0004	Imperial Valley-01	1938	0606	0242	El Centro Array #9	0.006528	0.005969	0.003846	0.002763
5	0005	Northwest Calif-01	1938	0912	0610	Ferndale City Hall	0.093364	0.092462	0.052367	0.042904
6	0006	Imperial Valley-02	1940	0519	0437	El Centro Array #9	0.367939	0.365002	0.318345	0.304149
7	0007	Northwest Calif-02	1941	0209	0945	Ferndale City Hall	0.035262	0.030124	0.028145	0.025855
8	0008	Northern Calif-01	1941	1003	1614	Ferndale City Hall	0.080867	0.068681	0.048418	0.035450
9	0009	Borrego	1942	1021	1622	El Centro Array #9	0.055669	0.051048	0.046784	0.050495
10	0010	Imperial Valley-03	1951	0124	0717	El Centro Array #9	0.042356	0.037374	0.029404	0.031951
11	0011	Northwest Calif-03	1951	1008	0411	Ferndale City Hall	0.076859	0.072843	0.069555	0.054563
12	0012	Kern County	1952	0721	1153	LA - Hollywood Stor FF	0.129340	0.115335	0.094503	0.075818
13	0012	Kern County	1952	0721	1153	Pasadena - CIT Athenaeum	0.152845	0.135382	0.094754	0.073301
14	0012	Kern County	1952	0721	1153	Santa Barbara Courthouse	0.264322	0.263076	0.207147	0.183493

Ilustración 13: Estructura de la base de historias de terremotos PEER.

La Ilustración 13; **Error! No se encuentra el origen de la referencia.** muestra la estructura de esta base de casos. Debido a la gran cantidad de rasgos o variables que contiene, solo se muestra una parte de ella. Esta base de sismos cuenta con 1557 casos o registros de terremotos y con casi 150 variables, dentro de las cuales se consideran 105 que pertenecen al período de tiempo total registrado de 0 a 10 segundos (columnas que comienzan con T, seguido del tiempo especificado). Las variables restantes dan información innecesaria para el proceso de búsqueda tales como: el nombre del terremoto, en qué año ocurrió, qué estación los reportó, entre otros datos.

Los datos que forman los espectros objetivos de nuestro país fueron suministrados por los especialistas en Ingeniería Sísmica. Estos datos se dividieron en cuatro perfiles de suelo o sitios con tres zonas de peligro sísmico cada uno Ilustración 14.

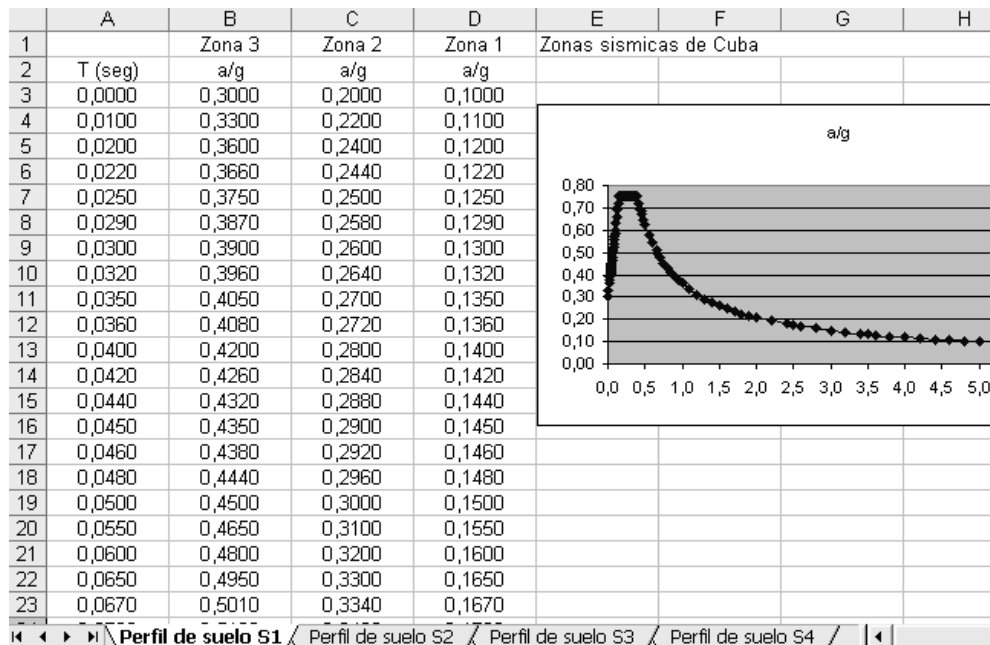


Ilustración 14: Ejemplo de la norma cubana para distintos perfiles de suelo peligroso.

La columna A son los períodos en segundos, hasta 5 segundos, con un paso coincidente con el de los espectros de respuesta de la base de terremotos PEER, para evitar así la existencia de puntos flotantes en el escalamiento, integración y funciones de ajuste. Las columnas de la B a la D son las ordenadas espectrales para cada zona sísmica expresada en términos de aceleración del terreno como fracción de la gravedad (a/g). Nuestro espectro actual es asintótico a partir de los 3 segundos de período; lo que se hizo fue alargar con ese mismo comportamiento hasta los 5 segundos, para tener la cantidad de datos que faciliten el match.

La base de casos inicial llega hasta los 10 segundos de períodos. Esta se modificó y se acotó a los 5 segundos, pues en nuestro país actualmente no existen estructuras que oscilen con valores de períodos tan altos. No obstante, este es un aspecto que pudiera ser modificable según las nuevas necesidades.

Análisis de los resultados obtenidos por los métodos presentados

En este epígrafe se realiza el estudio experimental de los resultados obtenidos por los métodos propuestos, para lo cual se utilizó el marco estadístico del Epígrafe 1.3. Los algoritmos fueron computados utilizando la base de casos sin filtrar, con el objetivo de verificar el comportamiento de los métodos para grandes volúmenes de datos. Además de esto, se realizaron pruebas utilizando o no el proceso de ajuste de las soluciones. Para todas las pruebas, la cantidad de terremotos que componen una solución es igual a siete ($k=7$).

