

The background of the banner features several faint technical plots. On the left, there is a scatter plot showing a cluster of points. In the center, a line graph compares 'mCPSO' (solid line) and 'mQSO' (dashed line) over an x-axis from 0 to 45 and a y-axis from 0 to 10. On the right, another line graph compares 'wPSO', 'wPSOOPT', and 'wSOOPT' over an x-axis from 0 to 45 and a y-axis from 0 to 5.0. The text is overlaid on these plots.

PSO MULTI-ENJAMBRE EN PROBLEMAS DINÁMICOS DE OPTIMIZACIÓN: UNA GUÍA DE APLICACIÓN.



Universidad de Holguín "Oscar Lucero Moya"
Facultad de Informática y Matemática

PSO MULTI-ENJAMBRE EN PROBLEMAS DINÁMICOS DE OPTIMIZACIÓN: UNA GUÍA DE APLICACIÓN.

Trabajo de diploma para optar por el título de Ingeniero en Informática

Autora: Elsy Ricardo Santiago

Tutor: Ing. Pavel Novoa Hernández

Holguín, Cuba
Junio, 2009

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora de este trabajo y autorizo al Departamento de Informática de la Universidad de Holguín “Oscar Lucero Moya” para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los 20 días del mes de junio de 2009.

Elsy Ricardo Santiago

Ing. Pavel Novoa Hernández

OPINIÓN DEL USUARIO DEL TRABAJO DE DIPLOMA

El Trabajo de Diploma, titulado _____,
fue realizado en nuestra entidad _____. Se
considera que, en correspondencia con los objetivos trazados, el trabajo
realizado le satisface:

Totalmente

Parcialmente en un _____ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los
beneficios siguientes:

Como resultado de la implantación de este trabajo se reporta un efecto
económico que asciende a _____ MN y/o _____ CUC.

Y para que así conste, se firma la presente a los _____ días del mes de
_____ del año _____

Nombre del representante de la entidad: _____

Cargo: _____

Firma y cuño

OPINIÓN DEL TUTOR SOBRE EL TRABAJO DE DIPLOMA

Título: PSO Multi-enjambre en problemas dinámicos de optimización: una guía de aplicación.

Autora: Elsy Ricardo Santiago

Durante el tiempo de desarrollo de la investigación la estudiante Elsy Ricardo Santiago cumplió con todas las tareas planificadas. Se debe destacar la dedicación presentada por la misma y su aprovechamiento de las jornadas de trabajo, los medios puestos a su disposición, así como la asimilación de las sugerencias hechas por parte de su tutor. Es de destacar también la responsabilidad con que asumió en todo momento cada tarea y el sólido conocimiento adquirido durante la realización de las mismas.

Con el desarrollo de esta guía de aplicación de algoritmos PSO Multi-enjambre en ambientes dinámicos no solo se da respuesta a la problemática planteada a la estudiante, sino que además se logra contribuir a un área del conocimiento relativamente reciente y muy novedosa, sobre la que su tutor se encuentra realizando su doctorado.

Por lo expresado anteriormente, considero que la estudiante cuenta con los requisitos necesarios para ejercer como Ingeniero Informático; y propongo que se le otorgue a este Trabajo de Diploma la máxima calificación.

Y para que así conste, firmo la presente a los 20 días del mes de junio de 2009.

Ing. Pavel Novoa Hernández

Tutor

Pensamiento



«Nadie es tan sabio que no necesite aprender más, y nadie es tan ignorante que no pueda enseñar algo a los demás.»

A. Frasso

Agradecimientos

A Dios por la vida y fuerzas para mirar siempre adelante...

A mi madre que con tanto amor y cariño ha sabido guiarme y hacer de mí la persona que soy, por toda la confianza que me da y su sacrificio incondicional para llegar a ser una profesional.

A mi padre y amigo que con mucho amor me aconseja y me escucha para ayudarme a ser mejor cada día, por toda su preocupación y creer siempre en mí.

A mi segundo padre Oscarito, siempre preocupado por mi bienestar y por todas sus atenciones para poder culminar mis estudios.

A mi tutor Pavel Novoa Hernández por todo el apoyo y dedicación en este trabajo, por su paciencia y deseos de enseñarme, gracias a tus conocimientos, amistad y comprensión pude terminar la investigación.

A TODA mi familia, mis abuelos, Delia e Isnorys por su cariño y cuidados, mis primas, Arlenys que tanta fuerza de voluntad y ánimo me dio cuándo más la necesitaba, por la seguridad que me enseñó a tener en lo que hago y a Arletys por su disposición y apoyo, A Leticia y Maribel por su cariño.

A Beatriz González por ser una excelente amiga y compañera de estudio.

A Iyatne y Yaima por ignorar la distancia para ayudarnos y reírnos.

A mis amistades de la infancia por existir y compartir tantas etapas de nuestra vida.

A mis padrinos Norma y Diosmede por su apoyo, cariño y preocupación.

Quisiera agradecer a todas aquellas personas que de una forma u otra influyeron a lo largo de mi formación como estudiante (Elena Álvarez que me enseñó a andar por el camino del conocimiento).

A mis profesores y compañeros de la Universidad.

Dedicatoria

A Dios por existir

A mis padres que los amo mucho

A Oscarito por tenerme como una hija

A Eve que es una razón más para vivir

A mis hermanos

A mi familia

Resumen

Esta investigación trata sobre la optimización de problemas dinámicos (PDO) a través de técnicas computacionales heurísticas, específicamente técnicas basadas en Optimización con Enjambre de Partículas (PSO) multi-enjambre. Varios fenómenos de la vida real pueden ser tratados como PDOs (ej. el sistema de semáforos de una ciudad, la variación de las ventas en un período o el control automático de un proceso industrial). Estos problemas se caracterizan por presentar funciones objetivos que varían con el tiempo, por lo que las técnicas para optimizarlos deben tratar de adaptarse a los cambios, en vez de re-optimizar constantemente. Una variante bastante eficaz en los últimos años ha sido PSO multi-enjambre. Sin embargo, la mayoría de las investigaciones que aplican PSO multi-enjambre a PDO son empíricas. En ellas se analiza el comportamiento de uno o varios algoritmos frente a determinadas instancias de un PDO artificial. Muy pocos trabajos analizan la robustez de estos algoritmos frente a contextos diferentes, por lo que se dificulta la selección y posterior aplicación de un algoritmo de este tipo en problemas reales. En ese sentido, la presente investigación pretende solucionar esta problemática a partir de una guía de aplicación sustentada en un amplio estudio experimental que compara y analiza estadísticamente los principales exponentes de PSO multi-enjambre frente a diferentes PDO.

Abstract

This investigation is about the optimization of dynamic problems (PDO) through computational heuristics based on Particle Swarm Optimization (PSO), specifically multi-swarm variants. Several real life phenomena can be treated as PDOs (e.g. semaphore systems in a city, the sales variation in one period, automatic control of an industrial process, etc.). This kind of problems contains objective functions that vary with time, for that reason the optimization techniques used to solve them should try to adapting to the changes, instead of constantly re-optimizing. A quite effective variant in the last years has been multi-swarm PSO. However, the investigations about PSO multi-swarm applied to PDO are mostly empiric. In those works the algorithm behavior is analyzed with specific instances of an artificial PDO. Very few works analyze the robustness of these algorithms in front of different contexts, that's why the selection and later application of an algorithm in real problems becomes difficult. In that sense, the present investigation seeks to solve this problem with an application guide supported on a wide experimental study that statistically compares and analyzes the principal exponents of multi-swarm PSO in front of different PDO.

Índice

Introducción.....	1
CAPÍTULO 1. Optimización en ambientes dinámicos	7
1.1 Optimización.....	7
1.1.1 Problemas estacionarios.....	8
1.1.2 Problemas dinámicos	9
1.2 ¿Qué varía en un PDO?.....	10
1.2.1 Posición	10
1.2.2 Valor	11
1.2.3 Forma de los óptimos	12
1.3 Problemas tipo dinámicos	13
1.3.1 Movimiento de Picos.....	13
1.4 Medidas de rendimiento	16
1.4.1 Medidas fuera de línea	17
1.5 Conclusiones.....	18
CAPÍTULO 2. Técnicas PSO multi-enjambre.....	19
2.1 Metaheurísticas	19
2.2 Algoritmo PSO.....	21
2.2.1 Factor de constricción.....	25
2.3 PSO en ambientes dinámicos	25
2.3.1 Memoria desactualizada.....	26
2.3.2 Pérdida de diversidad	26
2.4 PSO Multi-Enjambre.....	29
2.4.1 Trabajos relacionados	30
2.4.2 Enfoque de Blackwell - Branke	31
2.5 Teorema “No Free Lunch”	32
2.6 Conclusiones.....	32
CAPÍTULO 3. Estudio experimental.....	33
3.1 Algoritmos implementados	33
3.2 Técnicas estadísticas y herramientas utilizadas.....	37
3.2.1 Herramientas empleadas para la implementación y análisis	37
3.3 Aspectos generales de los experimentos	38
3.4 Experimentos	40
3.4.1 Efecto de variar la configuración multi-enjambre.....	40
3.4.2 Efecto de variar la severidad de cambio s	45
3.4.3 Efecto de variar el número de óptimos (p).....	46
3.4.4 Efecto de variar la frecuencia de cambio Δe	49
3.5 Conclusiones.....	51

CAPÍTULO 4. Guía de aplicación	52
4.1 Guía de aplicación.....	52
4.1.1 Severidad.....	52
4.1.2 Cantidad de óptimos	53
4.1.3 Frecuencia de cambio	54
4.2 Valoración de sostenibilidad.....	55
4.3 Conclusiones.....	58
Conclusiones generales	59
Recomendaciones.....	61
Anexos	I

Índice de tablas

Tabla 3-1. <i>Parámetros escenario 2 Movimiento de Picos (MPB)</i>	38
Tabla 3-2. <i>11 Configuraciones posibles de la población</i>	41
Tabla 3-3. Error fuera de línea para diferentes valores de S (severidad del cambio).....	45
Tabla 3-4 Error fuera de línea para diferentes valores de p (cantidad de óptimos).	47
Tabla 3-5. Variación de la frecuencia de cambio. (Número de evaluaciones)....	50

Índice de figuras

Figura 1-1 Variación de la localización del óptimo en el tiempo (posición)[8].....	11
Figura 1-2 Variación del valor del óptimo (altura)[6].....	12
Figura 1-3 Variación en la forma de los óptimos (ancho)[6].....	12
Figura 1-4 Integrando la variación de los 3 componentes	13
Figura 1-5. Movimiento de picos con un espacio de búsqueda de 2 dimensiones y 10 picos.	16
Figura 2-1. <i>Algoritmo PSO estándar</i>	24
Figura 2-2. Secuencia del comportamiento del fenómeno colapso lineal [39]....	27
Figura 2-3. Analogía del átomo[39].	29
Figura 3-1 Diagrama de clases	36
Figura 3-2 Metodología para el análisis estadístico	39
Figura 3-3. Variación del error fuera de línea para diferentes configuraciones de la población.	44
Figura 3-4. <i>Comportamiento de la severidad del cambio</i>	46
Figura 3-5 Variación de la cantidad de óptimos	49
Figura 3-6. <i>Variación de la frecuencia de cambio</i>	51

Introducción

Varios son los fenómenos que en la vida real se pueden analizar a través de un modelo matemático: el crecimiento de la población mundial, la selección de la mejor ruta en una ciudad, el comportamiento de las ventas en un período, etc. Una cantidad considerable de estos fenómenos, a través de los modelos apropiados, se convierten en problemas matemáticos que tienen como propósito encontrar el conjunto de condiciones adecuadas que proporcione los mejores resultados en una situación determinada, por ejemplo, la ganancia máxima, el costo mínimo, la satisfacción de los usuarios, entre otros. Tales problemas se conocen como problemas de optimización y durante décadas han sido ampliamente tratados con resultados muy favorables, dando lugar al surgimiento de nuevas ramas dentro de las matemáticas como la Investigación de Operaciones.

La mayor parte de los trabajos relacionados con la solución de problemas de optimización tratan problemas estacionarios, es decir, aquellos cuyas variables, relaciones y objetivos permanecen invariables en el tiempo, lo cual no siempre es así, ya que los procesos que se representan a través de estos modelos matemáticos (ej. sociales, económicos, industriales) son de naturaleza cambiante. Aunque pudiera tenerse información determinística sobre cuáles serán los valores posibles de aquellos elementos del modelo que cambiarán (coeficientes, dimensiones, relaciones, etc.), puede darse el caso de que no, y exista cierta incertidumbre relacionada con los mismos. Evidentemente, esto último aumenta la complejidad del problema objeto de análisis, el cual puede estar afectado ya por su alta dimensionalidad, funciones objetivo con expresiones complicadas, etc. Obsérvese que la dificultad principal radica ahora en la propia estructura del modelo, la cual no es siempre la misma durante el

tiempo. A estos problemas se les conoce como Problemas Dinámicos de Optimización (PDO).

El interés sobre el estudio de los PDO se ha incrementado en las últimas dos décadas con la aparición de trabajos que emplean metaheurísticas como técnicas de solución, donde sobresale el uso de los Algoritmos Evolutivos: EAs¹. El auge de estas propuestas basadas en métodos heurísticos no se justifica solo por la naturaleza dinámica del ambiente, sino por la complejidad inherente a los problemas de optimización. Se ha podido comprobar que los principales problemas de optimización pertenecen a la clase de complejidad computacional denominada NP, en la que se incluyen aquellos problemas para los que no se conoce un algoritmo polinomial de resolución.

Aunque lo ideal sería que para cada problema NP se pudiera encontrar una solución óptima mediante algoritmos exactos que requieran solamente una cantidad moderada de recursos, desafortunadamente, estos problemas no admiten este tipo de enfoque, haciendo necesario el desarrollo de técnicas que obtengan soluciones lo “suficientemente buenas” con un gasto de recursos computacionales (por lo general tiempo) dentro del rango definido por el usuario o decisor. Estas técnicas se conocen usualmente como heurísticas. El empleo exitoso en los últimos años de procedimientos heurísticos no solo ha contribuido al desarrollo científico de este campo de investigación, sino que ha despertado interés en cómo aprovechar lo mejor de cada técnica de manera que se pueda aplicar a otros problemas enmarcados en contextos diferentes. Surgen así las denominadas metaheurísticas Fred Glover en 1986 [1], que pueden ser definidas informalmente como “estrategias inteligentes para diseñar o mejorar procedimientos heurísticos muy generales con un alto rendimiento” [2].

¹ Siglas en inglés de Evolutionary Algorithms.

Una metaheurística muy popular en los últimos años ha sido la Optimización con Enjambres de Partículas (PSO)². La cual es una técnica de optimización bioinspirada que en el contexto de los PDO ha sido ampliamente utilizada, ejemplo de ello lo constituyen los trabajos[3-6]. El interés en el empleo de esta metaheurística se debe principalmente a las ventajas que brinda PSO: fácil implementación y un balance apropiado en dos aspectos importantes en toda metaheurística: la explotación (intensificación de la búsqueda) y la exploración (diversificación de las soluciones). Sin embargo, se ha podido observar que en ambientes dinámicos las propuestas basadas en PSO requieren ciertas modificaciones debido a la naturaleza variable del entorno, por lo general se han incluido operadores que favorecen la diversidad del conjunto de soluciones actual, o que evitan una convergencia innecesaria en un determinado momento.

Una variante de PSO que ha tenido bastante seguimiento en los últimos años es la denominada multi-enjambre (multi-swarm). Básicamente, este enfoque se sustenta en la existencia de varios enjambres que favorecen la diversidad en el algoritmo y por consiguiente, el seguimiento de los óptimos en aquellos PDO multimodales.

La mayoría de los trabajos que aplican PSO en ambientes dinámicos son empíricos, en especial aquellos basados en el enfoque multi-enjambre. En estos trabajos se presenta habitualmente uno o varios algoritmos que intentan resolver determinadas instancias de un tipo determinado de PDO (ej. el problema test Movimiento de Picos). En muy pocos trabajos se analiza la robustez de las propuestas frente a contextos con características diferentes al de los problemas para las que originalmente fueron creadas. Principalmente por esta forma aislada en que aparecen las propuestas y lo específico de sus estudios, es que no se cuenta en la actualidad con algún trabajo que sistematice y caracterice el

² Siglas en inglés de Particle Swarm Optimization.

comportamiento de estos algoritmos frente a diferentes situaciones. Además, con esta insuficiencia se dificulta la selección y posterior aplicación de un algoritmo de este tipo en problemas reales como: el ajuste automático de parámetros en algún proceso industrial, la mejor planificación de un horario en un determinado período, etc. Aunque lo ideal sería tener una caracterización de todas las variantes de PSO existentes, por razones evidentes de tiempo, esta investigación se centrará solo en aquellas basadas en el enfoque multi-enjambre.

La problemática descrita con anterioridad conlleva a plantearse como **problema de investigación** la siguiente interrogante: ¿Cómo facilitar el proceso de selección y aplicación de los algoritmos PSO Multi-enjambre existentes en ambientes dinámicos?

El problema anteriormente definido se enmarca en el **objeto de estudio**: La optimización de problemas dinámicos aplicando metaheurísticas basadas en PSO.

Para resolver este problema se plantea el siguiente **objetivo**: Elaboración de una guía de aplicación que sistematice las características y aplicaciones de los principales exponentes de la optimización con enjambre de partículas frente a diferentes problemas dinámicos de optimización.

Este objetivo delimita el **campo de acción**: Experimentación de metaheurísticas basadas en PSO empleadas en la optimización de problemas dinámicos.