A continuación, se presenta el análisis estadístico desarrollado sobre los resultados para la calidad de las soluciones. Debido a errores detectados por los especialistas en los datos del perfil de suelo dos, no se presentan soluciones para este caso.

Tabla 20: Resultados del test de Iman-Davenport para el caso de estudio.

Método	Valor del Test	Valor de distribución F	Valor p	Hipótesis
Iman-Davenport	76,155	<i>Sin el proceso de ajuste</i>		
		3,0088	2,09E-12	R
Iman-Davenport	82,002	<i>Con el proceso de ajuste</i>		
		3,0088	9,39E-13	R

La Tabla 20 muestra que tanto para el proceso sin ajuste como con ajuste, el test de Iman-Davenport detectó diferencias significativas entre los cuatro algoritmos computados para ambos casos. Debido a esto, la **¡Error! No se encuentra el origen de la referencia.** y la **¡Error! No se encuentra el origen de la referencia.** describen el test de Bonferroni-Dunn, donde el algoritmo de control (TS-MMAS) obtiene resultados

significativamente superiores que los algoritmos DMO y G-NNS en ambos casos, ya que sus rangos medios se encuentran por encima de los patrones de comparación (2,814 y 2,759) obtenidos para el valor crítico (CD=1,426) computado para esta prueba.

Con respecto a los resultados del algoritmo MMAS no se detectan diferencias con el algoritmo de control TS-MMAS para ambos casos de estudios.

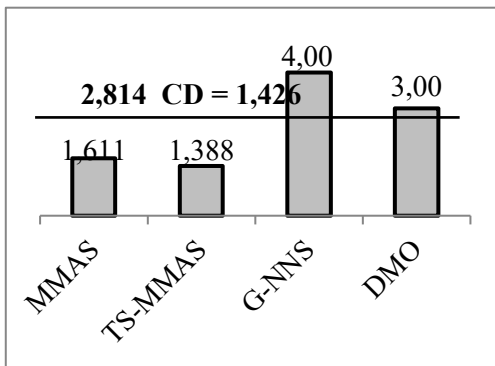


Ilustración 15: Resultados del test de Bonferroni-Dunn sin el proceso de ajuste.

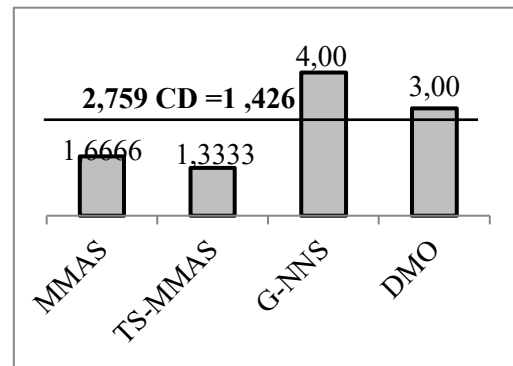


Ilustración 16: Resultados del test de Bonferroni-Dunn con el proceso de ajuste.

Para probar si no existen diferencias significativas, en cuanto a calidad de las soluciones, entre los algoritmos MMAS y TS-MMAS se presentan en la

Tabla 21 los resultados del test de Wilcoxon para ambos casos de estudios. En ellos, se puede apreciar como la hipótesis de igualdad es aceptada, por lo que se garantiza que las diferencias, aunque existen, no son significativas. En el caso sin ajuste la TS-MMAS alcanza mejores resultados y para el proceso con ajuste los mejores resultados los obtiene MMAS.

Tabla 21: Resultados del test de Wilcoxon para ambos casos de estudio.

Algoritmo	R ⁺	R ⁻	Valor p	Hipótesis
TS-MMAS vs MMAS	15	30	0,374	A
		<i>Sin el proceso de ajuste</i>		
TS-MMAS vs MMAS	23	22	0,953	A
		<i>Con el proceso de ajuste</i>		

No solo la eficacia de los algoritmos es importante cuando se quieren resolver problemas reales, también hay que tener en cuenta la velocidad (costo de tiempo de cómputo) en que se obtienen las soluciones, específicamente para grandes espacios de búsquedas. En este caso las bases utilizadas contienen gran cantidad de terremotos, por lo que se hace necesario seleccionar algoritmos que tengan buena relación entre eficiencia y eficacia.

En cuestiones de eficiencia el algoritmo que obtiene los resultados de manera más rápida es el G-NNS, el cual por su naturaleza determinista provoca que realice la menor cantidad de ejecuciones posibles; pero como se demostró anteriormente su eficacia es baja. Otro método que se comportó de manera eficiente fue la propuesta en dos etapas del algoritmo MMAS (TS-MMAS), computado para el valor que le reporta su mayor grado de eficiencia ($r=0,5$). Por último, los algoritmos DMO y MMAS son los que obtienen peores niveles de eficiencia para este problema.

Conclusiones parciales

En este capítulo se presentó un estudio para la solución del problema de selección y escalamiento de registros sísmicos fuertes, con el objetivo de seleccionar el algoritmo que formará parte del núcleo principal de un software implementado (SISMACH) para el estudio estructural en nuestro país.

Como conclusión de esta experimentación se tiene que:

- El algoritmo G-NNS obtuvo las soluciones de manera más rápida, pero con peor calidad entre todos los métodos propuestos.
- El algoritmo MMAS en una etapa obtuvo resultados competitivos en cuanto a calidad; pero, como se presentó en estudios anteriores, no es eficiente para grandes espacios de búsqueda.
- Por último, entre las propuestas presentadas en este trabajo (TS-ACO y DMO), la exploración en dos etapas de los algoritmos ACO, específicamente TS-MMAS, es la que mejor relación obtuvo entre eficiencia y eficacia para este problema; siendo capaz de alcanzar las mejores soluciones en cuanto a calidad conjuntamente con el MMAS, y de manera eficiente, solo superado por el G-NNAS. El algoritmo DMO por su parte obtuvo soluciones moderadas para la modelación realizada para este problema.

Debido a estas conclusiones el algoritmo TS-MMAS fue seleccionado para formar parte del software implementado, cuyos resultados fueron bien acogidos por los ingenieros estructurales. Según los especialistas la obtención de errores específicos por debajo de cinco es considerada un buen resultado. En relación con esto, la propuesta en dos etapas obtuvo valores inferiores a cinco en todos los casos de la experimentación.

CONCLUSIONES

Como resultado de la investigación para este libro se desarrollaron nuevos métodos de solución de problemas complejos; los cuales, basados en el enfoque poblacional, proporcionan a los investigadores en esta rama otras alternativas para enfrentar los problemas existentes; particularmente se encontraron buenas soluciones al problema de selección y escalamiento de movimientos sísmicos fuertes.

Se formuló una nueva forma de exploración del espacio de búsqueda para los algoritmos basados en Colonia de Hormigas (ACO) llamada Optimización Basada en Colonia de Hormigas en Dos Etapas (TS-ACO). Esta propuesta muestra que dividir en dos el proceso de búsqueda desarrollado por las hormigas permite aumentar la eficiencia de los algoritmos ACO entre un 40 % y 60 %, manteniendo la calidad de los resultados. Para aumentar la eficacia de dicha propuesta, se proponen dos alternativas (P-TS-ACO y S-TS-ACO), las cuales aumentan el nivel de exploración en la segunda etapa de TS-ACO. En los resultados experimentales se puede comprobar que la aplicación de estas alternativas mejora la calidad de las soluciones de ACO en los problemas de estudio.

Se propuso una metaheurística poblacional llamada Optimización Basada en Mallas Dinámicas (DMO); en la cual, se incorporaron los nuevos elementos siguientes:

- a. Involucrar en una misma iteración la exploración global y en vecindades, manteniendo la dirección de la búsqueda (exploración y explotación).

-
- b. Un proceso para explorar en dirección a las fronteras definidas para el dominio en cuestión a partir de los nodos más y menos externos de la malla.
 - c. Realizar una selección elitista en cuanto a calidad, teniendo en cuenta la separabilidad entre soluciones.
 - d. La utilización de un proceso de limpieza adaptativo, para variar las distancias permitidas entre soluciones durante la ejecución del algoritmo.

En los experimentos desarrollados para dominios continuos se probó que el método DMO presenta un alto nivel de escalabilidad, obteniendo mejores resultados de forma general que otros algoritmos del estado del arte.

El esquema general de aplicación que se propone para la solución del Problema de Selección y Escalamiento de movimientos sísmicos fuertes muestra que, las propuestas presentadas en este trabajo (TS-ACO y DMO), son una buena alternativa para la solución de problemas reales. Específicamente el algoritmo TS-ACO fue el que obtuvo mejor balance entre eficiencia y eficacia. En cuanto al nivel de satisfacción presentados por los expertos en el campo de la Ingeniería Estructural, se obtuvieron opiniones halagüeñas de la aplicación desarrollada, apoyando la validación de los resultados experimentales de las nuevas propuestas.

REFERENCIAS BIBLIOGRÁFICAS

1. Spendley, W., G.R. Hext, y F.R. Himsforth, *Sequential Application of Simplex Designs in Optimizacion and Evolutionary Operation*. Technometrics, 1962. **4**(4): p. 441-457.
2. Nemhauser, G. y L. Wolsey, *Integer and Combinatorial Optimization*. 1ra ed. 1998, New York: John Wiley.
3. Polya, G., *How to Solve It*. 1957, Princeton University Press.
4. Simon, H.A., *Experiments with a Heuristic Compiler*. Journal of the ACM, 1963. **10**(4): p. 493 - 506.
5. Hooker, J.H., *Testing Heuristics: We Heve it All Wrong*. Journal of Heuristics, 1995. **1**(1): p. 33-42.
6. Silver, E.A., R.V. Vidal, y D.D. Werra, *A Tutorial on Heuristic Methods*. European Journal of Operational Research, 1980. **5**: p. 153-162.
7. Wolpert, D.H. y W.G. Macready, *No Free Lunch Theorems for Optimization*. IEEE Transactions of Evolutionary Computation, 1997. **1**: p. 67-82.
8. Osman, I.H. y J.P. Kelly, *Meta-Heuristics: Theory and Applications*. 1996, Kluwer Academic: Boston.
9. Michalewicz, Z. y D. Fogel, *How to solve it*. Modern heuristics. 2004: Springer. 554.
10. Glover, F. y G. Kochenberger, eds. *Handbook of Metaheuristics*. 2003, Kluwer Academic Publishers.

-
11. Blum, C. y A. Roli, *Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison*. ACM Computing Surveys, 2003. **35**(3): p. 268-308.
 12. Glover, F. y M. Laguna, *Tabu Search*. 1ra ed. 1997, MA: Kluwer Academic Publishers.
 13. Kirkpatrick, S., C. Gellat, y M. Vecchi, *Optimization by simulated annealing*. Science, 1983. **220**(4598): p. 671–680.
 14. Hansen, P., N. Mladenović, y J.A.M. Pérez, *Variable neighborhood search*. European Journal of Operational Research, 2008. **191**(3): p. 593-595
 15. Voudouris, C. y E.Tsang, *Guided local search*, Technical Report CSM-247 1995, Department of Computer Science.
 16. Hoos, H.H. y T. Stutzle, *Local search algorithms for SAT: An empirical evaluation*. Journal of Automated Reasoning, 1983. **24**(4): p. 421-481.
 17. Bäck, T., U. Hammel, y H. Schwefel, *Evolutionary Computation: Comments on the History and Current State*. IEEE Transactions on Evolutionary Computation Estados Unidos, 1997. **1**(1): p. 3-17.
 18. Cano, J.R., F. Herrera, y M. Lozano, *Using Evolutionary Algorithms as Instance Selection for Data Reduction in KDD: An Experimental Study*. IEEE Transactions of Evolutionary Computation, 2003. **7**: p. 561-575.
 19. Bonabeau, E., *Swarm Intelligence: From natural to artificial systems*. 1ra ed. 1999: Oxford University Press.