Para guiar la investigación de este trabajo se enuncia la siguiente **hipótesis**: Una guía de aplicación soportada en un estudio experimental que compare y realice los análisis estadísticos apropiados de los principales exponentes de PSO multi-enjambre frente a diferentes problemas dinámicos de optimización, favorecerá el proceso de selección y aplicación de dichos exponentes.

La investigación transcurrirá a través de las siguientes **tareas**:

1. Elaboración de los fundamentos teóricos relacionados con la optimización de problemas dinámicos.
2. Descripción de los principales exponentes de PSO multi-enjambre para ambientes dinámicos.
3. Implementación de los exponentes seleccionados.
4. Desarrollo de experimentos que permitan la comparación y caracterización del comportamiento de los algoritmos seleccionados.
5. Diseño de una guía de aplicación.
6. Valoración de sostenibilidad de los resultados de esta investigación.

Para cumplimentar estas tareas se han empleado **métodos de investigación científica** teóricos y empíricos.

Entre los **métodos teóricos** empleados se encuentran:

Análisis y síntesis: Se utilizó en la elaboración de los fundamentos teóricos, en el procesamiento de la información y para arribar a las conclusiones de la investigación.

Histórico y lógico: Usado para el estudio y comprensión de la optimización de problemas dinámicos mediante metaheurísticas basadas en PSO.

Método hipotético-deductivo: Empleado para la elaboración de la hipótesis de este trabajo, la cuál será examinada y demostrada por la investigación realizada.

Los **métodos empíricos** que se usaron son:

Revisión de documentos: Este método fue de suma importancia en la comprensión de los algoritmos basados en PSO multi-enjambre y su aplicación en PDOs.

Experimento: A través de este método, se seleccionaron las variables y aspectos a tener en cuenta para medir el comportamiento de los algoritmos, lo cual permitió diseñar los experimentos necesarios para llegar a conclusiones evaluativas respecto a la efectividad de cada una de estas técnicas.

Estadístico-matemático: Este método se empleó con el propósito de analizar las muestras obtenidas a través de un análisis de ANOVA simple. Además de permitir el procesamiento de los datos y establecer valores de estimación que determinen diferencias en el resultado de las muestras.

Finalmente, este trabajo está organizado de la forma siguiente: en el Capítulo 1: *Optimización en ambientes dinámicos*, se abordan conceptos y definiciones que sirven de fundamentos teóricos a esta investigación. Un propósito similar tiene el Capítulo 2: *Técnicas PSO multi-enjambre*, que parte de aspectos generales sobre PSO estándar y hace énfasis en la variante multi-enjambre aplicada a problemas dinámicos. Por su parte, el Capítulo 3: *Estudio experimental*, detalla la experimentación desarrollada. El Capítulo 4: *Guía de aplicación*, describe la guía de aplicación obtenida, así como la valoración de sostenibilidad de este resultado. Más adelante, la sección *Conclusiones* está dedicada a las principales conclusiones derivadas de esta investigación, y los trabajos futuros se proponen en la sección *Recomendaciones*.

CAPÍTULO 1.

Optimización en ambientes dinámicos

En este capítulo se presentan los fundamentos teóricos relacionado con los Problemas Dinámicos de Optimización (PDO). Una definición formal de los problemas de optimización y conceptos básicos que lo caracterizan haciendo énfasis en las características de los problemas dinámicos y los tipos de PDO existentes. Por otro lado se relacionan los problemas test para tratar los PDO, específicamente el del movimiento de picos (MPB) y se abordan también las principales medidas de desempeño que evalúan el rendimiento de los algoritmos en ambientes dinámicos.

1.1 Optimización

A groso modo, optimizar es encontrar la mejor solución (entre varias) para un problema en específico[5]. A diario se optimiza, tómesese como ejemplos: el proceder de una compañía que se propone conocer el personal y medios necesarios que le permitan optimizar el uso y distribución de sus recursos disponibles, para obtener un máximo de ganancias y un costo mínimo de inversión. Otro ejemplo sería lograr la asignación idónea de una tripulación de

aerolíneas para reducir los vuelos ociosos de sus miembros. Como un tercer y último ejemplo, encontrar la mejor combinación de componentes para producir un material más resistente, o en el área de la investigación farmacéutica para demostrar la superioridad de una droga con respecto a otra, en dependencia de sus composiciones químicas.

Existen varios tipos de problemas de optimización atendiendo a diversos aspectos (dimensiones):

Cantidad de objetivos: De un solo objetivo o multi-objetivo (más de un objetivo).

Cantidad de óptimos globales: Unimodales (un solo óptimo global) o multimodales (más de un óptimo global).

Variabilidad en el tiempo: Estacionarios (permanecen invariables en el tiempo) o Dinámicos (permanecen invariables en determinados intervalos de tiempo).

Estas dimensiones se pueden considerar ortogonales, es decir, todo problema tiene una combinación de estas características. Ejemplo, *estacionario*, *multimodal*, *mono-objetivo*. Sin embargo, a menos que se establezca lo contrario, en este trabajo se tratarán solo problemas dinámicos y de un solo objetivo, sin importar la cantidad de óptimos globales.

1.1.1 Problemas estacionarios

Como se mostraba anteriormente, si el modelo de un problema de optimización, en especial su función objetivo, permanece invariable en el tiempo, entonces se está en presencia de un problema de optimización estacionario o estático. Estos problemas pueden definirse matemáticamente como sigue:

Definición 1. Problema de optimización estacionario. Sea el espacio de búsqueda cerrado Ω , una función de aptitud $f:\Omega \rightarrow \mathbb{R}$, y la relación de

comparación " \succ " $\in \{\leq, \geq\}$. El objetivo es determinar el conjunto del óptimo global $\mathcal{X} \subseteq \Omega$ definido como:

$$\mathcal{X} = \{x^* \in \Omega \mid f(x^*) \succ f(x), \forall x \in \Omega\} \blacksquare$$

Note que encontrar una solución exacta para este problema es una meta que puede tornarse en extremo difícil, (ej. si $\Omega \subset \mathbb{R}^n$). Por lo que habitualmente, en problemas reales, se trata de buscar una solución candidata x' con $f(x')$ tan cerca como sea posible a $f(x^*)$. Un ejemplo sencillo para ilustrar la definición anterior es el siguiente problema:

Ejemplo 1 (Problema con función esfera)[6] Encontrar en el espacio de búsqueda $\Omega = [-100.00, 100.00] \times [-100.00, 100.00] \subset \mathbb{R}^2$ el conjunto $\mathcal{X} = \{x^* \in \Omega \mid f(x^*) \leq f(x), \forall x \in \Omega\}$, si $f(x) = x_1^2 + x_2^2$ y $x = (x_1, x_2)$.

Evidentemente, se trata de un problema de minimización donde " $>$ " \equiv " $<$ " y el conjunto \mathcal{X} está formado por un solo elemento $x^* = (0.00, 0.00)$. Además, es unimodal por existir solamente un óptimo local y un óptimo global (en este caso el mismo).

1.1.2 Problemas dinámicos

Puede decirse entonces que los PDO constituyen una generalización de los problemas de optimización en los que las variables relevantes del problema, la función objetivo, la correspondiente instancia del problema, las restricciones, los datos y/o las estructuras del problema pueden cambiar en el tiempo[6] y formalmente se pueden formular así:

Definición 2. Problema dinámico de optimización. Sea el espacio de búsqueda cerrado Ω , una función de aptitud $f^{(t)}: \Omega \rightarrow \mathbb{R} (t \in \mathbb{N}_0)$, y la relación de comparación " \succ " $\in \{\leq, \geq\}$. El objetivo es determinar el conjunto del óptimo global $\mathcal{X}^{(t)} \subseteq \Omega$, en cada instante de tiempo t , definido como:

$$\mathcal{X}^{(t)} = \{x^* \in \Omega \mid f^{(t)}(x^*) \geq f^{(t)}(x), \forall x \in \Omega\} \blacksquare$$

Obsérvese que esta definición difiere de la anterior (problema estacionario), en que ahora se tiene en cuenta el factor tiempo. De hecho, esta variable es la que más influye en un PDO, debido a que en un determinado instante $t > 0$ puede que $f^{(t)} \neq f^{(t-1)}$, con la consecuente variación de la posición y valor de las soluciones óptimas. Es posible entonces conjeturar que un PDO puede ser *construido* a partir de un problema estacionario incluyendo a la dimensión tiempo. La siguiente sección analiza esta suposición.

1.2 ¿Qué varía en un PDO?

Un problema de optimización dinámica requiere encontrar una solución de calidad aceptable, posiblemente cercana al óptimo pero en un ambiente que cambia continuamente en el tiempo, ya sea de forma continua o discreta. Estos cambios califican a los PDO como un problema no estacionario que busca seguir la evolución de las regiones con importancia durante el tiempo, además de asumir que los cambios en el entorno pueden significar variaciones en la función objetivo, en la instancia del problema o en las restricciones que debe cumplir la solución [7]. Nótese entonces como la presencia del factor tiempo, característica especial de los PDO, provoca determinadas variaciones en los mismos, ya sea una sucesión de transformaciones en las coordenadas y localización del óptimo, así como cambios en el ancho y altura de los óptimos. A continuación se describen las posibles variaciones de un PDO que se pueden representar visualmente (al menos para tres dimensiones) en el problema del movimiento de picos que se tratará en la sección 1.3.1.

1.2.1 Posición

Uno de los parámetros sensibles a cambios en un PDO es la localización del óptimo, la cual está relacionada con las soluciones de las instancias previas,

característica que las técnicas de optimización pudieran aprovechar para realizar una búsqueda más eficiente. Entiéndase por variación de la posición, cuando un óptimo global, transcurrido cierto valor de tiempo, se desplaza de lugar. Obsérvese en la Figura 1-1 como en una determinada instancia del problema para $t > 0$ $x^{(t)} \neq x^{(t-1)}$. Este desplazamiento de la localización del óptimo tiene mucha relación con el grado de severidad con que ocurre el cambio, aspecto a analizar más adelante por su influencia en este tipo de problema.

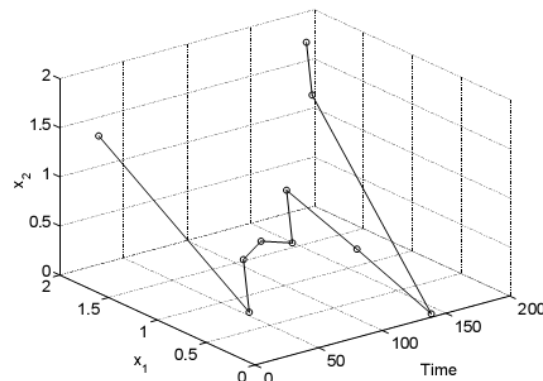


Figura 1-1 Variación de la localización del óptimo en el tiempo (posición)[8]

1.2.2 Valor

Otro de los aspectos que varían como consecuencia de los cambios que se suceden en un PDO es el valor de la función de aptitud. Éste valor para ambientes estáticos es el mismo para cada óptimo en la nueva generación, sin embargo para ambientes dinámicos la función objetivo puede modificar su valor de aptitud en cada intervalo de tiempo. Comúnmente a este valor se le conoce como altura del óptimo. Por ejemplo, véase en la Figura 1-2, como varía la altura de los óptimos para un instante posterior de tiempo.

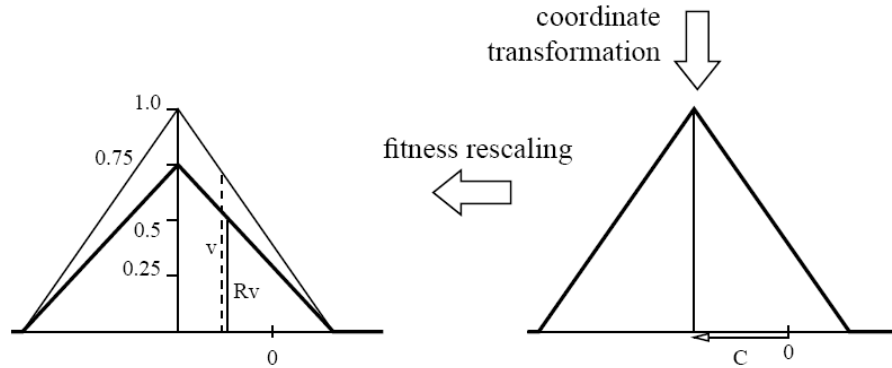


Figura 1-2 Variación del valor del óptimo (altura)[6]

1.2.3 Forma de los óptimos

De la misma manera que los cambios que acontecen en un PDO provocan variaciones en la posición de los óptimos y en el valor de la función, también tiende a modificar la forma de los óptimos. Dígase forma a la anchura del pico, que como se observa en la Figura 1-3 puede variar en dependencia del cambio.

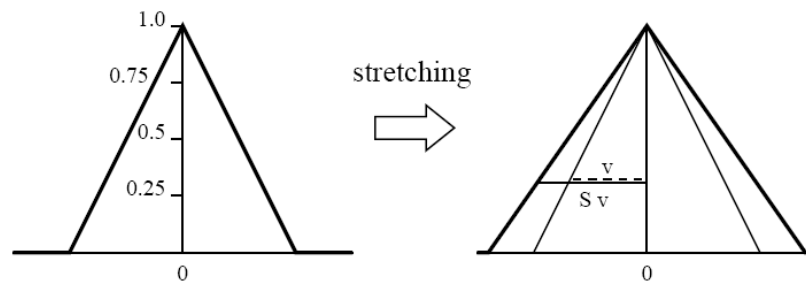


Figura 1-3 Variación en la forma de los óptimos (ancho)[6]

Los elementos explicados anteriormente no aparecen de forma aislada en un PDO. De hecho, es usual que estos se combinen como se muestra en la Figura 1-4. Precisamente, estas variaciones en la posición, altura, y anchura han servido de inspiración en la simulación de PDO como se verá en la próxima sección.

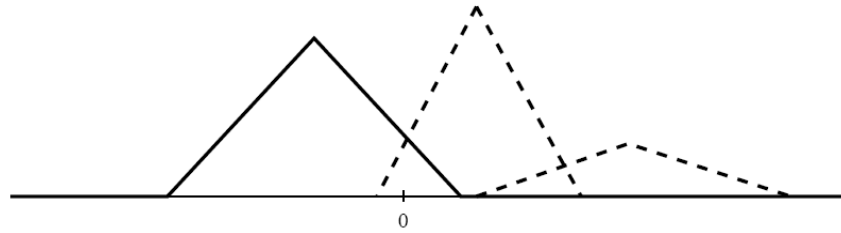


Figura 1-4 Integrando la variación de los 3 componentes

1.3 Problemas tipo dinámicos

Un problema tipo (PT) es en general, un problema artificial que sirve para estudiar el comportamiento de un algoritmo. En ese sentido, la importancia que reviste un PT se soporta en dos aspectos fundamentales: 1) a través de la simulación evitan las consecuencias negativas que pueden ocurrir en problemas reales y 2) favorecen la comparación de propuestas en la comunidad científica relacionada con este campo.

Hasta ahora el número de PT en ambientes dinámicos es relativamente pequeño, aunque existen trabajos que proponen generadores de PT como [9, 10]. Sin embargo, entre los más conocidos se encuentran el *Problema de la Mochila Variable en el Tiempo* [11, 12], el de la *Parábola Móvil* [13], *Bit-matching Dinámico* [14, 15], el de la *Planificación Dinámica de Tareas* [16], el generador DF1 introducido por Morrison y De Jong [17].

Quizás el PT más utilizado académicamente sea el popular *Movimientos de Picos* propuesto por Branke [18], el cual será analizado con más detalle en la sección siguiente.

1.3.1 Movimiento de Picos

El Movimiento de Picos (*MPB*) [18, 19] ha resultado de gran interés en la comunidad científica por ser un PT bastante flexible, y que tiene en cuenta los

elementos de la solución óptima que pueden variar con el tiempo: posición, valor, etc. Asimismo, es posible establecer la frecuencia en que ocurrirán los cambios, tomando como unidad de tiempo una evaluación en la función de aptitud.

Este PT define una función de aptitud que cambia con el tiempo y posee un número predeterminado de picos (varios óptimos). A su vez, cada pico constituye un óptimo global que tiene una determinada altura (valor de la función), ancho y posición, que son variables en el tiempo.

La función básica de prueba de este problema tiene 5 dimensiones, la cual usa parámetros de valores reales y se representa a través de la siguiente fórmula:

$$f(\vec{x}^{(t)}) = \max_{i=1,\dots,5} \frac{H_i^{(t)}}{1 + W_i^{(t)} \sum_{j=1}^5 (x_j - X_{ij}^{(t)})^2}$$

Donde $H_i^{(t)}, W_i^{(t)}, X_i^{(t)}$ son la altura, ancho y posición del pico i , respectivamente.

De modo más formal un cambio puede ser descrito de la siguiente forma:

$$\sigma \in \mathbb{N}(0,1)$$

$$H_i^{(t)} = H_i^{(t-1)} + \textit{severidad_altura} * \sigma$$

$$W_i^{(t)} = W_i^{(t-1)} + \textit{severidad_ancho} * \sigma$$

$$\vec{X}_i^{(t)} = \vec{X}_i^{(t-1)} + \vec{v}$$

El vector de cambio \vec{v}_i calificado como una combinación lineal de un vector aleatorio \vec{r} y el vector de cambio anterior $\vec{v}_i^{(t-1)}$ permiten normalizar el desplazamiento s como se muestra en la siguiente ecuación.

$$\vec{v}_i^{(t)} = \frac{s}{\left| \vec{r} + \vec{v}_i^{(t-1)} \right|} \left((1 - \lambda) \vec{r} + \lambda \vec{v}_i^{(t-1)} \right)$$

La manera de modelar la dinámica de este problema consiste en variar cada Δe evaluaciones a $H_i^{(t)}$, $W_i^{(t)}$ y $X_i^{(t)}$ según una variable gaussiana aleatoria [20]. De esta manera el parámetro s permite controlar la severidad del cambio o determinar cuánto ha sido desplazada la función de aptitud de su posición actual producto al dinamismo presente. Por su parte, Δe determina la frecuencia de cambio, es decir, indica con que periodicidad ocurren los cambios o cuantas generaciones permanecen estáticas entre los cambios [21].

Si se quiere hacer más compleja la función, se puede incrementar el número de dimensiones, el número de picos o introducir una función base más compleja.

La facilidad con que cuenta este PT para ajustar y modificar parámetros como el número de picos, las dimensiones, la frecuencia de cambio, la severidad de cambio, etc., le confieren una importancia especial para comparar estrategias de solución y resultados, así como simular problemas del mundo real con el objetivo de obtener las mejores soluciones.

En la Figura 1-5 se muestra una instancia del MPB, específicamente con dos dimensiones y diez picos. Actualmente este PT cuenta con 3 escenarios, que poseen configuraciones diferentes en los parámetros.

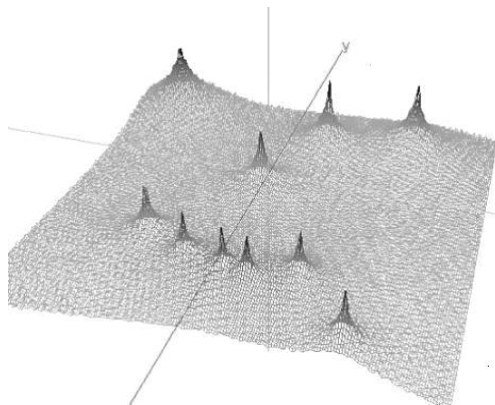


Figura 1-5. Movimiento de picos con un espacio de búsqueda de 2 dimensiones y 10 picos.

1.4 Medidas de rendimiento

En la optimización de problemas dinámicos medir el desempeño de los algoritmos resulta más complejo que en los estáticos, debido principalmente a la variabilidad que presenta la función de aptitud. En sentido general, las medidas de rendimiento permiten estudiar a los algoritmos frente a los problemas que son aplicados, incluso, pueden asistir al propio algoritmo durante su ejecución en el ajuste de parámetros y en la toma de decisiones relacionada con determinadas reglas implementadas por este.

Existen dos posibles clasificaciones de las medidas de rendimiento en los algoritmos poblacionales (ej. EAs) [22]. La primera está relacionada con la información utilizada y se distinguen dos clases:

- Medidas basadas en la aptitud (orientadas al estudio global del algoritmo)
- Medidas genotípicas o fenotípicas (orientadas a los individuos de las poblaciones: soluciones candidatas)

Una segunda clasificación tiene en cuenta el conocimiento previo disponible utilizado en la medición:

- La solución óptima es conocida (usualmente solo en los casos de problemas tipos).
- Se conoce solo el valor de la solución óptima.
- No se tiene ningún conocimiento global.

En [22, 23] existe un excelente estudio de estas medidas que el lector interesado puede consultar para más detalles sobre el tema. Sin embargo, en este trabajo solo se considerará como medida de rendimiento el *error fuera de línea*, la cual será tratada en la siguiente sección.

1.4.1 Medidas fuera de línea

De manera general, las medidas *fuera de línea* (*off-line measures*) dan una idea del comportamiento global del algoritmo, pues representan promedios a la larga de determinados criterios (ej. valor óptimo, error, etc.) tomando como base las generaciones. A diferencia de las medidas *en línea* (*on-line measures*) que toman en cuenta todas las evaluaciones de la función de aptitud, las *fuera de línea* utilizan al mejor individuo de cada generación.

Para problemas estacionarios, en 1975 De Jong [6] define como *rendimiento fuera de línea* la siguiente expresión:

$$perf_{offline} = \frac{1}{G} \sum_{i=1}^G f(x_{best}^{(i)})$$

Donde G representa el número de generaciones del algoritmo y $f(x_{best}^{(i)})$ la aptitud del mejor individuo de la población (mejor solución encontrada por el algoritmo en la generación i). Una variante de esta medida que representa el error de la mejor solución con respecto al óptimo global es la siguiente:

$$error_{offline} = \frac{1}{G} \sum_{i=1}^G (f(x_{global}) - f(x_{best}^{(i)}))$$

Donde $f(x_{global})$ es la aptitud del óptimo global. Note que a diferencia de $perf_{offline}$, esta última medida requiere que se conozca de antemano el valor del óptimo global.

A pesar de que estas medidas resultan bastante apropiadas en ambientes estacionarios, no lo es así para el caso de los PDOs, sobre todo porque f varía con el tiempo, y las soluciones óptimas en generaciones anteriores pueden no

serlas en las siguientes. Es por eso que Branke[21, 24] redefine estas medidas como:

$$perf_{offline} = \frac{1}{t} \sum_{i=1}^t f^{(i)}(x_{best})$$

$$error_{offline} = \frac{1}{t} \sum_{i=1}^t (f^{(i)}(x_{global}) - f^{(i)}(x_{best}))$$

Estas variantes representan mejor la dinámica del problema. Obsérvese que ahora el promedio contempla a cada instante de tiempo t (evaluación de la función de aptitud), y al mejor individuo de la población en dicho instante. Análogamente a la definición 2, aquí ($t \in \mathbb{N}_0$).

1.5 Conclusiones

De los temas tratados en este capítulo se puede concluir que los problemas dinámicos de optimización difieren de los estacionarios principalmente por la presencia de la dimensión tiempo, la cual influye en los elementos del modelo. Además, el papel que juegan los problemas tipo en el estudio de los algoritmos de optimización dinámica es de suma importancia. Similarmente, las medidas de rendimiento son vitales para el análisis de estas técnicas, especialmente las que realizan una estimación *fuera de línea*. El siguiente capítulo prosigue con los fundamentos teóricos de esta investigación, esta vez con la exposición de las técnicas de optimización basadas en PSO Multi-enjambre.

CAPÍTULO 2.

Técnicas PSO multi-enjambre

Este capítulo está dedicado al establecimiento de los aspectos fundamentales de las metaheurísticas, prestando una especial atención a la optimización con enjambre de partículas. En ese sentido, se analizan las principales dificultades de PSO frente a los ambientes dinámicos, y las principales propuestas de solución encontradas en la literatura consultada. Un especial énfasis se le brinda al enfoque PSO multi-enjambre. Finalmente se expone el popular teorema *No Free Lunch*.

2.1 Metaheurísticas

En los últimos años numerosos procedimientos heurísticos han sido diseñados para resolver los más variados problemas de optimización. Dígase heurístico a la tarea de resolver de un modo inteligente problemas reales utilizando el conocimiento disponible [25], o a la manera de aportar soluciones (o acercarse a ellas) usando una cantidad de recursos (generalmente tiempo) razonable, con el propósito de encontrar una solución óptima según algún criterio de comparación entre ellas, y que además satisfaga las condiciones de factibilidad.

Aunque los mecanismos heurísticos son procedimientos sencillos, adaptables y robustos que se pueden aplicar a un conjunto amplio de problemas, las técnicas heurísticas tradicionales no son lo suficientemente efectivas para tratar problemas complejos de tipo *no estacionarios* debido a la carencia de diferentes estrategias que puedan enfrentarse a los cambios producidos en el entorno, como la variabilidad de los óptimos globales. Esto último complejiza el problema que se desea resolver, el cual ya pudiera estar *complejizado* debido a la presencia de un espacio de búsqueda con una alta dimensionalidad, así como funciones con expresiones difíciles de evaluar.

Como alternativa a la polémica descrita anteriormente y para solventar la carencia de mecanismos que les permitan escapar de los óptimos locales se introducen algoritmos de alto nivel conocidos como metaheurísticas [1]o heurísticas modernas [26, 27], las cuales se consideran como un conjunto de métodos computacionales aproximados que se basan en criterios heurísticos para encontrar soluciones de alta calidad en un intervalo de tiempo mucho menor.

El término metaheurística hace referencia a un procedimiento genérico de alto nivel que guía heurísticas para explorar eficientemente el espacio de soluciones, combinando de forma inteligente diferentes conceptos para explorar y explotar el espacio de búsqueda [28], lo cual justifica el uso de las metaheurísticas poblacionales en los PDO, ya que las mismas al mantener simultáneamente un conjunto de soluciones favorecen la diversidad.

A estas metaheurísticas empleadas anteriormente en problemas estacionarios (ej. algoritmos genéticos), se le han realizado ciertas modificaciones para lograr la adaptación continua, flexibilidad y robustez necesarias en ambientes dinámicos [29], ejemplo de ella es la novedosa y prometedora metaheurística poblacional conocida como optimización con enjambre de partículas, que por su importancia en esta investigación se detalla en la sección que continúa.

2.2 Algoritmo PSO

Encontrar una estrategia capaz de adaptarse continuamente a los cambios en los PDO que no solo permita optimar la solución, sino seguir a los óptimos en el tiempo constituye una tarea difícil que sugiere el uso de técnicas computacionales basadas *SOFT COMPUTING* (Lógica difusa, Algoritmos Evolutivos, otros.)

En ese sentido, los algoritmos *bio-inspirados* resultan bastante apropiados debido a que son sistemas que evolucionan constantemente en el tiempo [30]. Dentro de esta gran familia de algoritmos, existe una subclase denominada Inteligencia de Enjambres (SI³). En SI los algoritmos se apoyan en un conjunto de técnicas basadas en el estudio del comportamiento colectivo en sistemas auto-organizados y descentralizados (distribuidos) [30]. El principio de operación que rige a las técnicas basadas en SI es el siguiente:

La inteligencia de enjambre es una propiedad que poseen ciertos sistemas biológicos compuestos por agentes sencillos (hormigas, peces, pájaros, etcétera) en los que el comportamiento colectivo viene descrito por la interacción local entre agentes, de tal forma que se obtiene un funcionamiento global coherente [31].

Desde un enfoque algorítmico, un enjambre está compuesto por agentes que establecen relaciones de cooperación para conseguir un objetivo determinado. Cada agente utiliza un conjunto de reglas sencillas (locales) de forma relativamente independiente a los demás (salvo la cooperación entre agentes próximos), en la cual no existe necesariamente un líder que determine la estrategia a seguir.

³ Siglas en inglés de *Swarm Intelligence*.

En estas condiciones, surge una inteligencia colectiva del conjunto de agentes, dando lugar a fenómenos de auto-organización, los cuales constituyen una de las aplicaciones más comunes de la Inteligencia Artificial (IA) por la búsqueda de la solución óptima en problemas de alta complejidad, tanto en espacios continuos como discretos[30].

Dentro de SI, se encuentran tres paradigmas o metaheurísticas principales, a saber:

Colonias de Hormigas (ACO⁴)[32].

Optimización con Enjambre de Partículas (PSO).

Búsqueda por Difusión Estocástica (SDS⁵).

Aunque estas metaheurísticas poseen puntos en común con la gran familia de los Algoritmos Evolutivos (ej. Se basan en poblaciones que evolucionan con el tiempo) es posible notar que ninguna implementa algún operador evolutivo como la mutación o selección. Sin embargo, existe también la tendencia a incluirlos dentro de los EA, sobre a todo a PSO, asumiendo como operador evolutivo a la expresión de velocidad de las partículas.

PSO fue propuesto por primera vez en 1995 por James Kennedy y Russel Eberhat [33], y se inspira en la sociabilidad de los individuos que componen un sistema biológico [34]. Por ejemplo bandadas de pájaros o cardúmenes de peces que exploran un determinado espacio en busca de comida y que poseen un comportamiento influenciado por la experiencia individual y la del líder del enjambre [35].

⁴ Siglas en inglés de Ant Colony Optimization

⁵ Siglas en inglés de Stochastic Diffusion Search

PSO se caracteriza por la fácil implementación y el uso de la evolución de las relaciones sociales como modelo computacional, donde cada partícula representa una solución candidata al problema, la cual cambia su estado al *volar* a través del espacio de búsqueda hasta que se ha encontrado un estado relativamente estable [30].

Cada partícula i es posicionada en el espacio de búsqueda Ω , de manera que sus coordenadas, representadas por el vector x_i , son los valores que toman las variables de decisión del problema. En cada iteración, PSO actualiza las posiciones x_i a través de la velocidad de cada partícula v_i . Por ejemplo, sea $\Omega \subseteq \mathbb{R}^n$, entonces $x_i = [x_{i1}, x_{i2}, \dots, x_{in}]$, y $v_i = [v_{i1}, v_{i2}, \dots, v_{in}]$, con $i \in \mathbb{N}_0, i = 0, 1, \dots, p$, donde $p + 1$ es el tamaño de la población (cantidad de partículas).

En cada *iteración*, las partículas actualizan su velocidad mediante la siguiente fórmula.

$$v_i^{(t+1)} = \omega v_i^{(t)} + \eta_1 r_1 (pbest_i - x_i^{(t)}) + \eta_2 r_2 (gbest - x_i^{(t)}) \quad (1)$$

Donde $v_i^{(t)}$ y $v_i^{(t+1)}$ son vectores que representan la velocidad anterior y actual de la partícula respectivamente. Además $x_i^{(t)}$ representa la posición actual de la partícula, $pbest$ su mejor posición hasta el momento y $gbest$ la mejor posición de todo el enjambre.

El coeficiente $\omega \in [0.0, 1.0]$ es un peso inercial, formulado por Shi y Eberhart 1998 [36], utilizado para controlar el impacto de las velocidades previas en la velocidad actual. Un peso inercial mayor facilita la exploración global para buscar nuevas áreas, mientras que un peso inercial menor tiende a facilitar exploración local para refinar el área de búsqueda actual. Una correcta selección del peso inercial puede proveer un balance entre las habilidades de búsqueda local y global, por lo tanto requiere menores iteraciones para encontrar el punto óptimo.

Por otro lado, η_1 y η_2 son constantes de aceleración, mientras r_1 y r_2 son vectores de números aleatorios en el intervalo $[0.0, 1.0]$.

Por último la posición de la partícula es actualizada de la siguiente forma usando únicamente su velocidad.

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \quad (2)$$

En todos estos casos se almacena también el valor de la función objetivo de adaptación (*fitness*). El algoritmo PSO estándar queda resumido como se muestra en la Figura 2-1.

```
FOR EACH particle  $i$ 
  Randomly initialize  $v_i, x_i = pbest_i$ 
  Evaluate  $f(pbest_i)$ 
   $gbest = \arg \max f(pbest_i)$ 
REPEAT
  FOR EACH particle  $i$ 
    Update particle position  $x_i$  according to eqs...
    Evaluate  $f(x_i)$ 
    //Update personal best
    IF  $f(x_i) > f(pbest_i)$  THEN
       $pbest_i = x_i$ 
    //Update global best
    IF  $f(x_i) > f(gbest_i)$  THEN
       $gbest_i = \arg \max f(pbest_i)$ 
  UNTIL termination criterion reached
```

Figura 2-1. Algoritmo PSO estándar...

2.2.1 Factor de constricción

A pesar de que PSO ha demostrado cierta robustez en la mayoría de los problemas en los que se ha aplicado, se ha podido demostrar que su convergencia no siempre ocurre. Varios trabajos se han propuesto para solucionar esta dificultad, entre ellos, quizás el más sobresaliente es el factor de constricción presentado por Clerc [37]. El cual ayuda de manera significativa en la obtención de mejores tasas de convergencia a través de la inclusión de un coeficiente en la ecuación de la velocidad. Por ejemplo, considérese la siguiente expresión de la velocidad, donde χ sería el factor de constricción:

$$v_i^{(t+1)} = \chi \left(v_i^{(t)} + c_1 r_1 (pbest_i - x_i^{(t)}) + c_2 r_2 (gbest - x_i^{(t)}) \right) \quad (3)$$

Según el análisis simplificado de este modelo realizado por Clerc, el factor de constricción vienen dado por:

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (4)$$

Donde: $\varphi = c_1 + c_2, \varphi > 4, c_1 = c_2 = 2.05$.

Cuando PSO incluye este factor de constricción usualmente se le denomina PSO Canónico o Estándar [38-40]. El cual fue explicado en la sección anterior, de manera que en la expresión (1), $\eta_1 = \eta_2 \approx 1.4960$ y $\omega \approx 0.7298$.

2.3 PSO en ambientes dinámicos

La aplicación de PSO a los problemas dinámicos ha sido explorada por varios autores, [41],[42],[43],[44],[45],[46],[47],[48],[49] ,[50]. En la mayoría de estos trabajos se ha tomado como base al PSO clásico, al cual se le han hecho adaptaciones principalmente debido a dos dificultades en estos ambientes:

Memoria desactualizada (*outdated memory*)

Pérdida de diversidad (*diversity loss*)

A continuación se analizan estas dos dificultades.

2.3.1 Memoria desactualizada

La memoria desactualizada ocurre a causa de cambios en el ambiente (cuando la ubicación y/o valor del óptimo cambian). Tanto *gbest* (memoria global) como *pbest* (memoria local) dejan de ser válidas después de ocurrido un cambio. Esto puede traer consecuencias negativas para la convergencia de PSO, ya que el movimiento de cada partícula depende en gran medida de estas memorias, y si están desactualizadas, se convierten en falsos atractores.