-
20. Engelbrecht, A.P., *Fundamentals of Computational Swarm Intelligence*. 2006, John Wiley y Sons.
 21. Darwin, C., *On the Origin of Species*. 1859: London: John Murray. 126.
 22. Goldberg, D.E., ed. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1998, Addison-Wesley Publishing Company: University of Alabama.
 23. Kennedy, J. y R.C. Eberhart. *Particle swarm optimization*. en *IEEE International Conference on Neural Networks*. 1995. Piscataway, New York, USA: . Vol. 4: p. 1942-1948.
 24. Dorigo, M. (1992): *Optimization, Learning and Natural Algorithms*. Phd. Dipartimento di Elettronica. Politecnico di Milano. p 213.
 25. Talbi, E., *A Taxonomy of Hybrid Metaheuristics*. *Journal of Heuristics*, 2002. **8**: p. 541-564.
 26. Yagiura, M. y T. Ibaraki, *On Metaheuristic Algorithms for Combinational Operator Problems*. *Systems Computation*, 2001. **32**(3): p. 33–55.
 27. Voss, S., et al., eds. *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*. 1999, Kluwer Academic Publishers: Boston, MA.
 28. Koza, J., *Genetic programming: On the programming of computers by means of natural selection*. 1ra ed. 1992, Cambridge: MA: MIT Press. 819.
 29. Beyer, H.-G., ed. *The Theory of Evolution Strategies*. 2001: Springer.

-
30. Langdon, W.B. y R. Poli, *Foundations of Genetic Programming*. 1ra ed. 2002, New York: Springer-Verlag 260.
 31. Krasnogor, N. y J. Smith, *A Tutorial for Competent Memetic Algorithms: Model, Taxonomy, and Design Issue*. IEEE Transactions on Evolutionary Computation, 2005. **9**(5): p. 474–488.
 32. Ong, Y., et al., *Classification of Adaptive Memetic Algorithms: a Comparative Study*. IEEE Transactions on System, Man. and Cybernetic, 2006. **36**(1): p. 141-152.
 33. Muhlenbein, H., M. Schomisch, y J. Born. *The parallel genetic algorithm as function optimizer*. en *Fourth International Conference on Genetic Algorithms*. 1991. San Mateo, California: Morgan Kaufmann: p. 271–278.
 34. Merz, P., *NK-Fitness Landscapes and Memetic Algorithms with Greedy Operators and k-opt Local Search*. Recent Advances in Memetic Algorithms, 2004. **166**: p. 209-228.
 35. Merz, P. y K. Katayama, *Memetic Algorithms for the Unconstrained Binary Quadratic Programming Problem*. Bio System, 2004. **78**((1-3)): p. 99-18.
 36. Seront, G. y H. Bersini. *A new ga-local search hybrid for continuous optimization based on multi level single linkaga clustering*. en *of the Genetic and Evolutionary Computation Conference 2000*. 2000: p. 90-95.

-
37. Eshelman, L., *The CHC Adaptive Search Algorithm. How to Have Safe Search When Engaging in Nontraditional Genetic Recombination*. Foundations of Genetic Algorithms, 1991: p. 265-283.
 38. Jong, K.A.D. y W.M. Spears, *A Formal Analysis os the Role of Multi-Point Crossovers in Genetic Algorithms*. Annals of Mathematics and Artificial Intelligence, 1992. **5**(1): p. 1-26.
 39. kita, H., *A Comparison Study of Self-Adaptation in Evolution Strategies and Real-Coded Genetic Algorithms*. Evolutionary Computation 2001. **9**(2): p. 223-241.
 40. Herrera, F., M. Lozano, y A. Sánchez, *A Taxonomy for the Crossover Operator for Real Coded Genetic Algorithms: An Experimental Study*. International Journal of Intelligent Systems, 2003. **18**: p. 309-338.
 41. Eshelman, L.J. y J. Schaffer, *Real-coded genetic algorithms and Interval-Shemata*. Fundation of Genetic Algorithms, 1993. **2**: p. 187-202.
 42. Deb, K. y R.B. Agrawal, *Simulated Binary Crossover for Continuos Search Space*. Complex Systems, 1995. **9**: p. 115-148.
 43. Coigt, H., H. Muhlenbein, y D. Cvetkovic. *Fuzzy recombination for the breeder genetic algorithm*. en *Sixth International Conference on Genetic Algorithms*. 1995: San Mateo CA: Morgan Kaufmann: p. 104-111.
 44. Ballester, P. y J. Carter. *An effective real-parameter genetic algorithm with parent centric normal crossover for multimodal*

optimization en Genetic and Evolutionary Computation Conference 2004: LNCS: p. 901-913.