El problema de la memoria desactualizada se resuelve típicamente asumiendo que el algoritmo sabe de antemano cuando ocurrirán los cambios en el ambiente, o simplemente es capaz de detectarlos. En cualquier caso, el algoritmo debe tratar de responder de forma apropiada. Un método de averiguar si ha ocurrido un cambio es una re-evaluación del *pbest* de una o más partículas [43],[46]. Una variante simple y efectiva sería la actualización de la memoria de todas las partículas tomando como base el valor y posición actual de una partícula, garantizando que el *gbest* sea igual al valor del máximo argumento de la función evaluada en el (*pbest*). Una posible desventaja es que la función no haya cambiado en la posición seleccionada, sino que el cambio haya ocurrido en otra parte. Según estudios esto puede solucionarse re-evaluando la función para todos los *pbest*, a expensas de incrementar al doble el número total de evaluaciones por corridas [39].

2.3.2 Pérdida de diversidad

De las dos dificultades que presenta PSO en ambientes dinámicos, *la pérdida de diversidad* es la más importante, la cual surge cuando un enjambre está

convergiendo hacia un óptimo, es decir, tiene una aproximación considerable y ocurre un cambio. Este efecto repercute en el tiempo que necesita el enjambre (que ha convergido parcialmente) para lograr la diversificación suficiente que permita la localización del nuevo óptimo.

Evidentemente, este fenómeno afecta considerablemente el rendimiento del algoritmo si el cambio ha desplazado al óptimo hacia una posición que está fuera del alcance de las partículas. Este alcance viene dado por la velocidad de cada partícula, recuérdese que estas, debido al factor de restricción, describen movimientos desacelerados (velocidades cada vez más pequeñas), entonces si el enjambre tiene a sus partículas muy cercanas al óptimo (poca diversidad) y ocurre un cambio, las bajas velocidades impedirán el movimiento hacia el nuevo óptimo.

En ese caso, el enjambre tiende a oscilar hacia un falso *atractor* a lo largo de una línea perpendicular al verdadero óptimo y se le conoce a este fenómeno como **colapso lineal** [51], efecto que es ilustrado en la Figura 2-2. En esta, se muestra a través de una secuencia de imágenes el comportamiento cuando los cambios del óptimo son más grandes que la diversidad del enjambre.

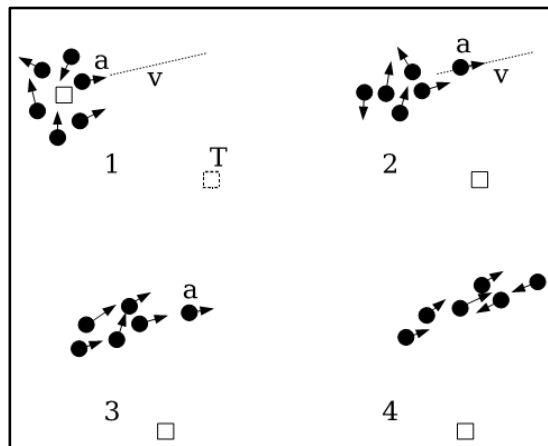


Figura 2-2. Secuencia del comportamiento del fenómeno colapso lineal [39].

Para contrarrestar esta dificultad, en la literatura se han propuesto varias soluciones interesantes. Por ejemplo, en [52] se utiliza una especie de diámetro para el enjambre: $|S|$, el cual representa su diversidad. En este trabajo, $|S|$ es definida como la distancia máxima entre dos partículas analizando cada dimensión x_{ij} . Con esta medida, es posible tomar decisiones en tiempo de ejecución como se verá más adelante.

Los enfoques existentes para compensar la pérdida de diversidad pueden agruparse en tres categorías, como se explica a continuación.

Diversidad después del cambio. En esta categoría se propone introducir diversidad después que el problema haya cambiado, ejemplo de ello es la Re-diversificación sugerida por Hu y Eberhart [46], que proponen varios métodos que involucran la re-inicialización parcial y completa del enjambre.

Una desventaja de este enfoque es que esta re-inicialización implica la pérdida de información recopilada hasta ese momento, además se hace difícil determinar qué cantidad de la población a re-inicializar es la más apropiada.

Diversidad durante la ejecución. Esta segunda categoría procura mantener la diversidad no importa si ha ocurrido o no un cambio. Para PSO, esto puede lograrse manteniendo durante todo el tiempo un mecanismo de tipo repulsivo o utilizando información de la vecindad. En ese sentido, Blackwell y Bentley [53],[54] introdujeron la variante CPSO, dónde algunas partículas que giran en torno al núcleo de partículas neutrales (partículas de PSO estándar) se repelen mutuamente.

La idea aquí es que los sub-enjambres cargados mantengan la diversidad de la población, por lo menos dentro de la extensión del espacio de las órbitas cargadas, para que el cambio en la función sea rápidamente detectado, y el enjambre puede adaptarse apropiadamente. Mientras, el núcleo puede continuar explorando la población en detalle hasta encontrar al óptimo. CPSO se ha

aplicado a varios problemas test unimodales y multimodales con valores de frecuencia de cambio alta y una severidad variante [41], mostrando un buen trabajo del auto-rendimiento del PSO convencional [47].

Otros autores han simplificado la idea y han reemplazado las partículas cargadas por partículas *quantums*, que básicamente se mueven aleatoriamente alrededor de la mejor posición global del enjambre[29]. La Figura 2-4 que muestra este esquema.

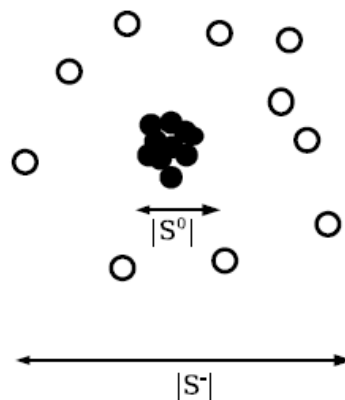


Figura 2-3. Analogía del átomo[39].

Enfoque PSO multi-enjambre. Los trabajos incluidos en esta categoría aprovechan la idea de contar con múltiples poblaciones (enjambres) con el objetivo de lograr una mejor exploración en problemas multimodales. Esta categoría es abordada con más detalle en la sección que continúa por ser un objetivo fundamental en este trabajo.

2.4 PSO Multi-Enjambre

Con el propósito de diversificar la búsqueda del óptimo global para evitar convergencias prematuras o quedar atrapados en óptimos locales, muchos investigadores han considerado el uso de las técnicas multi -poblaciones como una de las estrategias más exitosas para reforzar la diversidad de EAs frente a PDOs.

Esta idea de la multi-población es particularmente útil en los ambientes multi-modales con muchos óptimos. El objetivo aquí es permitir a cada población la convergencia hacia un pico prometedor. De manera que si un pico secundario se convierte en el óptimo global como resultado del cambio, una población esté cerca de él para detectarlo.

La aplicación de estas técnicas multi-enjambre desarrolla la habilidad de los algoritmos a adaptarse a un ambiente dinámico, aún cuando ocurran muchos cambios, permitiéndoles lograr la diversidad necesaria de su población para explorar eficientemente el espacio de búsqueda.

2.4.1 Trabajos relacionados

Por ejemplo, el algoritmo (SOS)¹ Self Organizing Scouts es un explorador auto-organizado presentado por Branke en [55], basado en una multi-población de EA que mostró excelentes resultados en aplicaciones a varios problemas test dinámicos como el MPB. Parrot y Li [56] crearon una variante de PSO (SPSO), que dinámicamente ajusta el número y el tamaño de los enjambres a través de una lista ordenada de partículas.

Recientemente, un algoritmo multi-enjambre bastante rápido (FMSO) fue propuesto por Changhe Li y Shengxiang Yang en [57]. FMSO utiliza la información intercambiada entre los enjambres para mantener la diversidad durante la ejecución. Un enjambre padre mantiene la exploración del espacio total de búsqueda usando un operador de búsqueda global basado en el esquema de un trabajo anterior: FEP⁶[58]. Simultáneamente varios enjambres hijos son creados dinámicamente alrededor de las mejores soluciones encontradas por el enjambre padre.

⁶ Siglas en inglés de Fast Evolutionary Programming.

Por su parte, Blackwell y Branke [59] establecen una versión de PSO multi-enjambre con el objetivo de mantener una multitud de enjambres en diferentes picos. Cada enjambre posee una estructura similar a la de un átomo: en el núcleo se encuentran partículas de PSO estándar, mientras que fuera de este se incluyen o partículas cargadas o de tipo quantum. Además un principio de exclusión permite que solamente un enjambre pueda rodear únicamente un pico.

2.4.2 Enfoque de Blackwell - Branke

Esta variante mostrada en pseudocódigo en el Anexo 1, está basada en trabajos anteriores realizados por los autores,[51],[60],[55].

Note que el algoritmo implementa como estrategia para la detección de cambios la evaluación de los *gbest* de cada enjambre. Además, lo que estos autores denominan *principio de exclusión* posibilita que cuando un enjambre se acerque a un pico que ya está siendo explorado por otro, el peor de ambos es reinicializado en el espacio de búsqueda. Por su parte, la *prueba de anti-convergencia* es realizada con el objetivo mantener la exploración en problemas donde existan más picos que enjambres, ya que con esta, se reinicia al peor enjambre de todos cuya diversidad sea inferior a un cierto umbral.

También es importante notar que mientras el núcleo (formado por partículas neutras que contienen a *gbest*) converge hacia el óptimo, alrededor de este se mueven otras de tipo quantum o cargadas. Particularmente, las quantum lo hacen a una distancia o radio prefijado, que depende de la severidad del cambio.

De las dos variantes propuestas por este enfoque: mCPSO y mQSO, basadas en partículas cargadas y quantum respectivamente, en este trabajo solo se utilizará la segunda, ya que la primera posee una complejidad computacional de orden $O(n^2)$ debido a las múltiples comparaciones que se realizan al simular la repulsión entre partículas.

2.5 Teorema “No Free Lunch”

Uno de los desarrollos teóricos más interesantes en el campo de la optimización lo constituye el denominado Teorema “No Free Lunch” (NFL) que fue propuesto por Wolpert and Macready[61],[62]. Este teorema establece que el rendimiento de todos los algoritmos de optimización (búsqueda) amortizado sobre el conjunto de todas las funciones objetivos posibles es equivalente.

Una de las implicaciones de este teorema es que ningún algoritmo puede ser diseñado de forma tal que sea mejor que una enumeración lineal del espacio de búsqueda o incluso superior a una búsqueda puramente aleatoria, es decir, a la larga todos los algoritmos tienen el mismo rendimiento como promedio.

Aunque parezca que no existen algoritmos mejores que otros, nótese que esta equivalencia solo es válida para *todas* las funciones objetivos, sin embargo, nada se ha dicho sobre qué sucede en un subconjunto de estas funciones, como es el caso de las funciones con descripciones compactas (una estructura determinada).

En este trabajo se asumirá que en los problemas test seleccionados, al ser un subconjunto de todas las funciones objetivos, es posible identificar diferencias significativas entre los algoritmos propuestos.

2.6 Conclusiones

Con el desarrollo de este capítulo se justifica el uso de la metaheurística PSO para encontrar soluciones óptimas en problemas de alta complejidad como los PDO. Se evidencia el éxito del enfoque PSO multi-enjambre para fortalecer la diversidad de la población, la cual se deteriora en la adaptación de PSO a problemas dinámicos. Además se relacionan varias propuestas, que basadas en este enfoque resuelven el problema de la diversidad (ej. mQSO). Estos trabajos son analizados a través de un estudio experimental en el siguiente capítulo.

CAPÍTULO 3.

Estudio experimental

En este capítulo se desarrolla el estudio experimental de las propuestas PSO multi-enjambre seleccionadas, de las cuales se describen brevemente sus parámetros y principales funcionalidades. Se especifican también las técnicas estadísticas y herramientas utilizadas tanto en el análisis de los resultados como en la ejecución de los algoritmos respectivamente. Por último se presenta el estudio experimental sustentado en los diferentes aspectos que permitieron la evaluación del rendimiento de los algoritmos, así como la configuración del escenario del problema tipo seleccionado para probar estas variantes.

3.1 Algoritmos implementados

El proceso de experimentación que corresponde a esta investigación exigió la implementación de las propuestas PSO-multi-enjambre seleccionadas a partir de la información encontrada en la bibliografía existente. Sin embargo, no fue posible la implementación de todas las variantes encontradas por no contarse con la información necesaria y suficiente para su correcta implementación. Esto

dificulta la obtención de resultados comparables a los que muestra la bibliografía donde fueron propuestos estos algoritmos.

Los algoritmos escogidos se muestran jerárquicamente en la figura 3.1 para una mejor comprensión de la evolución y surgimiento de las variantes multi-enjambre de PSO.

Como primer eslabón de esta jerarquía de clases se implementó el algoritmo *mPSO*. Este algoritmo no existe de manera explícita en alguna bibliografía, pero fue implementado con el objetivo de realizar una jerarquía apropiada entre las variantes. *mPSO* incluye la mayoría de los parámetros y funcionalidades que caracteriza al enfoque Blackwell-Branke, explicado en el capítulo anterior. Esta propuesta tiene asociados los siguientes parámetros, que heredarán las demás clases:

- Número de partículas: cantidad de partículas asociadas a un enjambre.
- Número de enjambres: cantidad de enjambres que tiene la población.
- Radio de exclusión (r_{excl}): distancia entre los enjambres que optan por un mismo óptimo.
- Radio de convergencia (r_{conv}): límite inferior de la distancia máxima entre las partículas de un enjambre.

A modo de resumen las funcionalidades que se observan en la figura 3.1, una vez inicializado el algoritmo iteran a través de un ciclo principal de 5 fases:

- Prueba para la convergencia
- Prueba para la exclusión
- Prueba para el cambio de la función
- Actualización de la partícula
- Actualización del atractor

Además de las funcionalidades heredadas de *mPSO*, el algoritmo *mQSO* [59] incluye el uso de partículas quantum e introduce un nuevo parámetro r_{cloud} . Es por eso que se implementó la clase *QParticle*, la cual hereda de la clase *Particle* como se muestra en el Anexo 5. Cada *QParticle* actualiza su posición según la siguiente expresión:

$$\vec{x}_{ni} \in B_n(r_{cloud}) \quad (5)$$

Es decir, toma una posición aleatoria dentro de la hiperesfera B_n que tiene un radio r_{cloud} .

Por otro lado, las variantes *mPSO-CPT* y *mQSO-CPT* constituyen adaptaciones de *mPSO* y *mQSO* respectivamente. Ambas incluyen un operador denominado CPT⁷, el cual se encarga de controlar las trayectorias de las partículas con un deterioro en la búsqueda de nuevas soluciones [63]. El pseudocódigo de este algoritmo se puede ver en el Anexo 2.

⁷ Control Particle Trajectories

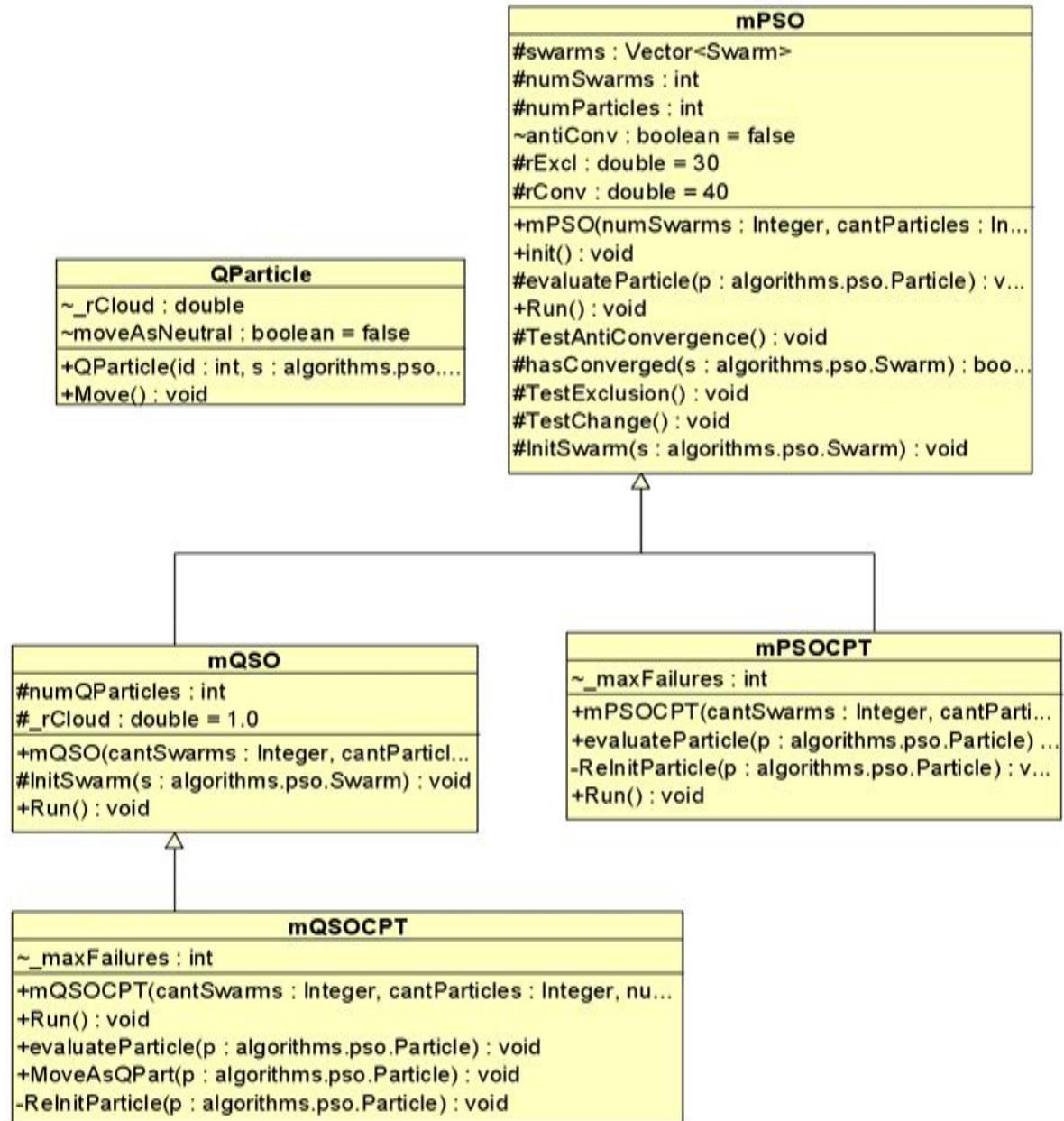


Figura 3-1 Diagrama de clases

La dinámica de los modelos multi-enjambres presentados es regida por tres parámetros r_{excl} , r_{conv} y Q ó r_{cloud} , los cuales se derivan de los principios

mencionados en la sección 2.3, exclusión, anticonvergencia y la inclusión de partículas quantum para preservar la diversidad del algoritmo.

3.2 Técnicas estadísticas y herramientas utilizadas.

Para clasificar, calcular, y analizar los datos obtenidos en los experimentos de manera sistemática, se utilizaron un conjunto de procedimientos conocidos como pruebas estadísticas. Particularmente se utilizaron pruebas paramétricas, las cuales asumen una distribución normal en los datos, característica presente en las simulaciones llevadas a cabo en los experimentos.

3.2.1 Herramientas empleadas para la implementación y análisis

Las implementaciones de los algoritmos multi-enjambre se realizó utilizando la tecnología Java, de la compañía Sun Microsystem [64], por ser un lenguaje de desarrollo de propósito general válido para realizar todo tipo de aplicaciones profesionales y una de sus principales características es que es utilizado para el desarrollo de una gran cantidad de proyectos, muchos de ellos *open source*, lo cual lo convierte en un lenguaje muy amplio y con una cantidad de librerías que permite realizar las operaciones más disímiles. Para usar esta tecnología se empleo el IDE NetBeans en su versión 6.0[65].

Para asistir la implementación y obtención de resultados de los algoritmos frente a los PDO seleccionados, se utilizó la aplicación DynOptLab[66]. Este software brinda la posibilidad de automatizar los experimentos a través de la extensión de un framework y una interfaz visual bastante sencilla, ver Anexo 3.

A los resultados obtenidos por DynOptLab se les realizaron las pruebas estadísticas correspondientes con la ayuda del paquete SPSS en su versión 15.0 (Anexo 4).

3.3 Aspectos generales de los experimentos

Cada algoritmo mencionado en la sección anterior se probó para diferentes instancias del movimiento de picos (MPB), debido a su particularidad de ser un problema altamente parametrizable, el cual no sólo permite configurar sus parámetros, sino que el poder experimentar con un subconjunto de problemas (es decir, diferentes instancias del mismo) permite simular problemas del mundo real y encontrar diferencias significativas entre el rendimiento de los algoritmos.

Los parámetros estándares establecidos en el problema se resumen en la Tabla 3-1, los cuales pertenecen al escenario 2 para facilitar la comparación con los resultados obtenidos en otros trabajos.

Parámetros	Configuración
Picos (p)	10
Dimensiones (d)	5
Altura de los picos (H_i)	$\in [30.0, 70.0]$
Ancho de los picos (W_i)	$\in [1, 12]$
Evaluaciones entre cambios (Δe)	5000
Severidad de los cambios (s)	1.0
Coefficiente de correlación (λ)	0.0

Tabla 3-1. Parámetros escenario 2 Movimiento de Picos (MPB).

Los experimentos se organizaron por cada problema tipo, con énfasis en los siguientes aspectos:

- Efecto de variar la configuración multi-enjambre.
- Efecto de variar la severidad de cambio (s).
- Efecto de variar el número de picos (p).
- Efecto de variar la frecuencia de cambio (Δe).

Por cada algoritmo se realizaron 50 corridas con semillas aleatorias diferentes tanto para éste como para el problema tipo correspondiente. Además se seleccionó como medida para evaluar la eficiencia de cada algoritmo el error fuera de línea (error offline), explicado en la sección 1.2.

En cuanto a la condición de terminación para cada experimento, se asume que el número de cambio es igual a 100, (lo que indica que el número de evaluaciones es igual a la frecuencia de cambio por el número de cambio). Por ejemplo, si $\Delta e = 5000$ entonces habrían 500 000 evaluaciones de la función objetivo.

La metodología seguida en los experimentos desarrollados es la que se muestra en la figura siguiente.

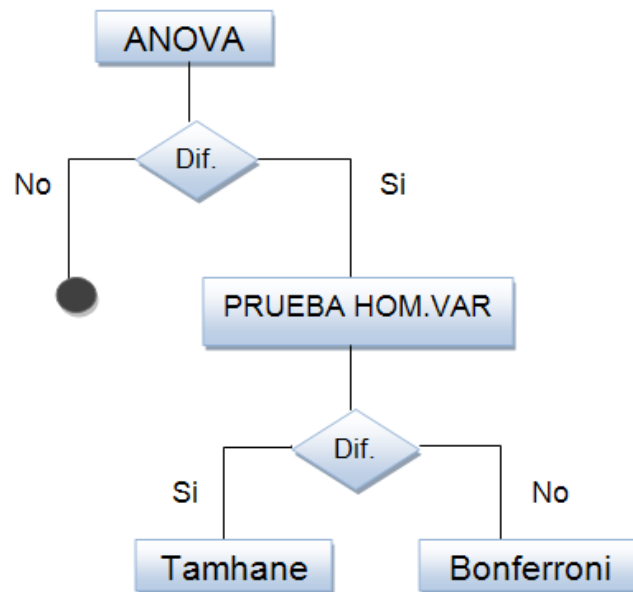


Figura 3-2 Metodología para el análisis estadístico

Como se muestra en el esquema anterior, primeramente se realizará un análisis de varianza (ANOVA) simple para detectar diferencias en el grupo de manera general. En caso de que existan estas diferencias, se aplicará una prueba de

homogeneidad de varianza que permita seleccionar la prueba *post hoc* correspondiente: Bonferroni para varianzas iguales y Tamhane para varianzas diferentes.

Para establecer la comparación se consideraron cuatro niveles de significación:

1. significativo, un resultado de significación menor que 0.05 y mayor que 0.01
2. altamente significativa, una significación menor que 0.01
3. medianamente significativo, un resultado menor que 0.1 y mayor que 0.05
4. no significativo, un resultado mayor que 0.1

3.4 Experimentos

A continuación se describirán los experimentos desarrollados, con énfasis en los resultados obtenidos.

3.4.1 Efecto de variar la configuración multi-enjambre.

El primer experimento analiza el efecto que tiene variar la configuración poblacional en el rendimiento de los algoritmos. De esta manera, quedará establecida la configuración más idónea que servirá de base en el resto de los experimentos.

Téngase en cuenta que estas configuraciones tienen la forma $M(N + N^q)$, donde M es la cantidad de enjambres, N cantidad de partículas neutrales (PSO clásico), y N^q cantidad de partículas quantum. Para los algoritmos que no tengan enjambres atómicos esta configuración significará M enjambres con $N + N^q$ partículas neutrales.

El tamaño total de la población N_{pop} (número total de partículas) fue fijado para 100 partículas, de las cuales se pueden probar diferentes configuraciones. Así

mismo el número de enjambres M está en el rango de $[1,100]$, sin embargo nótese como al usar un enjambre se pierde el concepto de multipoblación, debido a que una sola partícula no puede intercambiar información a través de la actualización del (g_{best}), por lo que las configuraciones a probar varían de 2 – 50 enjambres. Particularmente para la configuración 14(4 + 3) se utilizaron solamente 98 partículas.

La Tabla 3-2 muestra la media del error fuera de línea y la desviación estándar de los algoritmos propuestos, para las 11 configuraciones probadas.

Población	mPSO	mQSO	mPSO-CPT	mQSO-CPT
2(25+25)	7,32(1,66)	7,19(2,04)	7,02(1,59)	6,33(2,06)
3(17+16)	5,89(1,81)	5,21(1,24)	5,41(1,30)	4,27(1,40)
4(13+12)	4,77(1,29)	4,15(1,09)	4,27(0,88)	3,25(1,05)
5(10+10)	4,47(1,50)	3,36(0,97)	3,74(0,91)	2,60(0,69)
10(10+0)	4,41(1,50)	4,10(1,50)	2,98(0,63)	2,82(0,63)
10(0+10)	4,41(1,50)	2,69(0,37)	2,98(0,63)	2,68(0,36)
10(5+5)	4,41(1,50)	2,35(0,97)	2,98(0,91)	1,84(0,69)
14(4+3)	5,02(1,81)	2,57(0,48)	3,30(0,68)	2,27(0,42)
20(3+2)	6,73(1,76)	3,04(0,47)	4,30(0,79)	2,74(0,39)
25(2+2)	9,50(2,32)	3,36(0,50)	5,82(0,98)	3,15 (0,46)
50(1+1)	16,88(3,60)	4,91(0,73)	15,27(2,69)	4,75 (0,64)

Tabla 3-2. 11 Configuraciones posibles de la población.

A continuación se describe el comportamiento para algunas configuraciones que no reflejaron los mejores resultados, pero son necesarias para establecer las comparaciones. Para ello se tomó como referencia los valores mostrados anteriormente en la tabla 3-3.

La aplicación de un análisis de varianza (ANOVA) a los valores obtenidos en la ejecución de los algoritmos para la configuración $M = 50(1 + 1)$ arrojó que existen diferencias altamente significativas ($p = 0.00$) entre las medias del grupo, por lo que se utilizó una prueba de homogeneidad de varianzas que mostró diferencias altamente significativas ($p < 0.01$). Se consideró entonces

aplicar un test de Tamhane para varianzas diferentes que indicó que no existen diferencias entre *mPSO* y *mPSO-CPT*, así como entre *mQSO* y *mQSO-CPT*. No obstante si existe una diferencia grande entre estos pares. Además se aprecia un valor del error bajo (4.88) para las variantes que usan quantum mientras que las que utilizan solamente partículas neutrales presentan el error fuera de línea muy alto (16.07).

Especialmente la configuración 50(1 + 1), está constituida de enjambres muy pequeños de tan solo 2 partículas cada uno, lo que provoca un rendimiento bastante malo debido a la falta de diversidad dentro del enjambre. Resulta muy interesante observar en la figura 3-2 como las variantes *mQSO* y *mQSO-CPT* pueden alcanzar un valor más bajo que el obtenido en $M = 2(25 + 25)$ y $M = 3(17 + 16)$.

Para $M = 2(25 + 25)$ ANOVA mostró que existen diferencias significativas en el grupo ($p = 0.039$), por lo que se aplicó la prueba de Homogeneidad de varianzas que arrojó una diferencia medianamente significativa ($p = 0.083$) y entonces se empleó Bonferroni. Este test sólo mostró discrepancias entre *mPSO* con un error superior al de *mQSO-CPT*.

A diferencia de la configuración anterior ANOVA demostró que existen diferencias altamente significativas ($p = 0.00$) en el grupo para $M = 3(17 + 16)$. Entonces la prueba de homogeneidad de varianzas aplicada no mostró diferencias ($p = 0.29$). Por lo que se usó Bonferroni que sólo presentó diferencias entre *mQSO-CPT* con respecto a los demás algoritmos por tener el error más bajo (Anexo 6). De manera similar se analizó la configuración para $M = 4(13 + 12)$ hasta justificar el empleo de Bonferroni. Este análisis arrojó que las diferencias en esta configuración se aprecian en *mQSO-CPT* con respecto a las demás variantes al presentar el error más pequeño.

Semejante a los análisis anteriores también existen diferencias en el grupo con la configuración $M = 5(10 + 10)$. En este caso se empleó un test de Tamhane debido a la existencia de diferencias altamente significativas en la prueba de homogeneidad de varianzas ($p = 0.00$), el cual señaló que *mPSO* con un error elevado difiere de las demás propuestas, de la misma manera que lo hace *mQSO-CPT* pero con el error más bajo.

Entre las configuraciones con alrededor de 20 enjambres se escogió $M = 25(2 + 2)$, que con un desempeño parecido a la de $M = 50(1 + 1)$ presenta diferencias altamente significativas en el grupo. Con esta población nuevamente *mQSO-CPT* difiere por tener un error mínimo con respecto a los demás algoritmos, mientras *mPSO* conserva el error más grande.

Por otro lado, en la Figura 3-3 puede apreciarse que las configuraciones con mejores resultados son $M = 10(5 + 5)$, $M = 10(0 + 10)$, $M = 10(10 + 0)$ y $M = 14(4 + 3)$ por mostrar en todos los casos el error más bajo.

El análisis ANOVA y la prueba de homogeneidad de varianzas a la combinación $M = 10(0 + 10)$, constituida solamente por partículas quantum, infirió que existen diferencias altamente significativas ($p = 0.00$) en el grupo y las varianzas respectivamente. Esto permitió utilizar Tamhane y encontrar diferencias significativas entre *mPSO* con un error superior a las demás variantes, mientras que las que incluyen el operador CPT y quantum tienen un error más bajo.

La configuración anterior tiene mejores resultados que $M = 10(10 + 0)$, compuestas solamente por partículas neutrales. Sin embargo el rendimiento es superior si se utiliza una combinación de $M = 10(5 + 5)$. Esto se debe a que las partículas quantum ayudan a seguir los cambios, pero solamente las neutrales pueden converger y desarrollar rápidamente una buena solución, es decir el equilibrio entre partículas neutrales y quantum preserva la diversidad y garantiza la convergencia.

Por eso se establece la distribución de 10 enjambres $M = 10(5 + 5)$ como la óptima para usarse en los experimentos que se desarrollan a continuación.

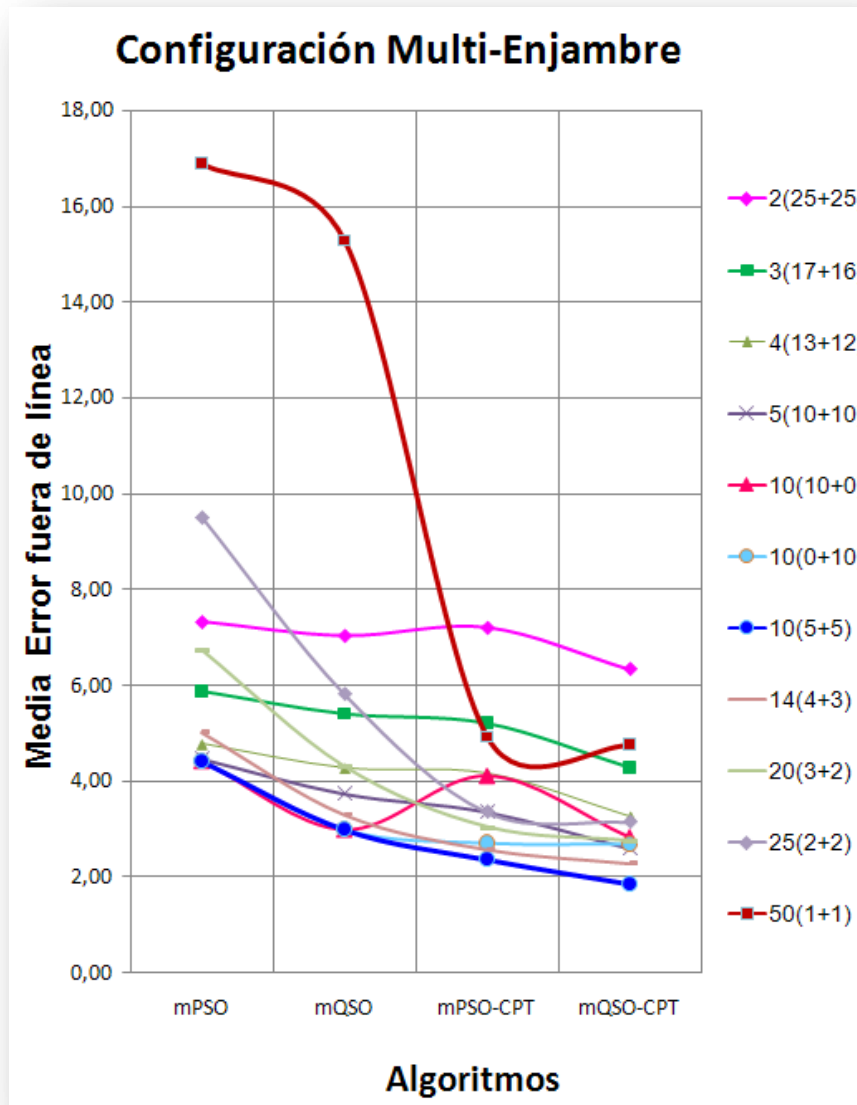


Figura 3-3. Variación del error fuera de línea para diferentes configuraciones de la población.

3.4.2 Efecto de variar la severidad de cambio (s).

El objetivo de este estudio es observar cómo varía el error fuera de línea de estas propuestas multi-enjambres a medida que aumenta el valor de la severidad del cambio (s) del PDO. Para ello se estableció $r_{cloud} = s$, como valores de s {0.0, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0} y se escogió como configuración de la población $M = 10(5 + 5)$, por lo explicado en el experimento anterior. Las medias obtenidas se pueden ver en la Tabla 3-2, las cuales son la base del siguiente análisis.