45. Kazarlis, S., et al., *Microgenetic algorithms as generalized hill-climbing operator for genetic algorithms optimization*. IEEE Transactions on Evolutionary Computation, 2001. **5**(3): p. 204–217.
46. Laguna, M. y R. Martí, *Scatter Search. Methodology and Implementation in C*. 1ra ed. 2003: Kluwer Academic Publishers.
47. Price, K.V., R.M. Rainer, y J.A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. 2005, Springer-Verlag: London, UK.
48. Larrañaga, P. y J.A. Lozano, *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation 2002*, Kluwer Academic Publishers: Boston/Dordrecht/London.
49. Moscato, P.A., *Memetic Algorithms: a Short Introduction*, D. Corne, M. Dorigo, and F. Glover, Editors. 1999, McGraw-Hill: London.
50. Hart, W., N. Krasnogor, y J. Smith, eds. *Recent Advances in Memetic Algorithms*. 2005, Springer Berlin Heidelberg: New York.
51. Dorigo, M. y T. Stutzle, *Ant Colony Optimization*. 1ra ed. 2004, Cambridge, MA: MIT Press. 324.
52. Dorigo, M., E. Bonabeau, y G. Theraulaz, *Ant algorithms and stigmergy*. Future Generation Computer Systems, 2000. **16**(8): p. 851-871.

-
53. Dorigo, M. y G.D. Caro, *The Ant Colony Optimization meta-heuristic*, en *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Editors. 1999, McGraw-Hill: London UK. p. 11-32.
 54. Dorigo, M., G.D. Caro, y L.M. Gambardella, *Ant algorithms for discrete optimization*. *Artificial Life*, 1999. **5**(2): p. 137-172.
 55. Dorigo, M. y T. Stutzle, *The ant colony optimization metaheuristic algorithms, applications, and advances*, en *Handbook of Metaheuristics*, F.Glover and G.A. Kochenberger, Editors. 2003, Kluwer. p. 251-258.
 56. Dorigo, M., M. Birattari, y T. Stutzle, *Ant colony optimization*. *Computational Intelligence*, 2006. **1**(4): p. 28-39.
 57. Dorigo, M. y L.M. Gambardella, *Ant colonies for the travelling salesman problem*. *Biosystems*, 1997. **43**(2): p. 73-81.
 58. Gambardella, L.M., È.D. Taillard, y M. Dorigo, *Ant colonies for the Quadratic Assignment Problem*. *Journal of the Operational Research Society*, 1999. **50**(2): p. 167-176.
 59. Bello, R. y A. Nowé. *Using ACO and Rough Set Theory to Feature* en *WSEAS Transactions on Information Science and Applications*. 2005. Vol. 2: p. 512-517.
 60. Crawford, B. y C. Castro, *Ant Colonies using Arc Consistency Techniques for the Set Partitioning Problem*. *LNAI* 2006. **4183**: p. 45-55.
 61. Dorigo, M. y C. Blum, *Ant colony optimization theory: a survey*. *Theory and Computer Science*, 2005. **344**(2-3): p. 243-278.

-
62. Dorigo, M., V. Maniezzo, y A. Colorni, *The Ant System: Optimization by a colony of cooperating agents*. IEEE Transactions on Systems Man, and Cybernetics, 1996. **Part B 26**(1): p. 29-41.
 63. Dorigo, M. y L.M. Gambardella, *Ant Colony System: A cooperative learning approach to the traveling salesman problem*. IEEE Transactions on Evolutionary Computation, 1997. **1**(1): p. 53-66.
 64. Stützle, T. y H. Hoos, *MAX-MIN Ant System*. Future Generation Computer Systems, 2000. **16**(8): p. 889-914.
 65. Bullnheimer, B., R.F. Hartl, y C. Strauss, *A new rank-based version of the Ant System: A computational study*. Central European Journal for Operations Research and Economics, 1999. **7**(1): p. 25-38.
 66. Cordón, O., et al. *new ACO model integrating evolutionary computation concepts: The Best-Worst Ant System*. en ANTS 2000. 2000. Université Libre de Bruxelles, Belgium,: Springer: p. 22-29.
 67. Heinonen, J. y F. Pettersson, *Hybrid ant colony optimization and visibility studies applied to a job-shop scheduling problem* Applied Mathematics and Computation, 2007. **187**(2): p. 989-998.
 68. Liu, Y.X., J. Xiong, y B.M. Sun, *Research on dynamic scheduling of job-shop production with the ant colony optimal algorithm*. Applied Mechanics and Materials, 2008. **10**(12): p. 109-113.
 69. Manno-lugano y L.M. Gambardella, *An Ant Colony System Hybridized with a New Local Search for the Sequential Ordering Problem* INFORMS Journal on Computing, 2000. **12**(3): p. 237-255

-
70. Wong, K.Y. y P.C. See, *A new minimum pheromone threshold strategy (MPTS) for max-min ant system* *Jornal Soft Computing*, 2009. **9**(3): p. 882-888.
 71. MAKOTO, K., et al., *Applying the MAX-MIN Ant System to the DNA Sequencing and Its Improvement*. *IEICE Transactions on Information and Systems*, 2006. **J89-D**(5): p. 911-918.
 72. Pitakaso, R., et al., *A MAX-MIN ant system for unconstrained multi-level lot-sizing problems* *Source Computers and Operations Research archive*, 2007. **34**(9): p. 2533-2552.
 73. Huang, K.L. y C.J. Liao, *Ant colony optimization combined with taboo search for the job shop scheduling problem*. *Computers and Operations Research*, 2008. **35**(4): p. 1030-1046.
 74. Naimi, H.M. y N. Taherinejad, *New robust and efficient ant colony algorithms: Using new interpretation of local updating process*. *Expert Systems with Applications*, 2009. **36**(1): p. 481-488.
 75. Wu, Z., et al., *Population declining ant colony optimization algorithm and its applications*. *Expert Systems with Applications*, 2009. **36**(3): p. 6276-6281.
 76. Wu, Z., et al., *New robust and efficient ant colony algorithms: Using new interpretation of local updating process*. *Expert Systems with Applications*, 2009. **36**(1): p. 481-488.
 77. Demsar, J., *Statistical comparisons of classifiers over multiple data sets*. *Journal Machine Learn*, 2006. **7**: p. 1-30.

-
78. García, S. y F. Herrera, *An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons*. Journal of Machine Learning Research, 2008. **9**: p. 2677-2694.
 79. Sheskin, D., *Handbook of parametric and nonparametric statistical procedures*, CRC, Editor. 2006, Chapman & Hall: London/West Palm Beach.
 80. García, S., et al., *A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability*. Soft Computing and Applications, 2009. **13**(10): p. 959-977.
 81. García, S., et al., *A Study on the Use of Non-Parametric Tests for Analyzing the Evolutionary Algorithms' Behaviour: A Case Study on the CEC'2005 Special Session on Real Parameter Optimization*. Journal of Heuristics, 2009. **In press**. DOI: 10.1007/s10732-008-9080-4.
 82. Luengo, J., S. García, y F. Herrera, *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, 2009. **36**: p. 7798-7808.
 83. Iman, R.L. y J.M. Davenport, *Approximations of the critical region of the Friedman statistic*. Commun Stat, 1980. **18**: p. 571-595.
 84. Dunn, O.J., *Multiple comparisons among means*. Journal of the American Statistical Association, 1961. **56**: p. 52-64.
 85. Holm, S., *A simple sequentially rejective multiple test procedure*. Scand Journal Stat, 1979. **6**: p. 65-70.

-
86. Wilcoxon, F., *Ajusted p-values for simultaneous inference*. *Biometrics*, 1945. **48**: p. 1005-1013.
 87. Puris, A., et al. *Optimización basada en Colonias de Hormigas en Dos Etapas aplicada al problema de Planificación de la Fuerza de Trabajo*. en *Memorias del VI Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados*. 2009. Málaga, España: p. 239-247.
 88. Puris, A. y R. Bello, *Optimización Basada en Colonias de Hormigas en Dos Etapas para el Problema de Asignación Cuadrática*, en *CD de Memoria del XIV Congreso Latino Ibero Americano de Investigación de Operaciones CLAIO*. 2008: Cartagena de Indias, Colombia.
 89. Puris, A., et al. *Two-Stage ACO to solve the Job Shop Scheduling Problem*. en *12th Iberoamerican Congress on Pattern Recognition CIARP 2007*. 2007. Viña del Mar/Valparaíso, Chile: Springer-Verlag Berlin Heidelberg, Lecture Note in Computer Science. Vol. 4756: p. 969-975.
 90. Puris, A., et al. *Two-Step Ant Colony Optimization for solving the Traveling Salesman Problem*. en *2nd. INTERNATIONAL WORK-CONFERENCE on the INTERPLAY between NATURAL and ARTIFICIAL COMPUTATION (IWINAC 2007)*. 2007. España: Springer-Verlag Berlin Heidelberg, Lecture Note in Computer Science. Vol. 4528: p. 307-316.
 91. Puris, A., et al. *Improving Ant Colony Optimization to Solver Traveling Salesman Problem*. en *18h Belgium-Netherlands Conference on Artificial Intelligence*. 2006. Belgium: p. 267-272.

-
92. Puris, A., et al. *Two Step Ant Colony System for solving the Traveling Salesman Problem*. en *Fifth Mexican International Conference on Artificial Intelligence, MICAI*. 2006. Mexico: Research in Computing Science. Vol. 26: p. 85-92.
 93. Bello, R., A. Puris, y Y. Martinez. *Two Step Ant Colony System to Solve the Feature Selection Problem*. en *11th Iberoamerican congress in pattern recognition, CIARP 2006*. 2006. Cancun, Mexico: Springer Berlin / Heidelberg Lecture Notes in Computer Science. Vol. 4225: p. 588-596.
 94. Gomez, Y., et al., *Two Step Swarm Intelligence to Solve the Feature Selection Problem*. *Journal of Universal Computer Science*, 2008. **14**(15): p. 2582-2596.
 95. Stützle, T. y H.H. Hoos, *The MAX-MIN ant system and local search for the traveling salesman problem*, en *International Conference on Evolutionary Computation (ICEC'97)*, T. Bäck, Z. Michalewicz, and X. Yao, Editors. 1997, IEEE: USA. p. 309-314.
 96. Gutin, G. y A. Punnen, *The Traveling Salesman Problem and Its Variations*. 2da ed. *Combinatorial Optimization*, ed. D. Kluwer Academic. Vol. 12. 2002: Springer. 830.
 97. Karp, R., *Reducibility among combinatorial problems*, en *Complexity of Computer Computations*, R. Mille and J. Thatcher, Editors. 1972: New York. p. 85-103.
 98. Reinelt, G., *TSPLIB - A traveling salesman library*. *ORSA Journal on Computing*, 1991. **3**: p. 376-384.

-
99. Dorigo, M. y T. Stützle, *ACO Algorithms for the Traveling Salesman Problem*, en *Evolutionary Algorithms in Engineering and Computer Science*. 1999, John Wiley Sons: EEUU. p. 132-160.
100. Cela, E., *The Quadratic Assignment Problem: Theory and Algorithms*. 1ra ed. Combinatorial Optimization, ed. D. Ding-Zh, Pardalos, and M. Panos. Vol. 1. 1998, London: Kluwer Academic. 287.
101. Sahni, S. y T. Gonzalez, *P-complete approximation problems*. Journal of the Association for Computing Machinery, 1976. **23**: p. 555–565.
102. Maniezzo, V. y A. Colomi, *The Ant System applied to the Quadratic Assignment Problem*. IEEE Transactions on Knowledge and Data Engineering, 1999. **11**(5): p. 769-778.
103. Parsopoulos, K.E. y M.N. Vrahatis, *Recent approaches to global optimization problems through Particle Swarm Optimization*. Natural Computing 2002. **1**: p. 235-306.
104. Puris, A. y R. Bello. *Optimización basada en Mallas Dinámicas. Su aplicación en la solución de problemas de optimización continuos*. en *Memorias del VI Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados*. 2009. Málaga, España: p. 441-448.
105. Puris, A. y R. Bello, *Búsqueda heurística usando optimización basada en mallas dinámicas*, en *CD de Memorias del XIV Congreso Latino Ibero Americano de Investigación de Operaciones CLAIO*. 2008, : Cartagena de Indias, Colombia.