S	mPSO 10(5+5)	mQSO 10(5+5)	mPSO-CPT 10(5+5)	mQSO-CPT 10(5+5)
0.0	2.82(1.41)	1.05(0.43)	0.85(0.55)	0.43(0.31)
0.5	3.61(1.48)	2.02(0.44)	1.98(0.51)	1.62(0.36)
1.0	4.41(1.50)	2.35(0.44)	2.98(0.63)	1.84(0.40)
2.0	5.96(1.72)	2.85(0.48)	4.74(0.88)	2.38(0.44)
3.0	7.54(1.93)	3.40(0.60)	6.32(1.15)	2.87(0.48)
4.0	9.16(2.19)	3.89(0.72)	7.81(1.34)	3.39(0.51)
5.0	10.67(2.38)	4.34(0.68)	9.07(1.66)	3.96(0.68)

Tabla 3-3. Error fuera de línea para diferentes valores de S (severidad del cambio).

Para el análisis de las muestras se procedió de la misma manera que el primer experimento. En este caso el análisis Anova y la prueba de homogeneidad de varianzas arrojaron diferencias altamente significativas ($p = 0.00$) para todos los valores de s . Por lo que se utilizó un test de Tamhane en las 7 combinaciones establecidas.

Para valores como $s = 0$ el error de los algoritmos alcanza los valores más bajos. Particularmente en este caso el error más pequeño lo tienen los algoritmos que incluyen el operador CPT, principalmente la variante mQSO-CPT. Sin embargo el valor de mQSO para $s = 0$ difiere muy poco de las propuestas que incluyen CPT, mientras mPSO se mantiene al margen con un error superior como se observa la figura 3-3.

A medida que se incrementa el valor de s , el error aumenta para todos los algoritmos propuestos. Pero lo más interesante resulta que hasta $s < 1$ no se hallaban diferencias notables entre $mQSO$, $mPSO-CPT$ y $mQSO-CPT$. Sin embargo para $s \geq 1$, $mPSO-CPT$ incrementa su error y establece diferencias notables con $mQSO$ y $mQSO-CPT$. El error en estas variantes que incluyen partículas quantum se mantiene por debajo de las demás y muy similar entre ellas. Aunque $mPSO-CPT$ aumentó mucho el valor de su error y se acerca bastante a $mPSO$, este último se mantiene con el error más alto.

A modo de resumen podemos decir que para las variantes $mQSO$ y $mQSO-CPT$ el error se mantuvo estable y con buenos resultados, mientras que $mPSO-CPT$ sólo obtuvo efectos deseables para valores de severidad muy baja.

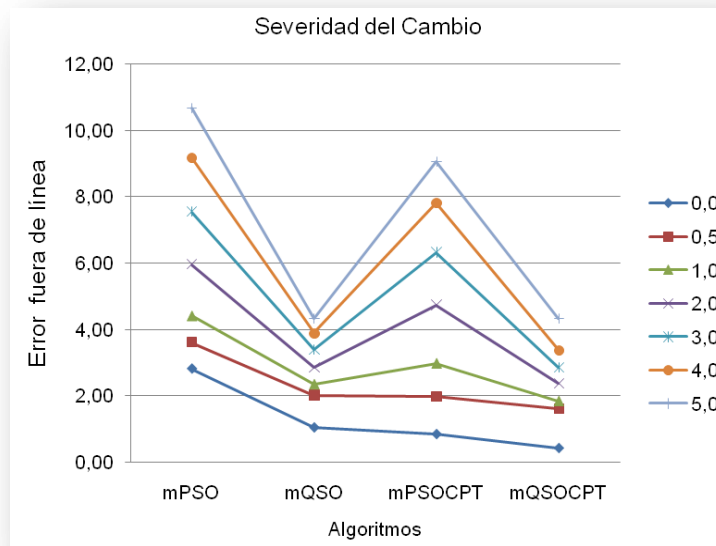


Figura 3-4. Comportamiento de la severidad del cambio

3.4.3 Efecto de variar el número de óptimos (p).

Variar el número de óptimos es un experimento que permite investigar cuanto escala con el número de picos las propuestas analizadas en este trabajo.

También se toma en cuenta si la idea de la anti-convergencia es relevante solamente cuando existen más picos que enjambres.

El número de óptimos es variado entre 1 y 200 para la configuración $M = 10(5 + 5)$ y las pruebas se ejecutan activando el operador de la anti-convergencia para $p > 10$.

Óptimos	<i>mPSO</i>	<i>mQSO</i>	<i>mPSO-CPT</i>	<i>mQSO-CPT</i>
1	5.92(4.83)	3.39(0.92)	4.57(1.27)	2.87(0.79)
2	6.69(6.71)	2.82(0.78)	3.96(1.21)	2.37(0.65)
5	4.46(1.97)	2.38(0.54)	3.32(0.83)	2.00(0.37)
7	4.18(1.41)	2.25(0.51)	3.07(0.77)	1.81(0.37)
10	4.41(1.50)	2.35(0.44)	2.98(0.63)	1.84(0.40)
20	4.03(0.84)	2.89(0.45)	3.39(0.54)	2.42(0.38)
30	4.03(0.76)	3.14(0.60)	3.51(0.52)	2.56(4.44)
40	3.86(0.66)	3.21(0.39)	3.54(0.53)	2.66(0.35)
50	3.76(0.49)	3.33(0.46)	3.46(0.46)	2.60(0.33)
70	3.55(0.35)	2.49(0.16)	6.88(0.33)	2.58(0.17)
85	3.53(0.30)	2.63(0.16)	7.18(0.46)	2.59(0.15)
90	3.40(0.36)	2.35(0.15)	7.03(0.42)	2.37(0.12)
100	3.60(0.36)	2.63(0.20)	7.48(0.39)	2.68(0.15)
200	3.43(0.30)	2.54(0.15)	6.10(0.30)	2.43(0.12)

Tabla 3-4 Error fuera de línea para diferentes valores de p (cantidad de óptimos).

Análogo a los experimentos ya realizados, en este caso ANOVA también mostró diferencias altamente significativas ($p = 0.00$) entre todos los algoritmos para las combinaciones mencionadas. Por lo que se aplicó una prueba de homogeneidad de varianzas que indicó la misma significación y por ello se empleó el test de Tamhane.

Para problemas de 1 a 5 óptimos el error fuera de línea alcanza valores elevados para *mPSO*, mientras que *mPSO-CPT*, *mQSO* y *mQSO-CPT* presentan un error más bajo según el orden que son mencionados.

Semejante al comportamiento anterior si el número de óptimos queda establecido entre 7 y 10 *mPSO* conserva el error por encima de las demás

variantes. Estas combinaciones se pueden considerar las más óptimas ya que mPSO-CPT, mQSO y mQSO-CPT presentan el error más bajo de todas las combinaciones probadas, sobre todo mQSO-CPT con (1.81) y (1.84) en cada combinación respectivamente.

Por otro lado si se varía el número de óptimos en un rango [20 – 50] el desempeño de cada algoritmo es muy similar por combinación, aunque entre los exponentes si existen diferencias. Obsérvese en la figura 3-5 como mQSO-CPT con un error bastante aceptable es inferior a mQSO y mPSO-CPT, los cuales no presentan mucha diferencia entre ellos, mientras mPSO continúa con un error elevado. Téngase en cuenta que para estas combinaciones y las que siguen se establece el principio de la anticonvergencia.

Si las combinaciones establecidas son $p = \{70,85,90,100\text{y}200\}$, resulta interesante observar en la figura 3-5, aunque sólo aparezcan las más representativas, como ocurren cambios en todos los exponentes. Nótese como en este caso mPSO-CPT aumenta considerablemente el error fuera de línea para obtener el valor más alto durante el experimento (7.48). Sin embargo mPSO y mQSO disminuyen el valor del error, éste último incluso con un error menor al de mQSO-CPT para $p = \{70,90 \text{ y } 100\}$, el cual incrementó el valor de su error después de haber permanecido como óptimo en las configuraciones anteriores.

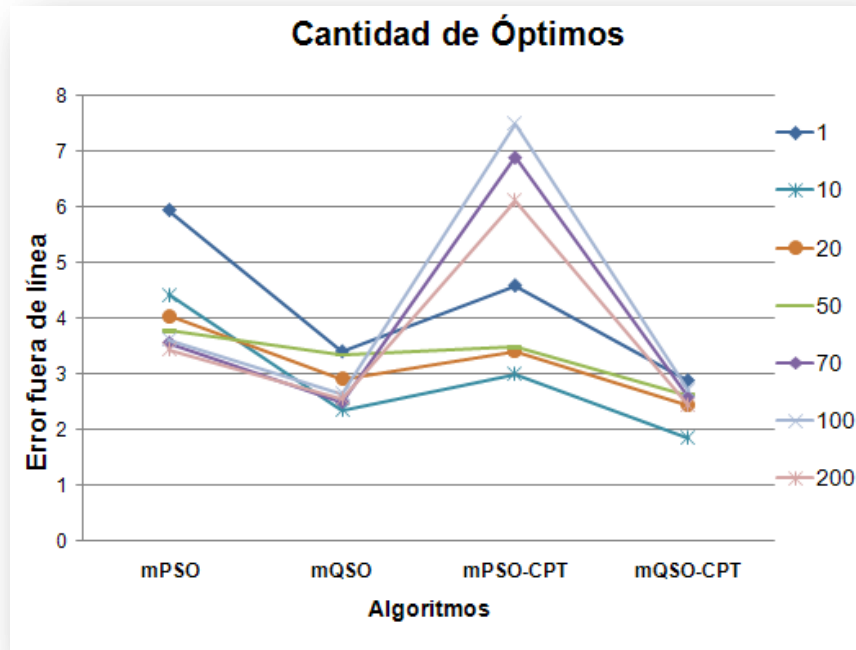


Figura 3-5 Variación de la cantidad de óptimos

3.4.4 Efecto de variar la frecuencia de cambio (Δe).

Este experimento permite analizar a través de las muestras obtenidas (error fuera de línea), la velocidad de respuesta de los algoritmos a medida que aumenta la frecuencia de cambio Δe . Los valores de Δe establecidos para el experimento son: {500, 1000, 2000, 5000, 7000, 10000}, lo cual indica que los óptimos del problema cambiarán cuando hayan pasado Δe iteraciones.

Para cada combinación de Δe se realizaron 50 ejecuciones y en cada una de ellas se consideraron 100 cambios, por tanto el valor de iteraciones que determina el criterio de parada es $\Delta e \times 100$. El promedio de los datos para cada configuración de Δe se muestra en la Tabla 3-2.

Evaluaciones(Δe)	mPSO	mPSO-CPT	mQSO	mQSO-CPT
500	20,05(5,79)	10,82(1,82)	8,58(1,04)	7,06(0,97)
1000	11,28(2,75)	5,41(0,67)	6,39(1,12)	4,46(0,62)

Evaluaciones(Δe)	mPSO	mPSO-CPT	mQSO	mQSO-CPT
2000	6,95(2,19)	3,71(0,58)	4,35(0,83)	3,1(0,51)
5000	4,41(1.50)	2,35(0.63)	2,98(0.44)	1,84(0.40)
7000	3,92(1.40)	2,03(0.58)	2,67(0.47)	1,46(0.33)
10000	7,32(1.66)	1,59(0.55)	2,41(0.38)	1,14(0.28)

Tabla 3-5. Variación de la frecuencia de cambio. (Número de evaluaciones).

Análogo a los experimentos anteriores se efectuó un análisis ANOVA con diferencias altamente significativas ($p = 0.00$) en el grupo (Anexo 7). Entonces se aplicó la prueba de homogeneidad de varianzas que indicó el uso de Tamhane. Este análisis se hizo extensivo a todas las combinaciones.

La aplicación de Tamhane mostró las diferencias más considerables para $\Delta e = 500$. Nótese en la Figura 3-6 como el valor del error fuera de línea es diferente entre todos los algoritmos y mucho más alto con respecto a las demás configuraciones. Si se ordenaran las variantes por un descenso en su rendimiento, quedaría de la siguiente manera: primero *mPSO* con el error más elevado de todas las combinaciones (20.05), le seguiría *mPSO-CPT* (10.82) y por último *mQSO* y *mQSO-CPT* con un margen de diferencia menor.

Para un período de cambio entre 1000 y 2000, los algoritmos tienen comportamientos similares. En ambos casos *mPSO* y *mQSO* presentan el error más alto, sobre todo *mPSO* para $\Delta e = 1000$. Sin embargo las variantes que incluyen el operador CPT comienzan a disminuir su error con muy poca diferencia entre ellas.

Aunque Tamhane indicó diferencias entre todas las variantes, resulta interesante como para valores mayores de Δe , *mQSO* y *mQSO-CPT* presentan un error muy semejante, ejemplo de ello es para $\Delta e \geq 5000$. Nótese como para $\Delta e = 5000$ y $\Delta e = 7000$ *mPSO* y *mPSO-CPT* alcanzan un error más bajo, pese a que este valor se mantenga inferior con respecto las variantes que incluyen partículas quantum. Para $\Delta e = 10000$, *mQSO*, *mPSO-CPT* y *mQSO-CPT* revelan el error

más bajo del experimento, sin embargo *mPSO* presenta un error superior al que obtuvo para $\Delta e = 2000$.

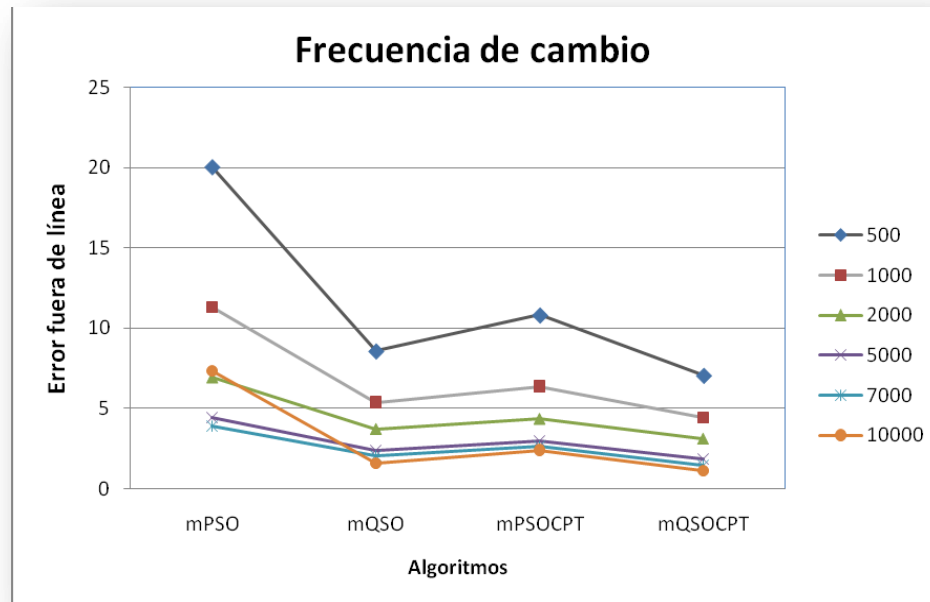


Figura 3-6. Variación de la frecuencia de cambio

3.5 Conclusiones

Al finalizar este capítulo, es posible apreciar que las técnicas estadísticas y herramientas utilizadas fueron bastante eficaces en la realización de los experimentos planificados. Por su parte, los experimentos seleccionados permitieron estudiar los aspectos más importantes para un algoritmo de optimización dinámica, a saber, configuración poblacional, adaptación frente a diferentes severidades, frecuencias y cantidad de óptimos. A su vez, los resultados obtenidos en este capítulo servirán de fundamentos para la obtención de una guía de aplicación, como se detallará a continuación.

CAPÍTULO 4.

Guía de aplicación

En este capítulo se expone una guía de aplicación basada en los experimentos realizados anteriormente, situando al algoritmo más apropiado para usar, según la necesidad del problema y restricciones del mismo. Estas propuestas están desglosadas según los aspectos tratados en cada experimento. Además se realiza la valoración de sostenibilidad que corresponde a esta investigación.

4.1 Guía de aplicación

Atendiendo a los resultados obtenidos acerca del desempeño de los algoritmos y las variadas problemáticas que pudieran presentarse, así como las restricciones que dieran origen a un determinado problema se propone esta guía de aplicación.

4.1.1 Severidad

Los resultados alcanzados muestran que para ambientes estáticos $s = 0$, los algoritmos tienen un buen rendimiento, lo que indica que las técnicas preservadoras de la diversidad no son desfavorables en contextos estáticos. Sin

embargo, cuando s se incrementa, la habilidad de los algoritmos para encontrar al nuevo óptimo después del cambio se degrada. Esto se debe a que la distancia entre el óptimo *viejo* y el *nuevo* aumenta. Particularmente, este requerimiento constituye una problemática a enfrentar en investigaciones posteriores, ya que hasta ahora, los algoritmos prefijan parámetros en dependencia de las características del problema, lo cual atenta contra su robustez. En este trabajo, el parámetro r_{cloud} es un ejemplo de esta situación (los mejores resultados se obtienen cuando $r_{cloud} = s$).

Basado en los experimentos anteriores, es posible observar entonces que tanto *mQSO* como *mQSO-CPT* presentan los mejores resultados. Esto se debe a que ambos incluyen partículas quantum como estrategia para seguir a los óptimos.

En situaciones donde la diferencia que existe entre ambos algoritmos no sea un problema y se busque rapidez en el proceso de optimización, se propone el uso de *mQSO*, ya que posee una complejidad computacional menor.

4.1.2 Cantidad de óptimos

Basado en los experimentos ya realizados se puede apreciar que el número de enjambres debe igualar el número de picos ($M = P$), ya que si se incrementa el número de enjambres más allá de diez se reduce el rendimiento. Esto se debe a que los enjambres adicionales inevitablemente escalan alrededor de los picos ocupados y solamente pueden ser reiniciados por exclusión.

Los valores obtenidos en la experimentación muestran un buen desempeño de los algoritmos que utilizan quantum para ambientes donde sólo hay modificación de 1 óptimo.