-
106. Bello, R., et al. *Feature selection through Dynamic Mesh Optimization*. en *13th Iberoamerican Congress on Pattern Recognition, CIARP*. 2008. Habana, Cuba: Lecture Notes on Computer Sciences LNCS. Vol. 5197: p. 348-355.
107. Suganthan, P., et al., *Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization*, 2005, Nanyang Technological University.
108. Minetti, G. y H. Alfonso. *Variable Size Population in Parallel Evolutionary Algorithms*. en *Fifth International Conference on Intelligent Systems Design and Applications*. 2005: IEEE: p. 350-355.
109. Syswerda, G. *Uniform Crossover in Genetic Algorithms*. en *Thrid International Conference on Genetic Algorithms*. 1989. San Mateo California: Morgan Kaufmann Publishers: p. 2-9.
110. Rahnamayan, S., H.R. Tizhoosh, y M.M.A. Salama, *Solving large scale optimization problems by Opposition-Based Differential Evolution*. *IEEE Transactions on Computation*, 2008. **7**(10): p. 1792-1804.
111. Fernandes, C. y A. Rosa. *A Study of non-Random Matching and Varying Population Size in Genetic Algorithm using a Royal Road Function*. en *2001 Congress on Evolutionary Computation*. 2001: p. 60-66.
112. Shi, Y. y C. Eberhart. *A Modified Particle Swarm Optimizer*. en *IEEE International Conference on Evolutionary Computation*. 1998. Anchorage, Alaska, USA: IEEE: p. 69-73.

-
113. Barbat, A.H. y J.M. Canet, *Estructuras sometidas a acciones sísmicas*. 1994: Centro Internacional de Métodos Numéricos en Ingeniería (CIMCNI).
114. Clough, R. y J. Penzien, *Dynamics of Structures*. 2nd ed. 1993: McGraw-Hill.
115. Lin, C.J. y J. Ghaboussi. *A New Method of Generating Artificial Earthquake Accelerograms Using Neural Networks*. en *7th Int. Conf. on Computers in Civil and Building Engineering*. 1997. Seoul, Korea: Earthquake Engineering & Structural Dynamics. Vol. 27: p. 377-396.
116. A.K.Chopra, *Estimating seismic demands for performance-based engineering of building*, en *13 th World Conference of Earthquake Engineering*. 2004, Paper No. 5007: Vancouver, B.C, Canada.
117. Rueda, J.E.M., *Scaling procedure for natural accelerograms based a system of spectrum intensity scales*. *Earthquakes Spectra*, 2004. **14**(1): p. 135-152.
118. Alimoradi, A., F. Naein, y S. Pezeshk, *GA-Based selection and scaling of strong ground motion records for structural design*, en *13th World Conference on Earthquake Engineering WCEE*. 2004: Vancouver, B.C., Canada. p. 156-172.
119. Carballo, J.E. y C.A. Cornell, *Probabilistic Seismic Demand Analysis: Spectrum Matching and Design*, 2000, Department of Civil and Environmental Engineering, Stanford University, Report NO. RMS-41.
120. Bolt, B.A. y N.J. Gregor, *Synthesized Strong Ground Motions for the Seismic Condition Assessment of the Eastern Portion of the San*

Francisco Bay Bridge, 1993, University of California, Earthquake Engineering Research Center, Berkeley, CA.

121. Silva, W. y K. Lee, *State-of-the-art for Assessing Earthquake Hazards in the United States; Report 24, WES RASCAL Code for Synthesizing Earthquake Ground Motions*, 1987, Miscellaneous Paper S-73-1, U.S. Army Engineer Waterways Experiment Station, Vicksburg, MS.
122. Gasparini, D. y E.H. Vanmarcke, *SIMQKE: A Program for Artificial Motion Generation*, 1976, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA.
123. Friedman, M., *A Comparison of Alternative Tests of Significance for the Problem of m Ranking*. *Annals of Mathematical Statistics*, 1940. **11**: p. 86-92.
124. Friedman, M., *The use of ranks to avoid the assumption of normality implicit in the analysis of variance*. *Journal of the American Statistical Association*, 1937. **32**: p. 675-701.
125. Zar, J., *Biostatistical analysis*. 1ra ed. 1999: Prentice Hall, Englewood Cliffs.

PRODUCCIÓN CIENTÍFICA DEL AUTOR

- 1 BELLO, R., PURIS, A. y MARTINEZ, Y. (2006): Two Step Ant Colony System to Solve the Feature Selection Problem. Proc. of the 11th Iberoamerican Congress in Pattern Recognition CIARP 2006, Cancun, México, Springer Berlin / Heidelberg, Lecture Notes in Computer Science vol. 4225, pp 588-596.
- 2 PURIS, A., BELLO, R., MARTÍNEZ, Y. y NOWE, A. (2006): Two Step Ant Colony System for solving the Traveling Salesman Problem. Proc. of the Fifth Mexican International Conference on Artificial Intelligence, MICAI, Research in Computing Science, vol. 26, pp 85-92.
- 3 PURIS, A., BELLO, R., NOWE, A. y MARTÍNEZ, Y. (2006): Improving Ant Colony Optimization to Solver Traveling Salesman Problem. Proc. of the 18h Belgium-Netherlands Conference on Artificial Intelligence, pp 267-272.
- 4 MARTÍNEZ, Y., PURIS, A. y BELLO, R. (2006): Solución para problemas de configuración utilizando métodos heurísticos. Cd con las Memorias del evento 7ma Conferencia Científica Internacional, UNICA2006. Ciego de Avila, Cuba, ISBN 959-16-0473-4.
- 5 MARTÍNEZ, Y., PURIS, A. y BELLO, R. (2006): Aplicación de métodos heurísticos en la solución de problemas de configuración. Cd con las memorias del V Conferencia Internacional de Ciencias Empresariales CICE 2006. Santa Clara, Cuba, ISBN 959-250-297.
- 6 PURIS, A., MARTÍNEZ, Y. y BELLO, R. (2006): Aplicación de la Optimización Basada en Colonia de Hormigas al Problema de Viajero Vendedor. Cd con las memorias del V Conferencia Internacional de Ciencias Empresariales CICE 2006. Santa Clara, Cuba, ISBN 959-250-297.