Sin embargo cuando el número de óptimos se incrementa, dos aspectos del problema cambian: el problema llega a ser más difícil, debido a que cada vez se dificulta seguir el movimiento de tantos óptimos. Por otro lado para las

soluciones estándar el error se reduce, debido a que la función del MPB alcanza el valor máximo para todos los óptimos.

Parece ser que el enfoque multi-enjambre realiza el mejor desempeño cuando el número de óptimo es igual o ligeramente más pequeño que el número de enjambres, con y sin la anti-convergencia. Esto coincide con la percepción de asociar un enjambre con cada óptimo

Cuando la modalidad del ambiente aumenta, la anti-convergencia ofrece una ventaja en aumento significativa. De ahí que mQSO exhiba un mejor rendimiento cuando se incrementa el número de óptimos.

A modo de resumen podemos decir entonces que tanto mQSO y mQSO-CPT ofrecen buenos resultados para situaciones donde haya un incremento del número de óptimos.

4.1.3 Frecuencia de cambio

Por los resultados alcanzados en los experimentos anteriores se puede afirmar que a medida que Δe aumenta, las posibilidades de encontrar el óptimo son mayores. Esto se debe a que existen más evaluaciones disponibles, sin embargo para valores pequeños Δe sucede lo contrario. Esto demuestra la superioridad de la variante *mPSO-CPT*, ya que la misma aprovecha al máximo las evaluaciones disponibles cuando se obtiene una ventana de tiempo mayor donde el problema se comporta estacionario.

El desempeño de los algoritmos es mejor para los valores de Δe más altos, principalmente las variantes *mQSO* y *mQSO-CPT*. Sin embargo se puede apreciar un comportamiento similar ara $\Delta e = 5000$, en ambos casos.

4.2 Valoración de sostenibilidad.

En esta nueva era de la información la sociedad está estrechamente relacionada con el mundo de la informática, debido a su contribución para con la misma en la extensión de la cultura, la educación de las personas y el incremento del nivel de vida, además de proyectarse hacia la asimilación de la información con una forma muy amena.

Son estos los principios fundamentales que conlleven a que las investigaciones que se desarrollen en torno a la informática tengan su principal repercusión en lo administrativo, socio-humanista, ambiental y tecnológico.

Sin estar exento de ello, el trabajo aquí presentado provoca un impacto que se hace notar en cada una de estas dimensiones. Distintamente a otros trabajos realizados, se hace constar que el desarrollo de esta investigación no genera ingresos directos, en la panorámica **administrativa**. El aporte económico se desprende del evidente fin social, industrial y económico de la Optimización con Enjambre de Partículas en ambientes dinámicos. Optimizar es encontrar la(s) mejor(es) alternativa(s) entre varias que brinde resultados óptimos. En ese sentido, todo proceso que se optimice conlleva a un ahorro de recursos y resultados superiores, lo cual deviene en logros para los proyectos que apliquen técnicas de esta naturaleza. En consecuencia se puede afirmar que repercute positivamente en la economía de las industrias y demás propósitos sociales que recurran al uso de la misma.

La guía que se obtiene como resultado de esta investigación, es de gran utilidad para el trabajo a desarrollar por futuras aplicaciones que necesiten de algunos de estos algoritmos en problemas más actuales como por ejemplo: el ajuste automático de parámetros en algún proceso industrial, la mejor planificación de un horario en un determinado período, o en la tecnología para predecir el comportamiento de determinados factores. Igualmente contribuye a facilitar el

trabajo experimental en el uso de los algoritmos multi-enjambres para la resolución de los PDO para los cuales no fueron creados originalmente, lo que significa un ahorro de tiempo de investigación y la cantidad de horas-máquina en el desarrollo de una aplicación, con lo cual se reducen los egresos por razones de electricidad y apoya el proceso de selección de los algoritmos a utilizar.

Durante el trabajo de investigación realizado no se incurrió en gastos significativos: cuenta únicamente el consumo eléctrico de la computadora utilizada y la iluminación requerida para ello. Puesto que la herramienta de programación empleada, NetBeans 6.5, se rige por los principios del software libre.

En el aspecto **socio-humanista** el primer punto de referencia lo constituye su contribución a que estudiantes y profesores amplíen sus conocimientos referentes a cómo se comportan los principales exponentes de las técnicas PSO multi-enjambre frente a diferentes PDO. De esta forma se favorece el proceso docente de enseñanza-aprendizaje de la disciplina Matemática Aplicada e Inteligencia Artificial.

La investigación se puede considerar una referencia palpable para la experimentación y mejor selección de técnicas basadas en PSO multi-enjambre en ambientes dinámicos. Por lo que esta observación alcanzaría beneficios en la optimización de respuestas y como consecuencia la obtención de mejores resultados en una situación determinada, por ejemplo, la ganancia máxima, el costo mínimo, la satisfacción de los usuarios, entre otros.

Este trabajo establece una base para futuras investigaciones, debido al apoyo de una guía práctica que no sólo relaciona el trabajo experimental y estadístico de los principales exponentes de PSO multi-enjambre en ambientes dinámicos, sino que insiste también en el aporte y adaptabilidad de estas técnicas a

diversas problemáticas que presenten dinamismo e incertidumbre en su solución.

Si se observa este trabajo desde el punto de vista **ambiental**, el documento que se presenta está redactado de un modo simple y claro en la medida de lo permisible. Se detallan las actividades experimentales realizadas y las conclusiones a que se llegó. El tipo y tamaño de letra empleados son correctos. Todo ello con el fin de propiciar un entendimiento fácil, dinámico y con un mínimo de riesgos de estrés psicológico, con lo que preserva la higiene mental de quien lo consulte. No provoca daños directos sobre el medio ambiente. Solo ocasiona gastos mínimos de energía eléctrica para ser leído o impreso, así como en la iluminación necesaria para esta actividad.

En el medio **tecnológico** se incluye este estudio en un área de investigación actual y de interés de la comunidad científica internacional. Además se señala que las conclusiones derivadas del mismo no demandan aumento de prestaciones del hardware ni el software que actualmente se usan para el desarrollo de aplicaciones que empleen técnicas PSO multi-enjambre para la resolución de problemas dinámicos.

Para consultar el documento solo se necesita una computadora con una herramienta para visualizar texto y cualquier tipo de impresora en el caso que se desee leer en formato duro. De igual forma, se cuenta con los medios necesarios para su publicación en Internet y servicios de ftp, etc.

A partir del análisis realizado en las cuatro dimensiones se puede afirmar que este estudio es sostenible en cada una de ellas, manteniendo su vigencia, aplicabilidad y continuidad en el futuro.

4.3 Conclusiones

Al finalizar este capítulo se puede concluir la guía de aplicación obtenida refleja el desempeño de los algoritmos PSO multi-enjambre según las principales características de los PDO. Además, la presente investigación resulta sostenible a raíz de la valoración de sostenibilidad desarrollada.

Conclusiones generales

Con el desarrollo del trabajo propuesto, dedicado a facilitar el proceso de selección y aplicación de los algoritmos PSO Multi-enjambre existentes en ambientes dinámicos, se ha dado cumplimiento al objetivo de esta investigación, pues como resultado se obtuvo una guía de aplicación fundamentada por un estudio experimental que compara y analiza el desempeño de los algoritmos multi-enjambres para diferentes PDO, quedando de esta forma demostrada la hipótesis enunciada en este documento. A continuación se relacionan las principales conclusiones a las que se arribaron:

- A través del estudio realizado se encontraron muy pocos trabajos que faciliten el proceso de selección y aplicación de los exponentes existentes de PSO multi-enjambre para diferentes PDO, por lo que se propuso como solución la elaboración de la guía desarrollada soportada en un estudio experimental.
- Las técnicas estadísticas y herramientas utilizadas fueron eficaces en la experimentación de los exponentes seleccionados, los cuales permitieron estudiar los aspectos más significativos de los algoritmos de optimización dinámica: configuración poblacional, adaptación frente a diferentes severidades, frecuencias y cantidad de óptimos.
- Los resultados obtenidos muestran claramente los beneficios de este trabajo: demostrar que existen diferencias altamente significativas entre el rendimiento de los algoritmos seleccionados para diferentes PDO.
- Los experimentos realizados infieren que existe un mejor rendimiento de manera general en los algoritmos que utilizan quantum con el operador CPT, que para la propuesta mPSO, constituida solamente por partículas neutrales.

- Las técnicas preservadoras de la diversidad no son desfavorables para aquellas instancias del problema donde no ocurran cambios.

Recomendaciones

Una vez concluida esta investigación se proponen las siguientes recomendaciones:

- Hacer extensivo el estudio de metaheurísticas PSO hacia otros tipos de problemas dinámicos que consideren en su solución más de un objetivo (multi-objetivos).
- Utilizar como referencia este trabajo para la selección y empleo de técnicas de optimización con enjambre de partículas en posibles problemas reales.

Bibliografía

1. Glover, F., *Future paths for integer programming and links to artificial intelligence*. *Computers and Operations Research*, 13:533-549. 1986.
2. Belén Melián, J.A.M.P., J. Marcos Moreno Vega *Metaheuristics: A global view* Revista Iberoamericana de Inteligencia Artificial. No.19, (2003): p. 7-28
3. Carlos Cruz, D.P., José L. Verdegay, *Propuesta de Solución a Problemas Dinámicos de optimización*. 2008.
4. Msc. María del Carmen Chávez, D.G.C.C., Dr. Rafael Bello and Dr. Ricardo Grau Ábalo *Modelo de red bayesiana para predicción de mutaciones en secuencias de la transcriptasa inversa del VIH usando PSO*. XIV Latin Ibero-American Congress on Operations Research, CLAIO 2008.
5. Susana Esquivel, V.A., Leticia Cagnina, *Optimización Mono y Multiobjetivo a través de Métodos de Aproximación de Soluciones en Ambientes Estacionarios y No Estacionarios*.
6. Weicker, K., *Evolutionary Algorithms and Dynamic Optimization Problems*. 2003, Stuttgart.
7. Branke, J., *Evolutionary Optimization in Dynamic Environments*. Kluwer Academic Publisher, 2002.
8. Sendho, Y.J.a.B., *Constructing Dynamic Optimization Test Problems Using the Multi-objective Optimization Concept*.
9. Jong, R.W.M.a.K.A.D. *A test problem generator for non-stationary environments* in *Congress on Evolutionary Computation*. 1999.
10. T.Back. *On the behavior of evolutionary algorithms in dynamic environments*. In *IEEE Congress on Evolutionary Computation*. 1998.
11. McGregor., D.D.a.D.R., *Purrillel Problem Solvingfioni Nutiire, chapter Nonstationary function optimization using the structured genetic algorithm, pages 145–154*. Elsevier Science Publisher. 1992.

12. Smith, D.G.a.R.E. *Nonstationary function optimization using genetic algorithms with diloidy and dominance*. In *2nd International Conference on Genetic Algorithms*. Lawrence Erlbaum Associates. 1987.
13. Angeline, P.J. *Tracking extrema in dynamic environments*. In *Proceedings of the 6th International Conference on Evolutionary Programming 1997*. Indianapolis, Indiana, USA: Springer Verlag.
14. M. Dorigo, E.B., and G. Theraulaz. , *Ant algorithms and stigmergy*. *Future Generation Computer Systems*, 16(8):851–871., 2000.
15. Swigger., K.P.a.E., *An analysis of genetic based pattern tracking and cognitive based component tracking models of adaptation*. In *Proceedings of National Conference on AI (AAAI-83 1983*.
16. K. Boese, A.K., and S. Muddu. , *Ant algorithms and stigmergy*. . *Operations Research Letters*, 1994. **16**: p. 103–113.
17. Jong., R.W.M.a.K.A.D., *A test problem generator for non-stationary environments*. , 1999. (Congr. on Evol. Comput.).
18. Branke, J., *Memory enhanced evolutionary algorithms for changing optimization problems*. 1999(Congr. on Evol. Comput).
19. Branke, J., "Evolutionary Algorithms for Dynamic Optimization Problems - A Survey -", *Technical Report No. 387, Institute AIFB, University of Karlsruhe, 76128 Karlsruhe e, Germany*. February 1999.
20. Carlos Cruz, D.A.P., José L. Verdegay, *Una Estrategia Cooperativa para la Resolución de Problemas Dinámicos de Optimización*. 2008.
21. Branke, J., *Evolutionary approaches to dynamic optimization problems: A survey*. Pp.134–137, *GECCO Workshops*, A. Wu (ed.). (1999b).
22. Weicker, K., *Performance Measures for Dynamic Environments*. Springer-Verlag Berlin Heidelberg 2002: p. 64–73.
23. Tfaili, W., J. Dreo, and P. Siarry, *Fitting of an Ant Colony approach to Dynamic Optimization through a new set of test functions*. *International Journal of Computational Intelligence Research*, 2007.
24. Yaochu Jin, S.M.I., and Jürgen Branke, Member IEEE, *Evolutionary Optimization in Uncertain .Environments—A Survey*. 2005 **9**, **NO. 3**.

25. Belén Melián, J.A.M.P., J. Marcos Moreno Vega *Metaheuristics: A global view* Revista Iberoamericana de Inteligencia Artificial., 2003. **No.19** p. 7-28
26. Fogel, Z.M.a.D.B., *How to Solve it: Modern Heuristic*. Springer-Verlag, Berlin, Heidelberg, Germany. 2000.
27. Hill, M., C. Reeves. *Modern Heuristic Techniques for Combinatorial Problems*, UK. 1995.
28. Osman., G.L.a.I.H., *Metaheuristics: A bibliography*. *Annals of Operations Research*, 63:513{623, 1996.
29. Branke, T.M.B.a.J. *Multi-swarm optimization in dynamic environments*, 3005 of LNCS, Springer. In *Applications of Evolutionary Computing* 2004.
30. Mario A. Muñoz, J.A.L.y.E.F.C., *Swarm intelligence: problem-solving societies (a review)*. *Revista Ingeniería e Investigación*, 2008. **VOL. 28 No. 2**: p. 119-130.
31. Fernández, J.J.P., *Resolución de Problemas de Optimización Dinámica mediante la Hibridación entre Filtros de Partículas y Metaheurísticas Poblacionales*, in *Departamento de Informática, Estadística y Telemática*. 2005, Universidad Rey Juan Carlos, Escuela Superior de Ciencias Experimentales y Tecnología.
32. Ganbardella, M.D.a.L.M., *Ant colony system: A cooperative learning approach to the travelling salesman problem*. *IEEE Transaction on Evolutionary Computation*, 1(1):53{66. 1997.
33. Eberhart, R., Kennedy, J., *A new optimizer using particle swarm theory*.,*Memorias del Sixth International Symposium on Micro Machine and Human Science*. . 1995: p. 39–43. .
34. Eberhart, J.K.a.R.C., *The Particle Swarm: Social Adaptation in Information Processing Systems*, chapter 25 in *New Ideas in Optimization*, pages 379{387. McGrawHill. 1999.
35. F.Javier Martínez García, J.A.M.P., *Optimización por enjambre para la p-mediana continua y discreta*
36. Y. Shi, R.C.E. "Parameter selection in particle swarm optimization", *Lecture Notes In Computer Science*, 1447. In *Proceedings of the 7th International Conference on Evolutionary Programming VII*. 1998.

37. Clerc, M. *The Swarm and the Queen: Towards and Deterministic an Adaptive Particle Swarm Optimization*. In *Proceedings of the Congress on Evolutionary Computations*. 1999. July, Washington DC, USA: IEEE, Service Center, and Piscataway, NJ.
38. Bergh, F.v.d., *An Analysis of Particle Swarm Optimizers*. November 2001, Pretoria.
39. Blackwell, T., *Particle Swarm Optimization in Dynamic Environments*.
40. Blackwell, T.M. *Swarms in dynamic environments*, Springer. In *Genetic and Evolutionary Computation Conference*. 2003.
41. Blackwell, T.M. *Swarms in dynamic environments* in E. Cantu-Paz, editor, *Genetic and Evolutionary Computation Conference*, Springer. 2003.
42. T. Blackwell and J. Branke. "Multi-swarm optimization in dynamic environments," in *Applications of Evolutionary Computing*. ser. *Lecture Notes in Computer Science*. R. Raidl et al., Eds. Berlin, Germany: Springer-Verlag, vol. 3005, pp. 489–500. 2004.
43. Dozier, A.C.a.G. *Adapting particle swarm optimization to dynamic environments*. In *Proc of int conference on artificial intelligence*. 2000. Pages 429–434.
44. J. P. Coelho, P.B.D.M.O., and J. E. Boaventura Cunha, "Nonlinear concentration control system design using a new adaptive particle swarm optimizer," in *Proc. Portuguese Conf. Autom. Control*, pp.132–137. 2002.
45. "Tracking and optimizing dynamic systems with particle swarms," in *Proc. Congr. Evol. Comput.* pp. 94–100. 2001.
46. Eberhart, X.H.a.R. "Adaptive particle swarm optimization: Detection and response to dynamic systems," in *Proc. Congr. Evol. Comput.* 2002.
47. Middendorf., S.J.a.M. *A hierarchical particle swarm optimizer for dynamic optimization problems* in *Applications of evolutionary computing*. Springer, 2004.
48. Dam, X.L.a.K.H. "Comparing particle swarms for tracking extrema in dynamic environments," in *Proc. Congr. Evol. Comput.* Pp.1772–1779. 2003.

49. Vrahatis, K.E.P.a.M.N. *Recent approaches to global optimization problems through particle swarm optimization*. In *Natural Computing*. 2002.
50. X. Zhang, L.Y., Y. Zheng, Y. Shen, G. Zhou, L. Chen, L. Xi, T. Yuan, J. Zhang, and B. Yang, "Two-stage adaptive PMD compensation in a 10 Gbit/s optical communication system using particle swarm optimization algorithm," *Optics Commun.*, vol. 231, pp. 233–242. 2004.
51. Bentley, T.M.B.a.P.J. "Dynamic search with charged swarms," in *Proc. Genetic Evol. Comput. Conf.*, W. B. Langdon et al., Eds., pp. 19–26, Morgan Kaufmann. 2002.
52. Blackwell, T.M. *Particle swarms and population diversity II: Experiments*. In *GECCO Workshop on Evolutionary Algorithms for Dynamic Optimization Problems*. 2003.
53. Bentley., T.M.B.a.P. *Don't push me! collision avoiding swarms*. In *Congress on Evolutionary Computation*, pages 1691–1696. 2002.
54. Bentley., T.M.B.a.P.J. *Dynamic search with charged swarms*. . In *Genetic and Evolutionary Computation Conference*, pages 19–26. Morgan Kaufmann. 2002.
55. J.Branke, *Evolutionary Optimization in Dynamic Environments*. Norwell, MA: Kluwer. 2001.
56. Li, D.P.a.X., "A particle swarm model for tracking multiple peaks in a dynamic environment using speciation," in *Proc. Congr. Evol. Comput.* pp. 98–103. 2004.
57. Yang, C.L.a.S. *Fast Multi-swarm Optimization for Dynamic Optimization Problems*. In *Fourth International Conference on Natural Computation*, IEEE. 2008.
58. Liu, X.Y.a.Y. *Fast evolutionary programming*. *Proc. of the 5th Annual Conference on Evolutionary Programming*, pp. 451-460. 1996.
59. Tim Blackwell and Jürgen Branke, M.I. *Multiswarms, Exclusion, and Anti-Convergence in Dynamic Environments*. In *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, AUGUST 2006.
60. Bentley, T.M.B.a.P. "Don't push me! Collision-avoiding swarms" in *Proc. Congr. Evol. Comput*, pp. 1691–1696. 2002.

61. Macready., D.H.W.a.W.G., *No Free Lunch Theorems for Search.* . Technical Report SFI-TR-95-02-010, The Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM, 87501, USA, February 6, 1995. Online available at <http://citeseer.ist.psu.edu/wolpert95no.html> and http://www.santafe.edu/research/publications/working_papers/95-02-010.pdf [accessed 2008-03-28]. 1995.
62. Macready., D.H.W.a.W.G., *No Free Lunch Theorems for Optimization.* . 1997. (IEEE Transactions on Evolutionary Computation, 1(1):67–82, April 1997. doi:10.1109/4235.585893. Online available at <http://citeseer.ist.psu.edu/wolpert96no.html> [accessed 2008-03-28].).
63. Novoa, P., Pelta, D., Cruz, C. and Garcia del Amo, I. *Controlling Particle Trajectories in a Multi-Swarm Approach for Dynamic Optimization Problems.3rd International Work-conference on the Interplay between Natural and Artificial Computation, IWINAC'09.Santiago de Compostela, Spain.* June 2009.
64. *Sun-Microsystems.Java Plataform.* [Cited mayo, 6, 2009]; Available from:<http://java.sun.com/>
65. *NetBeans.* [Cited mayo-2009]; Available from:<http://www.netbeans.org/downloads/index.html>
66. Caballero, B.G., *Laboratorio para la Experimentación de Problemas Dinámicos de Optimización (Trabajo de Diploma),* in *Matemática.* 2009, UHO Oscar Lucero Moya.

Anexos

ANEXO 1. ALGORITMO MULTI-ENJAMBRE.

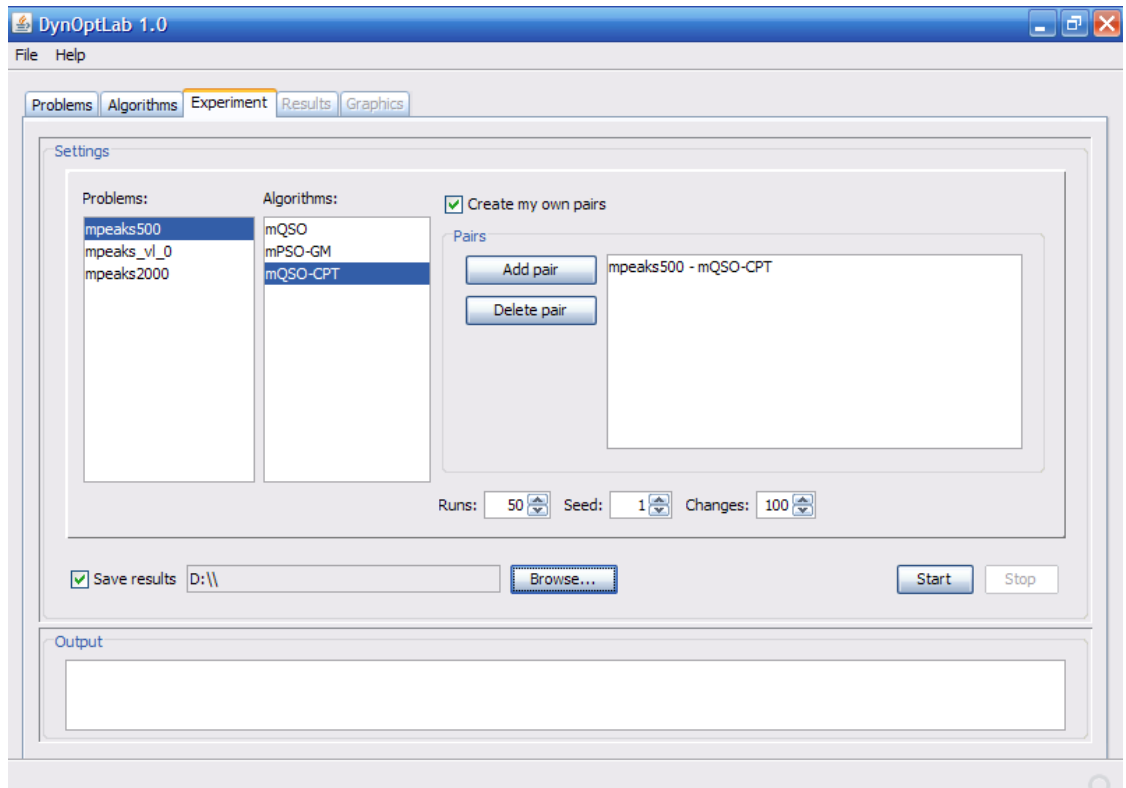
```
Update swarm attractor.
//Cancel randomization
init[n] := FALSE
FOR EACH particle  $i$  of swarm  $n$ 
  IF (init[n] = TRUE) THEN
    randomize particle
  ELSE
    //Update Particle
    Apply equations (3)-(9) depending
    on particle type.
    //Update Attractor
    Evaluate  $f(\vec{x}_{ni})$ .
    IF  $f(\vec{x}_{ni}) > f(\vec{p}_{ni})$  THEN
       $\vec{p}_{ni} := \vec{x}_{ni}$ .
    IF  $f(\vec{x}_{ni}) > f(\vec{p}_{ng})$  THEN
       $\vec{p}_{ng} := \vec{x}_{ni}$ 
    //Randomization complete
  init[n] := FALSE
UNTIL number of function evaluations
performed  $> max$ 
```

```
Update swarm attractor.
//Cancel randomization
init[n] := FALSE
FOR EACH particle  $i$  of swarm  $n$ 
  IF (init[n] = TRUE) THEN
    randomize particle
  ELSE
    //Update Particle
    Apply equations (3)-(9) depending
    on particle type.
    //Update Attractor
    Evaluate  $f(\vec{x}_{ni})$ .
    IF  $f(\vec{x}_{ni}) > f(\vec{p}_{ni})$  THEN
       $\vec{p}_{ni} := \vec{x}_{ni}$ .
    IF  $f(\vec{x}_{ni}) > f(\vec{p}_{ng})$  THEN
       $\vec{p}_{ng} := \vec{x}_{ni}$ 
    //Randomization complete
  init[n] := FALSE
UNTIL number of function evaluations
performed  $> max$ 
```

ANEXO 2. ALGORITMO MPSO-CPT.

```
1: for each particle  $p$  in swarm  $s$  do
2:   Randomly initialize  $p.v, p.x = p.x_{pbest}$ 
3:    $p.num\_failures = 0$ 
4:   Update  $s.x_{gbest}$ 
5: end for
6: while (!End) do
7:   Test for Exclusion
8:   Test for Convergence
9:   Test for Change
10:  for each particle  $p$  in swarm  $s$  do
11:    if  $p.num\_failures = max\_failures$  then
12:      Update  $p$  with expressions 4) y 5)
13:    else
14:      Update  $p$  with expressions 2) y 3)
15:      Evaluate  $p.x$ 
16:      Update  $p.x_{pbest}$ 
17:       $p.num\_failures = 0$ 
18:    end if
19:    Update  $s.x_{gbest}$ 
20:  end for
21: end while
```

ANEXO 3. SOFTWARE PARA LA EXPERIMENTACIÓN DE PROBLEMAS DINÁMICOS DYNOPTLAB.



ANEXO 4. PAQUETE SPSS 15.0 (ANALISIS ANOVA)

The screenshot displays the SPSS 15.0 interface. The main window is titled "deltaE.sav [Conjunto_de_datos1] - Editor de datos SPSS". The menu bar includes "Archivo", "Edición", "Ver", "Datos", "Transformar", "Analizar", "Gráficos", "Utilidades", and "Ventana ?". The toolbar contains various icons for file operations and data manipulation. The data editor shows a table with the following data:

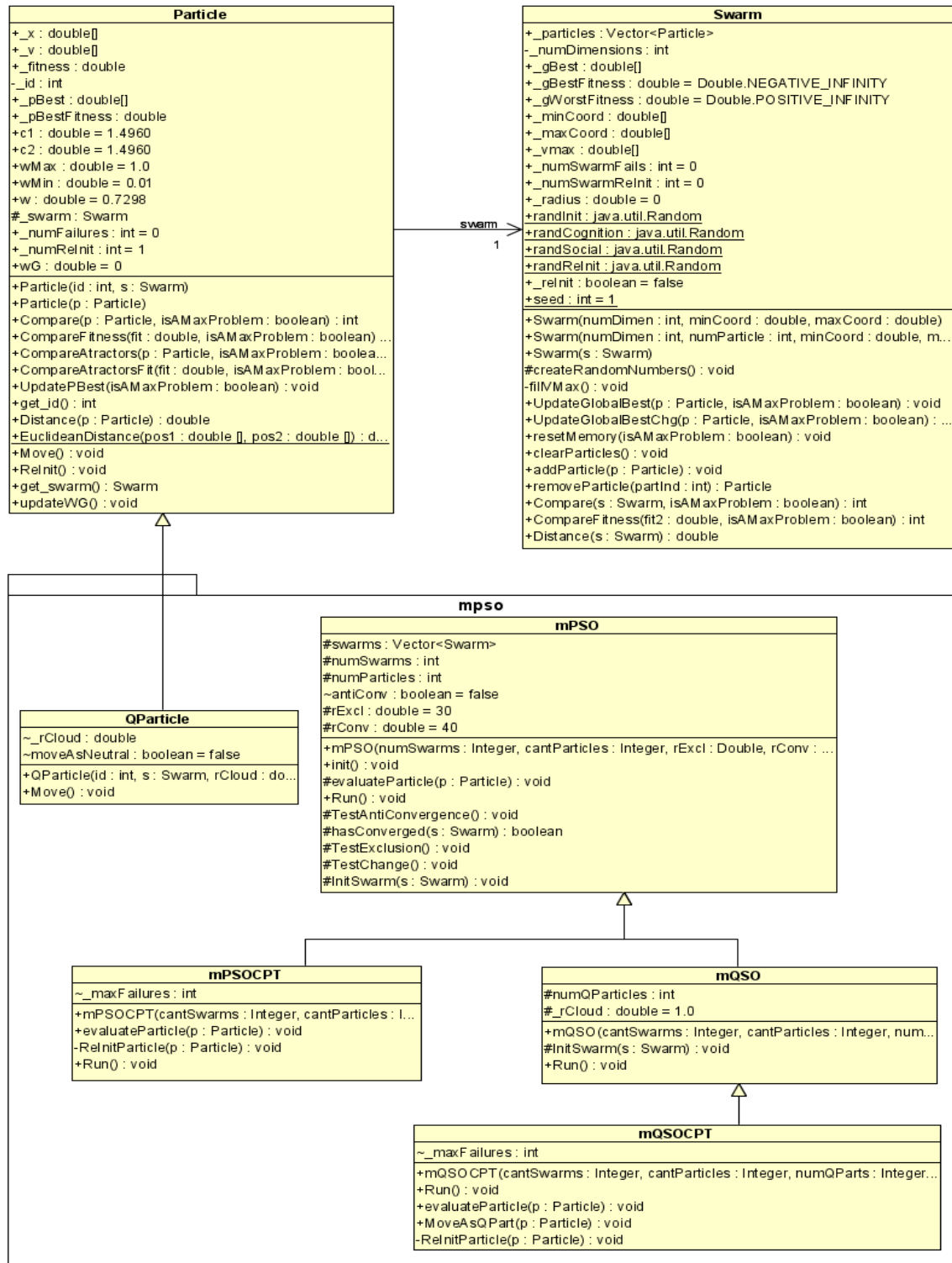
	algoritmo	error	performance	var	var	var	var	var
1	1	14,09430	51,79694					
2	1	10,56607	56,22099					
3	1	11,55081	55,17658					
4	1	9,08167	57,79286					
5	1	12,12712	55,01324					
6	1	8,42045	59,01538					
7	1	8,70089	57,82622					
8	1	10,39446	56,61905					
9	1	10,57308	56,88395					
10	1	8,63256	58,22700					
11	1	8,61215	58,87330					
12	1	16,28403	49,62660					
13	1	10,59244	55,92710					
14	1	18,90915	47,38510					
15	1	10,45220	56,46960					
16	1	12,26893	53,44070					
17	1	14,61334	52,02250					
18	1	10,86243	56,19450					
19	1	9,17940	57,42770					
20	1	10,31643	55,67360					
21	1	9,23754	57,16750					
22	1	10,05688	56,42714					
23	1	10,10513	55,29920					
24	1	8,94786	56,71491					
25	1	13,60551	52,39621					

An "ANOVA de un factor" dialog box is open, showing the following configuration:

- Dependientes: Rendimiento fuera de línea
- Factor: Algoritmos [algoritmo]
- Buttons: Aceptar, Pegar, Restablecer, Cancelar, Ayuda, Contrastes, Post hoc..., Opciones...

The status bar at the bottom indicates "SPSS El procesador está preparado".

ANEXO 5. CLASES IMPLEMENTADAS



ANEXO 6. CONFIGURACIÓN MULTI-ENJAMBRE PARA 4(13+12)

Comparaciones múltiples. Variable dependiente: Error fuera de línea

	(I) Algoritmos	(J) Algoritmos	Diferencia de medias (I-J)	Error típico	Sig.
			Límite inferior	Límite superior	Límite inferior
Bonferroni	mPSO	mQSO	,6162608(*)	,2185784	,032
		mPSO-CPT	,5016926	,2185784	,137
		mQSO-CPT	1,5143101(*)	,2185784	,000
	mQSO	mPSO	-,6162608(*)	,2185784	,032
		mPSO-CPT	-,1145683	,2185784	1,000
		mQSO-CPT	,8980493(*)	,2185784	,000
	mPSO-CPT	mPSO	-,5016926	,2185784	,137
		mQSO	,1145683	,2185784	1,000
		mQSO-CPT	1,0126175(*)	,2185784	,000
	mQSO-CPT	mPSO	-1,5143101(*)	,2185784	,000
		mQSO	-,8980493(*)	,2185784	,000
		mPSO-CPT	-1,0126175(*)	,2185784	,000

* La diferencia de medias es significativa al nivel .05.

ANEXO 7. DIFERENCIAS SIGNIFICATIVAS ENTRE LA MEDIA DE LOS ALGORITMOS PARA $\Delta e = 10000$.

Comparaciones múltiples

Variable dependiente: Error fuera de línea

Tamhane

(I) Algoritmos	(J) Algoritmos	Diferencia de medias (I-J)	Error típico	Sig.	Intervalo de confianza al 95%	
					Límite inferior	Límite superior
		Límite inferior	Límite superior	Límite inferior	Límite superior	Límite inferior
mPSO	mQSO	5,7246288(*)	,2411861	,000	5,066082	6,383175
	mPSO-CPT	4,9058137(*)	,2475790	,000	4,232122	5,579505
	mQSO-CPT	6,1730089(*)	,2383836	,000	5,520991	6,825027
mQSO	mPSO	-5,7246288(*)	,2411861	,000	-6,383175	-5,066082
	mPSO-CPT	-,8188151(*)	,0952313	,000	-1,075159	-,562471
	mQSO-CPT	,4483801(*)	,0678260	,000	,265931	,630829
mPSO-CPT	mPSO	-4,9058137(*)	,2475790	,000	-5,579505	-4,232122
	mQSO	,8188151(*)	,0952313	,000	,562471	1,075159
	mQSO-CPT	1,2671951(*)	,0878920	,000	1,029549	1,504841
mQSO-CPT	mPSO	-6,1730089(*)	,2383836	,000	-6,825027	-5,520991
	mQSO	-,4483801(*)	,0678260	,000	-,630829	-,265931
	mPSO-CPT	-1,2671951(*)	,0878920	,000	-1,504841	-1,029549

* La diferencia de medias es significativa al nivel .05.