-
- 7 PURIS, A., BELLO, R., NOWE, A. y MARTINEZ, Y. (2007): Two-Step Ant Colony Optimization for solving the Traveling Salesman Problem. Proc. of the 2nd. INTERNATIONAL WORK-CONFERENCE on the INTERPLAY between NATURAL and ARTIFICIAL COMPUTATION (IWINAC 2007), Springer-Verlag Berlin Heidelberg, Lecture Note in Computer Science, vol. 4528, pp 307-316.
 - 8 PURIS, A., BELLO, R., TRUJILLO, Y., NOWE, A. y MARTÍNEZ, Y. (2007): Two-Stage ACO to solve the Job Shop Scheduling Problem. Proc. of the 12th Iberoamerican Congress on Pattern Recognition (CIARP 2007), Springer-Verlag Berlin Heidelberg, Lecture Note in Computer Science, vol. 4756, pp 969-975.
 - 9 BELLO, R., PURIS, A., GÓMEZ, Y. y FALCÓN, R. (2008): Feature selection through Dynamic Mesh Optimization. Proc. of the 13th Iberoamerican Congress on Pattern Recognition, CIARP 2008, Springer-Verlag Berlin Heidelberg, Lecture Notes on Computer Sciences LNCS, vol. 5197, pp 348-355.
 - 10 PURIS, A. y BELLO, R. (2008): Optimización Basada en Colonia de Hormigas en Dos Etapas para el Problema de Asignación Cuadrática. CD de memoria del XIV Congreso Latino Ibero Americano de Investigación de Operaciones CLAIO.
 - 11 PURIS, A. y BELLO, R. (2008): Búsqueda heurística usando optimización basada en mallas dinámicas. CD de memorias del XIV Congreso Latino Ibero Americano de Investigación de Operaciones CLAIO. Cartagena de Indias, Colombia.
 - 12 GOMEZ, Y., BELLO, R., PURIS, A., GARCÍA, M. M. y NOWE, A. (2008). Two Step Swarm Intelligence to Solve the Feature Selection Problem. Journal of Universal Computer Science, 14(15), pp 2582-2596.

-
- 13 PURIS, A. y BELLO, R. (2009): Optimización basada en Mallas Dinámicas. Su aplicación en la solución de problemas de optimización continuos. Memorias del VI Congreso Español sobre metaheurísticas, Algoritmos Evolutivos y Bioinspirados, pp 441-448.
- 14 PURIS, A., BELLO, R., ALBA, E., SIMÓN, A. y CÁCERES, M. C. (2009): Optimización basada en Colonias de Hormigas en Dos Etapas aplicada al problema de Planificación de la Fuerza de Trabajo. Memorias del VI Congreso Español sobre metaheurísticas, Algoritmos Evolutivos y Bioinspirados, pp 239-247

Descubre tu próxima lectura

Si quieres formar parte de nuestra comunidad,
regístrate en <https://www.grupocompas.org/suscribirse>
y recibirás recomendaciones y capacitación



   @grupocompas.ec
compasacademico@icloud.com

Amilkar Yudier Puris Cáceres

Licenciado en Ciencias de la Computación (2004), Master en Ciencias de la Computación (2007) y PhD. Ciencias Técnicas, especialidad Inteligencia Artificial (2010)

Las principales investigación están relacionadas con las Metaheurísticas Poblacionales para resolver problemas complejos de búsqueda y el Análisis Inteligente de Datos. Cuenta con más de 25 artículos publicados en revistas de impacto, así como la participación en eventos científicos internacionales.

Actualmente es docente e investigador de la Universidad Técnica Estatal de Quevedo, en la facultad de Ciencias de la Ingeniería. <https://www.scopus.com/authid/detail.uri?authorId=15122856600>

Pavel Novoa Hernández

Ingeniero Informático (2007), Master en Ciencias de la Computación e Inteligencia Artificial (2010) y Doctor en Tecnologías de la Información y la Comunicación (2013).

Sus principales investigaciones están relacionadas con la diseño y aplicación de algoritmos evolutivos auto-adaptativos a problemas dinámicos de optimización. Cuenta con varios artículos publicados en revistas de impacto, así como la participación en eventos científicos internacionales.

Actualmente es docente e investigador de la Escuela de Ciencias Empresariales, Universidad Católica del Norte, Coquimbo, Chile. <https://www.scopus.com/authid/detail.uri?authorId=36005378400>

Byron Wladimir Oviedo Bayas

Docente – Director de Investigación UTEQ, Docente en la Facultad de Ciencias de la Ingeniería, Docente titular a tiempo completo, su formación es de Tecnólogo Programador Escuela Politécnica del Ejército. Ingeniero en Sistemas e Informática. Escuela Politécnica del Ejército. Diplomado Superior en Diseños Pedagógicos Universitarios, Universidad Técnica Estatal de Quevedo. Diplomado de Docencia en Ambientes Virtuales, Universidad Autónoma de Manizales. Diplomado Internacional en Tecnologías de la Comunicación e Información, Asociación de las Naciones Unidas en Venezuela. Master en Eléctrica Mención Conectividad y Redes de Telecomunicaciones Escuela Politécnica Nacional. Doctor

